

# Title Masterproef

Subtitle (optional)

first name LAST NAME

Promotor(en): My Promotor

Co-promotor(en): My Co-promotor  
My company promotor

Masterproef ingediend tot het behalen van  
de graad van master of Science in de  
industriële wetenschappen: discipline name  
specialisation

Academiejaar 2024 - 2025

©Copyright KU Leuven

Deze masterproef is een examendocument dat niet werd gecorigeerd voor eventuele vastgestelde fouten.

Zonder voorafgaande schriftelijke toestemming van zowel de promotor(en) als de auteur(s) is overnemen, kopiëren, gebruiken of realiseren van deze uitgave of gedeelten ervan verboden. Voor aanvragen i.v.m. het overnemen en/of gebruik en/of realisatie van gedeelten uit deze publicatie, kan u zich richten tot KU Leuven Technologiecampus Gent, Gebroeders De Smetstraat 1, B-9000 Gent, +32 92 65 86 10 of via e-mail [iiw.gent@kuleuven.be](mailto:iiw.gent@kuleuven.be).

Voorafgaande schriftelijke toestemming van de promotor(en) is eveneens vereist voor het aanwenden van de in deze masterproef beschreven (originele) methoden, producten, schakelingen en programma's voor industrieel of commercieel nut en voor de inzending van deze publicatie ter deelname aan wetenschappelijke prijzen of wedstrijden.

# Voorwoord

Het voorwoord vul je persoonlijk in met een appreciatie of dankbetuiging aan de mensen die je hebben bijgestaan tijdens het verwezenlijken van je masterproef en je hebben gesteund tijdens je studie.

In the preface, you write a personal account with an acknowledgement or expression of gratitude to the people who assisted you in completing your master's thesis and supported you during your studies.



*Copyright informatie:  
studiewerk van een student in het kader van de  
academische opleiding en examenbeoordeling.  
Na deze beoordeling vond geen correctie plaats  
van het studiewerk.*

# Samenvatting

De (korte) samenvatting, toegankelijk voor een breed publiek, wordt in het Nederlands geschreven en bevat **maximum 3500 tekens**. Deze samenvatting moet ook verplicht opgeladen worden in KU Loket.

# Summary

This (short) summary, accessible to a wide audience, is written in English and contains a **maximum of 3500 characters**. This summary must also be uploaded to KU Loket.

Keywords: Enter (more or less) five keywords.

# Inhoudsopgave

<b>1 Rationale en doelstelling</b>	<b>2</b>
1.1 Bestaande tools . . . . .	2
1.1.1 Waveform lighting . . . . .	2
1.1.2 OSRAM ColorCalculator . . . . .	3
1.1.3 Excel KU Leuven . . . . .	4
1.1.4 Conclusie . . . . .	4
<b>2 Relevante Concepten</b>	<b>6</b>
2.1 Spectral power distribution . . . . .	6
2.2 External quantum efficiency . . . . .	7
2.3 Chromaticiteit . . . . .	7
2.3.1 Correlated color temperature . . . . .	9
2.4 Color rendering index . . . . .	10
2.4.1 Berekening CRI [15, 16] . . . . .	10
2.5 tm30 . . . . .	12
2.5.1 Fidelity index . . . . .	12
2.5.2 Hue-angle bins . . . . .	13
2.5.3 Gamut (area) index . . . . .	13
2.5.4 Color vector graphic . . . . .	14
2.5.5 Local chroma shift . . . . .	14
2.5.6 Local hue shift . . . . .	15
2.6 Luminescente materialen . . . . .	15
2.6.1 Werking luminescente materialen . . . . .	15
2.6.2 Witte LEDs . . . . .	17
2.6.3 Verschillen tussen luminescente materialen . . . . .	18
<b>3 Web applicatie</b>	<b>19</b>

3.1	Architectuur . . . . .	19
3.1.1	Frontend . . . . .	19
3.1.2	Backend . . . . .	20
3.2	Algemene structuur . . . . .	20
3.2.1	Inputs . . . . .	20
3.2.2	Outputs . . . . .	21
3.3	Structuur Vue . . . . .	22
3.4	Implementatie inputs . . . . .	23
3.4.1	LED Input . . . . .	24
3.4.2	Fosfor input . . . . .	24
3.4.3	Quantum dot input . . . . .	25
3.4.4	Luminescent materiaal input . . . . .	26
3.5	Implementatie LEDs en SPD . . . . .	28
3.5.1	Implementatie LED model . . . . .	28
3.5.2	Implementatie SPD . . . . .	29
3.5.3	Model EQE . . . . .	30
3.6	Implementatie chromaticiteit . . . . .	31
3.7	Implementatie color rendering . . . . .	33
3.7.1	CIE Ra-15 . . . . .	33
3.7.2	ies-tm30 . . . . .	34
3.8	Implementatie luminescente materialen . . . . .	34
3.8.1	Simpele berekening . . . . .	34
3.8.2	Geavanceerde berekening . . . . .	35
3.8.3	Spectra . . . . .	37
3.9	Implementatie LED visualisatie . . . . .	38
3.10	Implementatie hyperspectral image rendering . . . . .	39
3.10.1	Foto . . . . .	40
3.10.2	Berekeningen . . . . .	42
3.11	Implementatie data manager . . . . .	43
3.12	Hosting . . . . .	44
<b>4</b>	<b>Uitbreidingen</b>	<b>45</b>
4.1	Gamification . . . . .	45
4.2	Analyse model luminescente materialen . . . . .	46

<b>5 Conclusie</b>	<b>47</b>
<b>A Een aanhangsel</b>	<b>50</b>
<b>B Beschrijving van deze masterproef in de vorm van een wetenschappelijk artikel</b>	<b>51</b>
<b>C Poster</b>	<b>53</b>
<b>D GenAI code of conduct for students</b>	<b>54</b>

# Acroniemen

**CCT** color correlated temperature. 8, 9, 10, 22, 32

**CES** color evaluation samples. 12, 13

**CMF** color matching functions. 7, 8

**CRI** color rendering index. 2, 3, 4, 8, 10, 11, 17, 20, 21, 33

**CSV** comma-separated values. 26, 30, 32, 44

**EQE** external quantum efficiency. 4, 7, 17, 24, 29, 30, 31, 44

**FWHM** full width at half maximum. 24, 28, 29

**LED** Light Emitting Diode. 2, 3, 4, 6, 7, 10, 15, 17, 20, 21, 22, 23, 24, 28, 29, 30, 31, 32, 33, 38, 39, 45

**nm** nanometer. 3

**PLQY** Photoluminescence quantum yield. 23, 25, 26, 27, 36

**SPD** spectral power distribution. 2, 3, 4, 6, 7, 8, 28, 29, 30, 31, 33

**UI** user interface. 4

# **Hoofdstuk 1**

## **Rationale en doelstelling**

Sinds de introductie van de Light Emitting Diode (LED) is de verlichtingswereld sterk veranderd. In vergelijking met traditionele verlichtingstechnologieën zijn LEDs kleine en efficiënte lichtbronnen. LEDs kunnen verschillende kleuren produceren door basiskleuren te combineren, maar het is moeilijk om efficiënt kleuren zoals wit te maken. Om dit te bereiken, gebruikt men luminescente materialen zoals fosforen, waarmee men op een efficiënte manier LEDs in deze kleuren kan creëren. Daarnaast moeten LEDs voldoen aan specifieke kleurweergave-eisen.

Belangrijke begrippen zoals kleur en kleurweergave spelen hierbij een rol. Deze begrippen, de werking van LEDs en de effecten van verschillende factoren zijn echter moeilijk intuïtief te begrijpen.

Om het aanleren van deze concepten te vergemakkelijken, wordt in deze thesis een educatieve LED webapp ontwikkeld. In deze app kunnen gebruikers experimenteren met verschillende waarden, LEDs, luminescente materialen, enzovoort, en op een experimentele manier leren over de effecten hiervan.

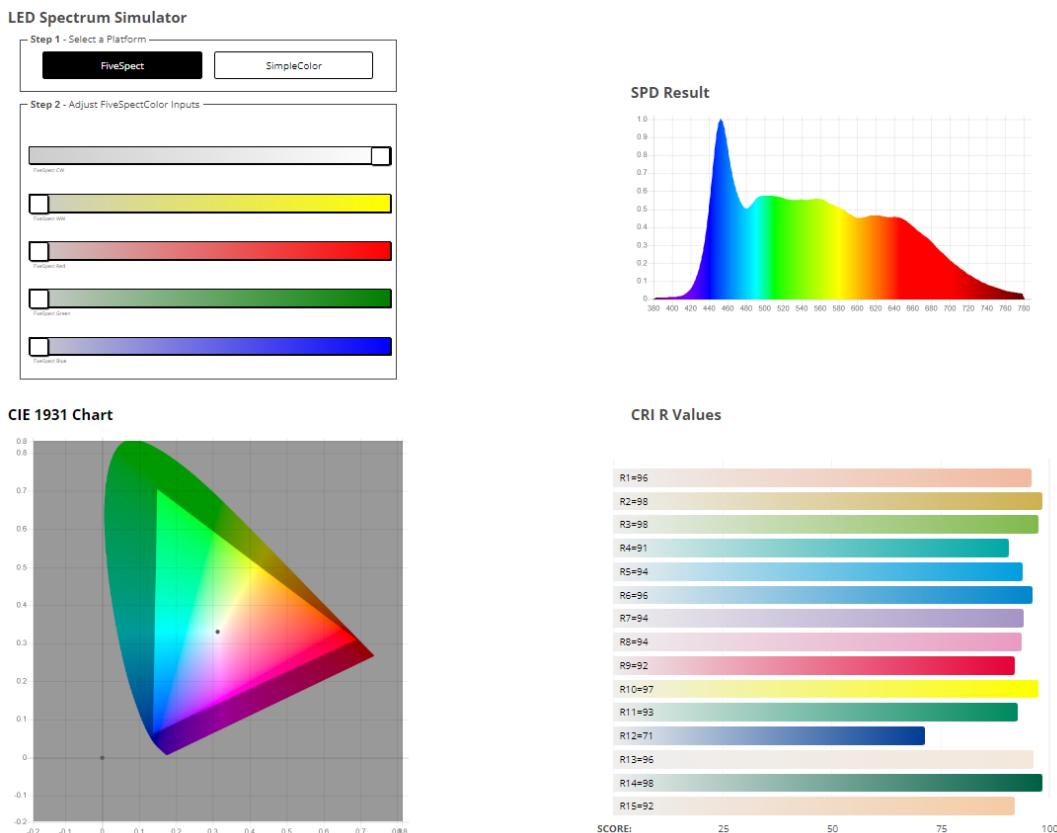
### **1.1 Bestaande tools**

Er bestaan natuurlijk al tools die vergelijkbare functionaliteiten bieden als die wij in de app willen implementeren. Daarom hebben wij verschillende tools onderzocht en gekeken hoe zij deze functionaliteiten realiseren.

#### **1.1.1 Waveform lighting**

Op de website van Waveform Lighting zijn enkele eenvoudige webapps te vinden voor het berekenen van de spectral power distribution (SPD), chromaticiteit en de color rendering index (CRI). Twee van hun beste tools zijn de “LED Spectrum Simulator” en de “Color Rendering Index Calculator”. Deze tools zijn echter op een vrij simpele manier ontworpen.

In de spectrum simulator kan de gebruiker bijvoorbeeld alleen het vermogen van vier LEDs met een vast profiel aanpassen. Dit beperkt de mogelijkheid om verschillende combinaties van LEDs uit te proberen en staat dus minder experimentatie toe. De color rendering index calculator biedt wat meer vrijheid, maar is minder gebruiksvriendelijk. De gebruiker moet namelijk alle 401 vermogenswaarden van 380 nanometer (nm) tot 780 nm handmatig invoeren om de CRI te berekenen. Bovendien is het bij deze tools minder handig dat deze allemaal apart zijn. Het is dus niet mogelijk om een SPD te maken en hiervan het effect op chromaticiteit, CRI, enzovoort te zien. Het zijn echter wel makkelijk te vinden tools die vrij makkelijk te gebruiken zijn.

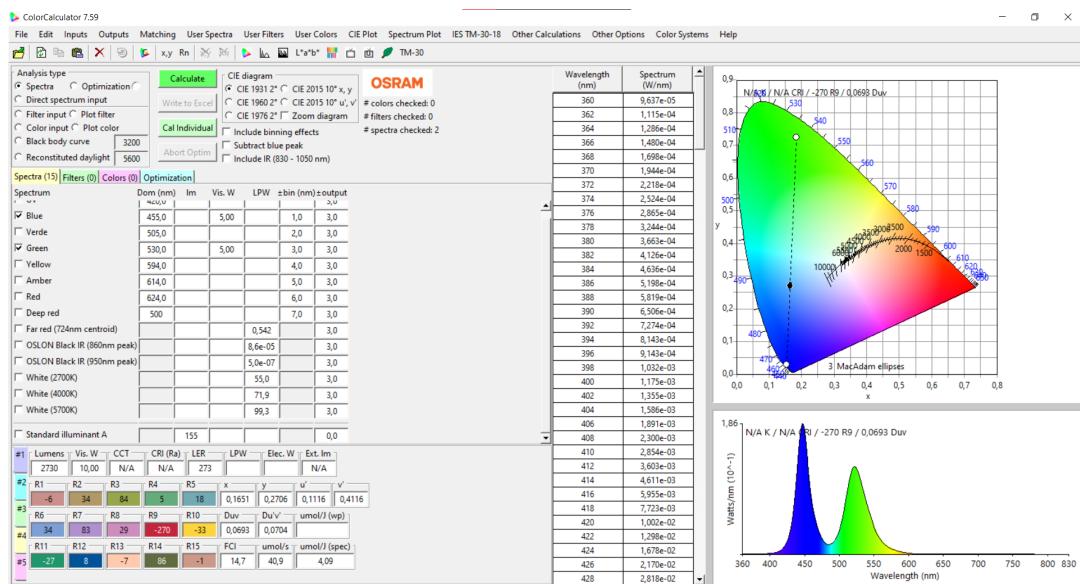


**Figuur 1.1:** Waveform lighting spectrum simulator [1]

### 1.1.2 OSRAM ColorCalculator

De OSRAM ColorCalculator applicatie is een tool die bedoeld is voor het ontwikkelen van "color mixing LED lighting solutions". De software heeft twee hoofdmodi: een "General photometry" mode en een "optimization" mode. Net als de "LED Spectrum Simulator" tool van Waveform Lighting bevat het een chromaticity diagram, CRI-waarden en een SPD diagram. Daarnaast biedt het een reeks andere plots, zoals  $L^*a^*b^*$ ,  $\Delta(u,v)$ , enzovoort. Deze plots zijn echter moeilijker te vinden. Het grootste probleem van de OSRAM Color-

Calculator applicatie is echter de drukke user interface (UI), die de tool moeilijk bruikbaar maakt. Dit zorgt ervoor dat dit een tool is die minder geschikt is voor een lerende gebruiker. Bovendien heeft de applicatie, net als de Waveform Lighting tools, geen mogelijkheid om vrij een LED toe te voegen met een golflengte tussen 380 nm en 780 nm. Gebruikers moeten LEDs toevoegen met golflengtes in een specifiek bereik. Het grootste probleem is echter dat de applicatie moeilijk te vinden is. Alle downloads op de website van OSRAM zijn verwijderd, en de enige plaats waar de applicatie nog beschikbaar is, is bij de paper op ResearchGate [2]. Dit maakt het moeilijk voor een gebruiker om de tool te vinden en te gebruiken.



**Figuur 1.2:** OSRAM ColorCalculator [2]

### 1.1.3 Excel KU Leuven

Binnen de richting Ingenieurswetenschappen aan KU Leuven Gent wordt een Excel-bestand gebruikt. Dit bestand biedt de mogelijkheid om vier LEDs en twee fosforen in te stellen. Het bevat grafieken van de SPD, de chromaticiteit, de external quantum efficiency (EQE) van de LEDs en een color view (met de CRI). Deze tool is echter niet erg gebruiksvriendelijk, aangezien de gebruiker alle waarden handmatig moet invoeren. Dit maakt het moeilijk voor de gebruiker om met waarden te experimenteren en de effecten daarvan te zien. Bovendien wordt experimenteren bemoeilijkt doordat de resultaten en de inputs zich op verschillende tabbladen bevinden. Ten slotte is deze Excel alleen beschikbaar voor studenten van KU Leuven die het vak volgen waarin deze wordt gebruikt.

### 1.1.4 Conclusie

Er zijn dus al enkele tools beschikbaar die vergelijkbare functionaliteiten bieden. Deze tools zijn echter te simpel, te ongebruiksvriendelijk en moeilijk te vinden. Bovendien zijn er geen

tools die functionaliteit voor luminescente materialen zoals fosforen en quantum dots aanbieden. Het is dus zeker een goed idee om een gebruiksvriendelijke, gemakkelijk te vinden webapp te ontwikkelen die deze functionaliteiten implementeert. De grootste voordelen van onze webapp zijn dus de vrijheid van gebruik en de simulatie van luminescente materialen. Daarnaast bied de app een grotere mogelijkheid tot experimenteren.

# **Hoofdstuk 2**

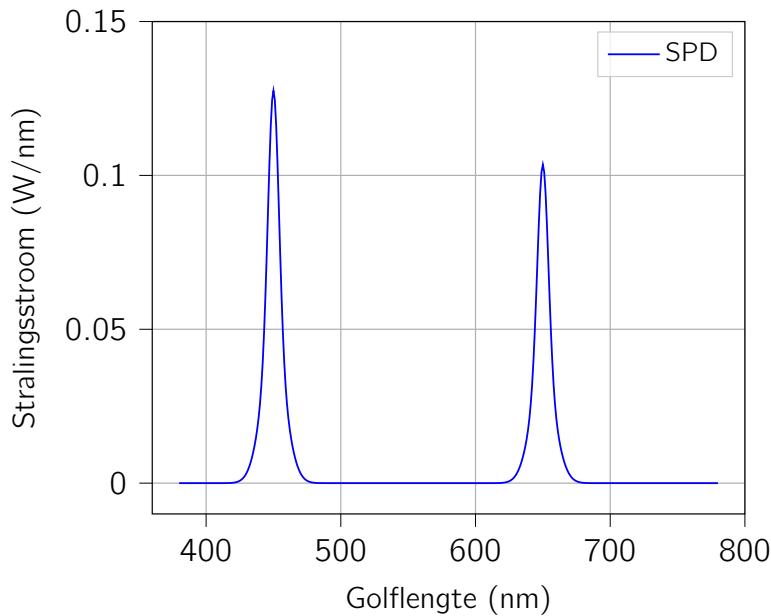
## **Relevante Concepten**

In dit hoofdstuk leggen we alle concepten uit die nodig zijn om de volledige functionaliteit van de webapp te begrijpen.

Dit hoofdstuk is hoofdzakelijk gebaseerd op [3–6]

### **2.1 Spectral power distribution**

De SPD is een essentieel concept voor het gebruik van de app en de verschillende berekeningen die hierin worden uitgevoerd. Het vormt de basis voor alle andere berekeningen. Bovendien maakt de SPD het eenvoudig om experimentele aanpassingen snel te visualiseren. Aan de hand van de SPD kunnen we per golflengte zien hoeveel optische straling wordt uitgezonden. Het wordt gedefinieerd als de hoeveelheid optische straling die door een bron wordt uitgezonden binnen een smalle band  $\Delta\lambda$ , geцentreerd rond een specifieke golflengte  $\lambda$ . Door een grafiek van de spectral power distribution te maken, kunnen we bijvoorbeeld zien bij welke golflengte een LED de grootste optische straling heeft en welk effect het toevoegen van luminescente materialen heeft.



**Figuur 2.1:** Voorbeeld spectral power distribution

## 2.2 External quantum efficiency

Het volgende belangrijke concept is de EQE. Dit begrip geeft aan hoeveel fotonen er voor de inkomende elektronen de LED verlaten. Dit is cruciaal bij het berekenen van de spectral power distribution van een LED. Afhankelijk van de centrale golflengte kan de external quantum efficiency namelijk sterk variëren. Dit zorgt ervoor dat de hoeveelheid stralingsstroom voor bepaalde LEDs een stuk kleiner is, doordat minder fotonen de LED kunnen verlaten. Zo hebben bijvoorbeeld groene LEDs een veel lagere efficiëntie dan andere LEDs (the green gap [7–10]), wat ervoor zorgt dat deze LEDs meer vermogen nodig hebben. Het effect van de EQE is rechtstreeks af te lezen in de spectral power distribution. Zo zal een groene LED met hetzelfde ingangsvermogen als een blauwe LED minder licht produceren.

## 2.3 Chromaticiteit

Door de verschillende SPDs van lichtbronnen ontstaat voor de mens een bepaalde kleur. Om te bepalen met welke kleur een SPD overeenkomt, maken we gebruik van een chromaticiteitsdiagram. Dit diagram helpt ons de kleur te visualiseren die wordt waargenomen op basis van de SPD van een lichtbron. Om op het chromaticiteitsdiagram tot een kleur te komen, moeten we eerst gebruik maken van de  $\bar{x}(\lambda)$ ,  $\bar{y}(\lambda)$ , en  $\bar{z}(\lambda)$  color matching functions (CMF) en de X, Y en Z tristimuluswaarden.

De CMF (colour matching functions) zijn numerieke waarden die de chromatische respons van de waarnemer beschrijven. Chromatische respons betekent de manier waarop het menselijk

oog verschillende golflengten van licht (kleuren) waarneemt. Het menselijk oog is namelijk gevoelig voor verschillende golflengtes, en de CMF geven aan hoe sterk het oog reageert op licht bij verschillende golflengtes. Dit zijn als het ware de "gevoelighedsfuncties" voor de verschillende kleuren die we kunnen waarnemen.

De X, Y en Z tristimuluswaarden zijn net zoals de RGB-waarden (rood, groen, blauw) een manier om een kleur voor te stellen, maar ze zijn gebaseerd op hoe een gemiddeld menselijk oog reageert op licht. De tristimuluswaarden X, Y en Z geven een gestandaardiseerde manier om de kleur van een lichtbron te berekenen op basis van de waarneming van de mens.

De waarnemer die we gebruiken om de CMF te berekenen, is gebaseerd op de chromatische respons van een gemiddelde mens, die een respons heeft binnen een boog van twee graden van het gezichtsveld. Dit model wordt gebruikt omdat het menselijk zicht niet overal in het gezichtsveld even gevoelig is voor kleur. De CMFs kunnen we dan zien als de spectrale "intensiteit" curves van drie lichtdetectoren, die samen de CIE-tristimuluswaarden X, Y en Z opleveren.

Aan de hand van de CMF en de SPD van de te analyseren lichtbron kunnen we de X, Y en Z waarden berekenen op de volgende manier [3, 11]:

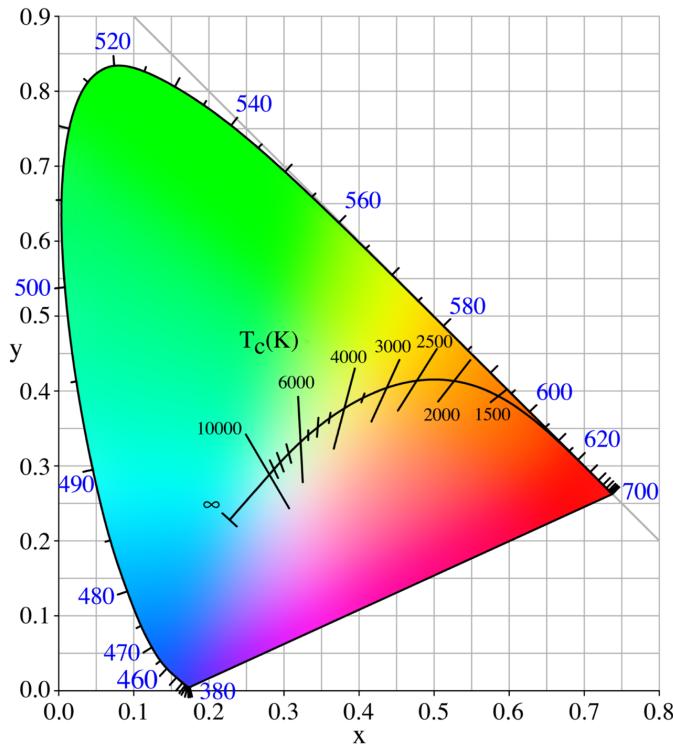
$$\begin{aligned} X &= k \cdot \int_{\lambda} \bar{x}(\lambda) P(\lambda) d\lambda \\ Y &= k \cdot \int_{\lambda} \bar{y}(\lambda) P(\lambda) d\lambda \\ Z &= k \cdot \int_{\lambda} \bar{z}(\lambda) P(\lambda) d\lambda \end{aligned}$$

Met  $P(\lambda)$  de SPD van de lichtbron en  $k$  een normalisatieconstante.

Aan de hand van deze tristimulus waarden kunnen vervolgens de chromaticiteitscoördinaten berekend worden.

$$\begin{aligned} x &= \frac{X}{X + Y + Z} \\ y &= \frac{Y}{X + Y + Z} \\ z &= \frac{Z}{X + Y + Z} \end{aligned}$$

Conventioneel worden de x- en y-coördinaten in het chromaticiteitsdiagram gebruikt om een punt of kleur aan te duiden. Het is belangrijk om te begrijpen dat deze coördinaten niet bedoeld zijn om een specifieke kleur vast te leggen, maar om de kleur te situeren. Dit komt doordat het chromaticiteitsdiagram enkel rekening houdt met de tint (hue) en verzadiging (saturation), maar niet met de lichtheid. Toch kunnen de chromaticiteitscoördinaten verder worden gebruikt om de color correlated temperature (CCT) en CRI waarden te berekenen.



**Figuur 2.2:** Chromaticiteitsdiagram [12]

### 2.3.1 Correlated color temperature

Een andere belangrijke waarde bij de chromaticiteit is de CCT. De CCT is een maat voor de kleur van een lichtbron aan de hand van een temperatuur. Deze maat is gebaseerd op temperaturen bij "blackbody radiators". Een "blackbody radiator" is een object dat alle invallende straling absorbeert en vervolgens weer uitzendt.

Bij een "blackbody radiator" is de kleur volledig afhankelijk van de temperatuur. Hierdoor kunnen we de temperatuur van een "blackbody radiator" gebruiken om zijn kleur te beschrijven. Zo bevindt een witte "blackbody radiator" zich bijvoorbeeld rond 5000 K.

We kunnen de "Color temperature" linken met de chromaticiteit aan de hand van de curve in Figuur 2.2. Deze curve wordt de "blackbody locus" of blackbody curve genoemd en stelt de kleur voor van "blackbody radiators" bij verschillende temperaturen.

Wanneer de chromaticiteit van een lichtbron op de "blackbody locus" ligt, kan de "Color temperature" gegeven worden. Ligt de chromaticiteit van een lichtbron echter niet op de "blackbody locus", dan kan deze waarde niet worden gebruikt. In dit geval wordt de CCT waarde gebruikt, die de temperatuur van een "blackbody radiator" weergeeft die ongeveer dezelfde kleur heeft als de lichtbron.

## 2.4 Color rendering index

De CRI is een maat voor hoe goed een lichtbron kleuren kan weergeven [3, 13]. Het is een getal tussen 0 en 100, waarbij 100 de beste kleurweergave voorstelt. De CRI wordt berekend aan de hand van de kleurweergave van 8 of meer testkleuren.

De CRI waarde wordt bepaald door de kleursverandering van een testkleur onder de lichtbron te vergelijken met de kleursverandering van de testkleur onder een referentie lichtbron. Van elke van deze 8 of meer kleuren wordt de  $R_i$  waarde berekend, wat de kleurverandering van de testkleur onder de lichtbron aangeeft in vergelijking met de referentiebron. Aan de hand van deze  $R_i$  waarden wordt een gemiddelde berekend, de CRI waarde.

In de standaardmethode is de referentiebron een black body radiator wanneer de CCT van de lichtbron kleiner is dan 5000K. Is de CCT groter dan 5000K, dan wordt een mathematisch model van daglicht gebruikt als referentiebron. Naast de acht standaardkleuren kunnen er ook zes extra kleuren optioneel worden toegevoegd. Deze kleuren verbeteren de nauwkeurigheid van de CRI waarde.

Van deze extra kleuren is  $R_9$  een zeer belangrijke [14]. Lichtbronnen met een lage  $R_9$  waarde worden namelijk vaak als onaangenaam ervaren. Dit komt doordat de testkleur voor  $R_9$  een rode kleur is, die belangrijk is voor de weergave van bijvoorbeeld huidskleur. Veel lichtbronnen stralen minder licht uit bij de rode golflengtes, waardoor de  $R_9$  waarde vaak laag is. Dit is echter niet zichtbaar in de standaard CRI waarde.

De CRI waarde is belangrijk voor het beoordelen van de kwaliteit van een lichtbron, maar het is niet altijd de beste maatstaf. De CRI is niet in staat om lichtbronnen goed te rangschikken, vooral niet wanneer LEDs worden gebruikt. Dit komt door de dunne emissiespectra van LEDs, waarin het licht wordt gemengd. Voor deze reden worden nog andere methodes gebruikt/ontwikkeld om de kleurweergave van lichtbronnen te beoordelen. Het is belangrijk dat deze methodes worden gebruikt als een hulp bij het beoordelen van de kwaliteit van een lichtbron en niet als enige maatstaf.

### 2.4.1 Berekening CRI [15, 16]

In de standaardberekening van de CRI waarde wordt de "color rendering" van de testbron vergeleken met die van een referentiebron. Zoals eerder vermeld, wordt een black body radiator als referentiebron gebruikt wanneer de CCT van de testbron kleiner is dan 5000K. Is de CCT groter dan 5000K, dan wordt een mathematisch model van daglicht gebruikt.

De berekening begint met het bepalen van de x,y-coördinaten van de testbron. Deze coördinaten worden omgezet naar u,v-coördinaten. De u,v-coördinaten zijn een andere manier om kleuren te representeren en worden vaak gebruikt in kleurwetenschap vanwege hun perceptuele uniformiteit. De u,v-coördinaten worden vervolgens gebruikt om de CCT te berekenen door het dichtstbijzijnde punt op de Planckian locus te vinden. Aan de hand van deze CCT waarde wordt de referentiebron bepaald.

Daarna wordt de afstand van de testbron tot de Planckian locus (DC) berekend. Dit geeft aan of het resultaat relevant is. De CRI waarde is namelijk enkel gedefinieerd voor lichtbronnen die ongeveer wit zijn.

$$DC = \sqrt{(u - u_{\text{Planckian}})^2 + (v - v_{\text{Planckian}})^2}$$

Als de waarde van DC boven  $5.4 \times 10^{-3}$  ligt, is de CRI waarde niet relevant.

Vervolgens worden de testsamples belicht met zowel de testbron als de referentiebron. Aan de hand hiervan worden de  $u, v$ -coördinaten van het licht dat door de testsamples gereflecteerd wordt, bepaald. Daarna wordt op elk van deze samples een von Kries-transformatie toegepast. Dit is een transformatie die wordt uitgevoerd om rekening te houden met "chromatic adaptation".

"Chromatic adaptation" is het fenomeen waarbij de gevoeligheid van de ogen verandert om zich aan te passen aan veranderingen in de lichtbron [3, 17]. Dit fenomeen zorgt ervoor dat een object onder verschillende belichteningen grotendeels hetzelfde blijft. Zo zal een appel zelfs onder verschillende lichtbronnen nog steeds rood lijken.

De von Kries-transformatie wordt uitgevoerd aan de hand van de volgende formules:

$$c = \frac{4.0 - u - 10.0v}{v} \quad (2.1)$$

$$d = \frac{1.708v - 1.481u + 0.404}{v} \quad (2.2)$$

$$u_{c,i} = \frac{10.872 + 0.404 \left( \frac{c_r}{c_t} \right) c_{t,i} - 4 \left( \frac{d_r}{d_t} \right) d_{t,i}}{16.518 + 1.481 \left( \frac{c_r}{c_t} \right) c_{t,i} - \left( \frac{d_r}{d_t} \right) d_{t,i}} \quad (2.3)$$

$$v_{c,i} = \frac{5.520}{16.518 + 1.481 \left( \frac{c_r}{c_t} \right) c_{t,i} - \left( \frac{d_r}{d_t} \right) d_{t,i}} \quad (2.4)$$

Nu kan voor elk sample de euclidische afstand tussen de coördinaten van de testbron en de referentiebron berekend worden. Ten slotte kan de CRI waarde berekend worden aan de hand van de volgende formule:

$$R_i = 100 - 4.6\Delta E_i \quad (2.5)$$

Met  $\Delta E_i$  de euclidische afstand tussen de coördinaten van de testbron en de referentiebron voor de  $i$ -de sample.

Door het nemen van het gemiddelde van de  $R_i$  waarden kan de CRI waarde berekend worden. Het is echter natuurlijk ook belangrijk deze waarden apart te bekijken.

## 2.5 tm30<sup>1</sup>

Tm30 is een methode voor het evalueren van lichtbronnen die gebruik maakt van een statistische aanpak. Ze stelt zowel algemene eigenschappen (zoals de gamut area) als hue-specifieke eigenschappen (zoals chroma shift, hue shift, ...) voor. Deze methode probeert niet één specifieke waarde te geven die de lichtbron evalueert, maar biedt de gebruiker de benodigde kennis om een keuze te maken op basis van zijn of haar ervaring.

De methode maakt gebruik van 99 color evaluation samples (CES), die op statistische wijze werden gekozen uit 100.000 mogelijke CES. Om de kleur voor te stellen, maakt de methode gebruik van een groot aantal berekeningen (50). Afhankelijk van de situatie kunnen bepaalde van deze berekeningen worden gebruikt om een keuze te maken. De belangrijkste berekeningen zijn:

- Fidelity index ( $R_f$ )
- Gamut index ( $R_g$ )
- Color Vector Graphic (CVG)
- Local chroma shift
- Local hue shift

### 2.5.1 Fidelity index

De fidelity index is een waarde die aangeeft hoe goed een lichtbron de kleuren van een referentiebron kan weergeven.

De fidelity index van tm30 wordt berekend door de "CAM02-UCS" coördinaten van elke CES onder de testbron en de referentiebron te berekenen. Vervolgens wordt het rekenkundig gemiddelde genomen, geschaald met een factor 6.73 en afgetrokken van 100.

$$R'_f = 100 - 6.73 \left[ \sum_{i=1}^{919} 99 (\Delta E_{Jab,i}) \right] \quad (2.6)$$

Met  $\Delta E_{Jab,i}$  de euclidische afstand tussen de "CAM02-UCS" coördinaten van de testbron en de referentiebron voor de  $i$ -de CES.

Vervolgens wordt nog een aanpassing van de schaal gedaan zodat geen negatieve waarden bekomen worden.

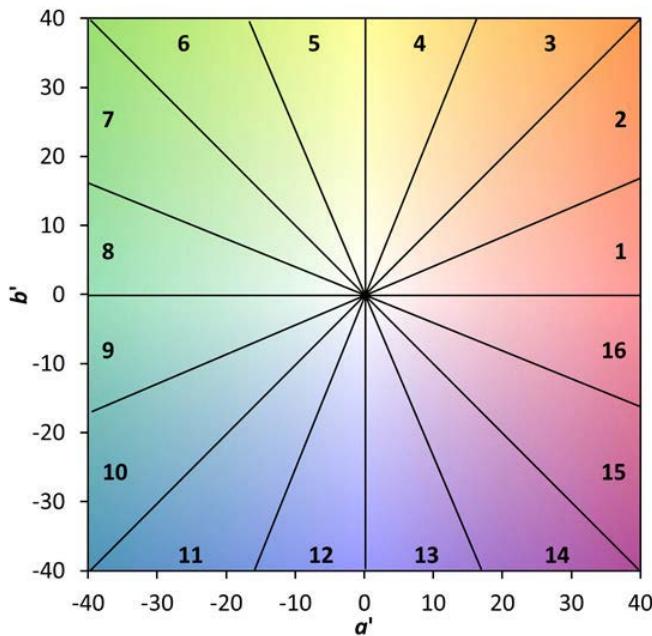
$$R_f = 10 \ln \left[ \exp \left( \frac{R'_f}{10} \right) + 1 \right] \quad (2.7)$$

---

<sup>1</sup>Gebaseerd op [18].

### 2.5.2 Hue-angle bins

Voor het berekenen van de volgende waarden worden de CES in 16 hue-angle bins verdeeld. Deze bins worden dan gebruikt om de volgende waarden te berekenen. De bins worden bepaald door de  $a'$ - $b'$  coördinaten van de CES in 16 stukken te verdelen volgens een radiaal patroon zoals in Figuur 2.3. Het  $a'$ - $b'$  coördinatenstelsel is een kleurruimte die de kleur van een lichtbron voorstelt. De  $a'$ -as staat voor de rood-groen as en de  $b'$ -as voor de geel-blauw as.



**Figuur 2.3:** Hue-angle bins

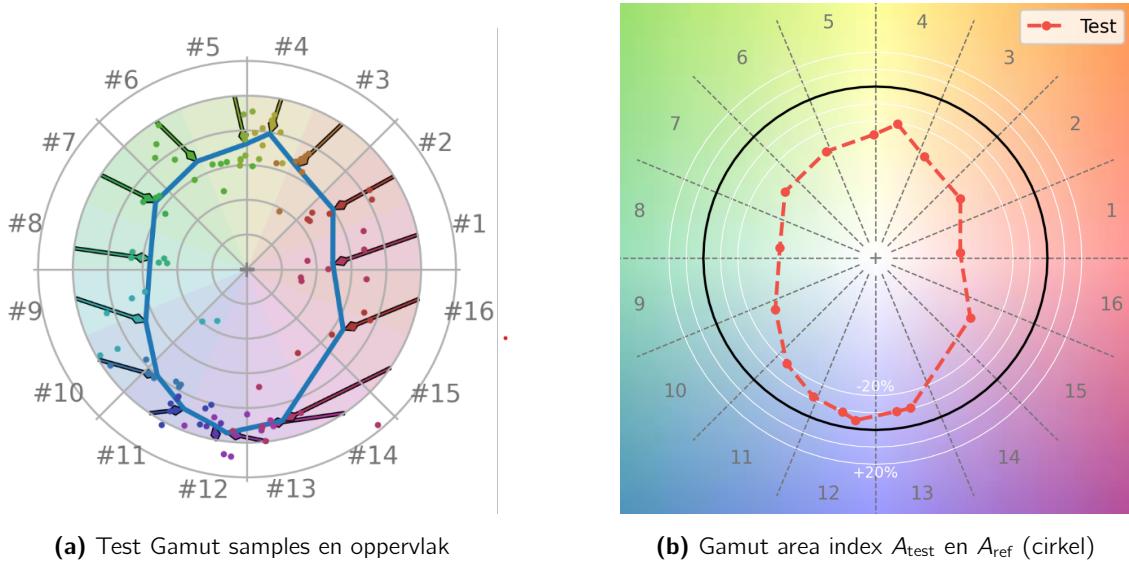
### 2.5.3 Gamut (area) index

De gamut (area) index is een waarde die het gebied aangeeft dat wordt beslagen door de gemiddelde ( $a'$ ,  $b'$ ) coördinaten van de CES in elke hue-angle bin. Voor de berekening wordt gewerkt met veelhoeken voor de test- en referentiebron. De gamut index wordt berekend aan de hand van Vergelijking (2.8) met  $A_{\text{test}}$  en  $A_{\text{ref}}$  als de oppervlakten van de veelhoeken van respectievelijk de test- en referentiebron zoals in Figuur 2.4b. [18, 19]

$$100 \times \frac{A_{\text{test}}}{A_{\text{ref}}} \quad (2.8)$$

Hiervoor wordt ten eerste voor alle samples het coördinaat van de test en referentiebron berekend zoals in Figuur 2.4a voor de testbron te zien is. Vervolgens wordt per bin het gemiddelde van alle test- en referentiecoördinaten genomen. Dit levert een veelhoek op voor

de test en de referentiebron zoals te zien voor de testbron in Figuur 2.4. Deze veelhoeken worden dan gebruikt om de oppervlakten te berekenen. Aan de hand hiervan kan dan met Vergelijking (2.8) de gamut (area) index berekend worden (Figuur 2.4b).



**Figuur 2.4:** Gamut samples (links) en gamut area index (rechts)

## 2.5.4 Color vector graphic

Een Color Vector Graphic (CVG) laat zien hoe een lichtbron kleuren beïnvloedt. De grafiek toont verschuivingen in kleurintensiteit (chroma) en kleurtint (hue) in een cirkelvormige weergave.

### 2.5.4.1 Chroma (kleurintensiteit)

- Als de lijn van de testlichtbron buiten de referentielijn ligt, worden de kleuren intenser (meer verzadigd).
- Ligt de lijn binnen de referentielijn, dan worden de kleuren minder intens (meer dof).

### Hue (kleurtint)

- Als een kleur verschuift langs de cirkel (zonder naar binnen of buiten te gaan), betekent dit dat de tint verandert.
- Bijvoorbeeld: een rode tint kan meer oranje worden of juist meer roze.

## 2.5.5 Local chroma shift

De local chroma shift beschrijft hoe de kleurverzadiging lokaal verandert.

- Een positieve waarde betekent dat de kleurverzadiging toeneemt (levendiger).
- Een negatieve waarde betekent dat de kleurverzadiging afneemt (doffer).
- Dit wordt berekend door de gemiddelde chroma van de testlichtbron te vergelijken met de gemiddelde chroma van de referentielichtbron in elk van de 16 hue-angle bins en wordt voorgesteld als een percentage.

### 2.5.6 Local hue shift

De local hue shift beschrijft hoe de kleurtint lokaal verandert.

- Een positieve waarde betekent dat de kleurtint verschuift naar een andere kleur tegen de klok in.
- Een negatieve waarde betekent dat de kleurtint verschuift naar een andere kleur met de klok mee.
- Dit wordt berekend door te kijken naar de tangentiële verplaatsing van de kleuren binnen elk van de 16 hue-angle bins.

## 2.6 Luminescente materialen

Het maken van wit licht vergt zeven kleuren (violet, blauw, groen, geel, oranje, rood en paars). Dit zou dus betekenen dat om wit licht met LEDs te maken er zeven LEDs moeten gebruikt worden.

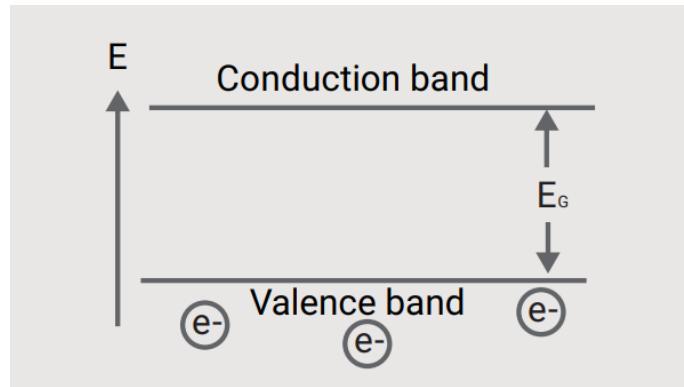
Er kan echter ook gebruik gemaakt worden van "Additive color mixing" om de perceptie van wit licht te creëren. Dit kan gedaan worden door rood, groen en blauw te mengen. Omdat het menselijk oog trichromatic is levert dit een wit licht op. Het grote probleem met het gebruiken van deze "mixing" technieken is dat er meerdere LEDs nodig zijn. Dit zorgt voor een grotere kost, het probleem ligt zich hier vooral bij de groene LED en de eerder vermelde "green gap" (Paragraaf 2.2).

Om dit probleem op te lossen kan gewerkt worden met luminescente materialen zoals fosforen. Hierbij kan een wit licht bekomen worden met slechts één LED. Dit bijvoorbeeld met een blauwe LED en een gele fosfor.

### 2.6.1 Werking luminescente materialen

In de meeste materialen wordt geabsorbeerd licht omgezet naar warmte. Bij luminescente materialen wordt een deel van dit licht echter ook omgezet naar licht met een hogere golf-lengte. In luminescente materialen resulteert de moleculaire structuur in energiebanden met elektronen. Hierbij is er een "valence" band waar de elektronen in een stabiele toestand zijn

en een "conduction" band waar de elektronen vrij kunnen bewegen zoals te zien in Figuur 2.5. De kloof tussen deze twee banden wordt bepaald door de configuratie van de atomen in het materiaal en kan door het toevoegen van onzuiverheden gemanipuleerd worden.



**Figuur 2.5:** Energiebanden [20]

Wanneer een luminescent materiaal belicht wordt zullen fotonen van het licht geabsorbeerd worden. Als de energie van de fotonen gelijk is aan het energieverlies tussen de twee banden zal een elektron van de "valence" naar de "conduction" band springen. De energie van een foton wordt bepaald door de volgende formule:

$$E = \frac{hc}{\lambda} \quad (2.9)$$

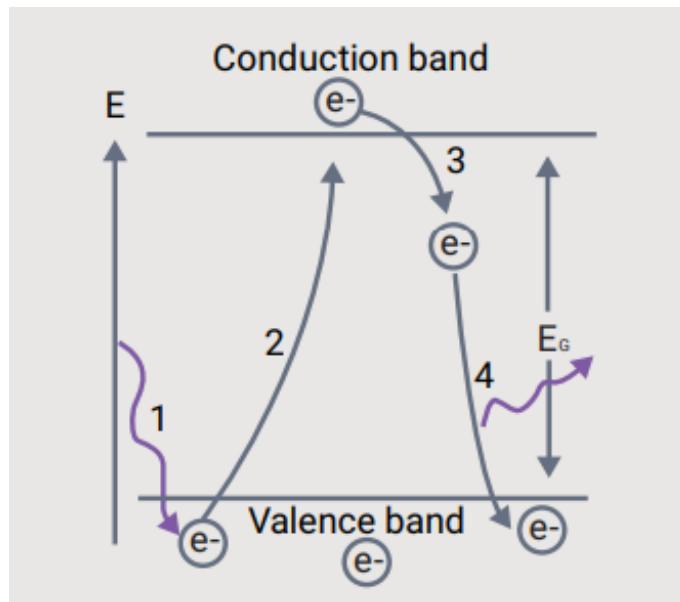
waarbij:

$h$  = Planck's constante

$c$  = de snelheid van het licht

$\lambda$  = de golflengte van het licht

Na een tijd zal deze elektron terug naar de "valence" band vallen. Dit gaat gepaart met het uitgeven van energie in de vorm van licht. In de realiteit zijn er naast deze twee banden nog andere subtoestanden die het toelaten dat een elektron kan verspringen met een lagere hoeveelheid energie. Dit bepaalt het absorptie spectrum van het materiaal.



**Figuur 2.6:** Proces luminescente materialen [20]

## 2.6.2 Witte LEDs

Om wit licht te bekomen moeten alle drie de cones in de retina van het oog geactiveerd zijn met een gelijkaardige intensiteit. Dit komt daarnaast overeen met een centrale positie in het chromatischeitsdiagram. Wit licht kan op een aantal verschillende manieren gegenereerd worden. Zoals eerder vermeld kan dit door kleuren te mengen. Dit kan dichromatic, trichromatic en tetrachromatisch (2, 3 of 4 LEDs). Op deze manieren kan een wit licht gegeneerd worden met een hoge CRI waarde. Er zijn echter veel toepassingen waar de CRI waarde niet de grootste prioriteit is.

In vele toepassingen is namelijk de efficiëntie van de lichtbron belangrijker. Om efficiënt een witte lichtbron te maken met LEDs wordt gebruik gemaakt van luminescente materialen. Aan de hand van deze materialen wordt het licht van een LED met een bepaalde centrale golflengte omgezet naar licht gecentreerd rond een andere golflengte. Omdat niet al het licht geconverteerd wordt kan zo een witte lichtbron bekomen worden door bijvoorbeeld blauw licht om te zetten naar geel licht.

Naast het feit dat niet al het licht van de LED geabsorbeerd wordt zijn er ook nog verliezen bij de omzetting van het licht. Er zijn ten eerste verliezen door de EQE van het converterende materiaal.

$$\eta_{\text{ext}} = \frac{\text{aantal fotonen uitgezonden in vrije ruimte door } \lambda\text{-converter per seconde}}{\text{aantal fotonen geabsorbeerd door } \lambda\text{-converter per seconde}} \quad (2.10)$$

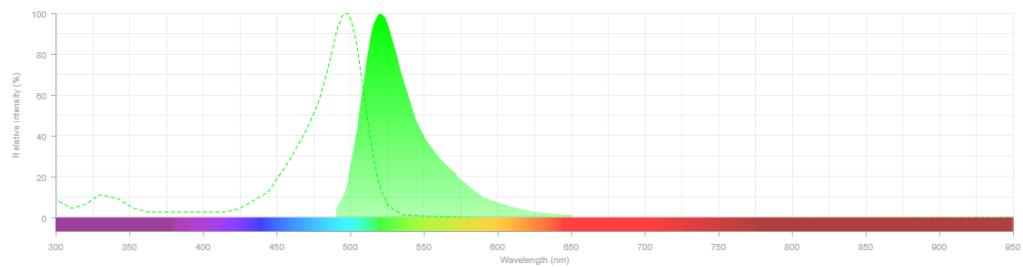
Ten tweede treden er ook verliezen op door de conversie naar een hogere golflengte. Dit verschijnsel wordt ook wel de stokes shift genoemd en treedt op bij het converteren van

een foton met een bepaalde golflengte naar een foton met een hogere golflengte. Deze verliezen trden op omdat de energie van een foton omgekeerd evenredig is met de golflengte (Vergelijking (2.9)). Dit betekent dat een foton met een hogere golflengte minder energie heeft dan een foton met een lagere golflengte. Dit verschil in energie wordt omgezet in warmte.

$$\Delta E = \frac{hc}{\lambda_1} - \frac{hc}{\lambda_2} \quad (2.11)$$

### 2.6.3 Verschillen tussen luminescente materialen

De verschillende luminescente materialen verschillen van elkaar in absorptie golflengtes, emissie golflengtes, eficiëntie, excitatie- en emissiespectra. Deze excitatie- en emissiespectra tonen aan hoeveel een luminescent materiaal elektronen naar een hoger niveau brengt en licht emiteert bij elke golflengte. Het is dan ook logisch dat in bepaalde toepassingen een smal emissiespectrum handiger kan zijn dan een breed emissiespectrum.



**Figuur 2.7:** Emissiespectra (rhodamine 110) [21]

# **Hoofdstuk 3**

## **Web applicatie**

In dit hoofdstuk leggen we uit hoe de verschillende concepten zijn geïmplementeerd. We beschrijven de gebruikte berekeningen, simulaties en overwegingen.

### **3.1 Architectuur**

Er zijn veel mogelijkheden voor het maken van een app. Talen zoals Python, C++, Java, ... kunnen gebruikt worden om een app te ontwikkelen die lokaal uitgevoerd kan worden. Daarnaast kunnen ook JavaScript, HTML en CSS, met eventueel frameworks, worden ingezet om een webapp te maken.

Omdat we een zo toegankelijk mogelijke app willen bieden, hebben we besloten om een webapp te ontwikkelen. Hierdoor kan de app door iedereen gebruikt worden zonder dat er een installatie nodig is.

#### **3.1.1 Frontend**

De webapp is geschreven in Vue.js, een Javascript-framework dat werkt met componenten en reactiviteit biedt. Deze reactiviteit zorgt ervoor dat de gebruiker direct de effecten van gemaakte aanpassingen kan zien. Andere vergelijkbare opties zijn React en Angular, maar we kozen voor Vue.js omdat het de efficiëntste van de drie is voor het maken van kleine, maar toch performante apps.

Voor de vele gebruikte standaardcomponenten (zoals knoppen, inputs, ...) maken we gebruik van het Vuetify Vue-componentframework.

We besloten om de frontend in twee talen aan te bieden: Nederlands en Engels. Dit wordt geïmplementeerd met I18n [22]. Via JSON-bestanden wordt voor elke taal de tekst die in de webapp gebruikt moet worden bijgehouden. De gebruiker kan eenvoudig veranderen van taal aan de hand van een knop in de app.

### 3.1.2 Backend

Het grootste deel van de berekeningen wordt in de webapp zelf uitgevoerd en dus op het apparaat van de client. Dit minimaliseert het aantal calls naar de server en de daarbijhorende laadtijden. We minimaliseren het aantal calls omdat we het belangrijk vinden dat de gebruiker het effect van zijn aanpassingen zo snel mogelijk ziet. Dit omdat we de gebruiker willen kunnen laten spelend leren. Als de gebruiker dan telkens vrij lang moet wachten op resultaten is dit niet mogelijk. Toch worden enkele complexere berekeningen naar de backend verplaatst. Dit komt omdat Javascript niet de meest geschikte taal is voor deze berekeningen en omdat dergelijke berekeningen het apparaat van de gebruiker te veel kunnen beladen. Toch proberen we er bij deze berekeningen ook voor te zorgen dat de tijd tot het krijgen van resultaten zo minimaal mogelijk is.

Om de complexe berekeningen uit te voeren maken we gebruik van een Python-backend met Flask. We kiezen voor Python vanwege de vele libraries voor datamanipulatie en wiskundige berekeningen, zoals numpy en scipy. Bovendien bestaat er een Python-library genaamd luxpy [23], die veel berekeningen voor licht en kleur al implementeert. Deze library wordt onder andere gebruikt voor het berekenen van de CRI en Ra-waardes, het genereren van een tm30-rapport en de berekeningen voor hyperspectral image rendering.

Omdat we willen dat de reactie van de backend zo rap mogelijk is maken we gebruik van websockets. Alternatief zouden we gebruik kunnen maken van HTPP, maar dit is niet even efficiënt. We kiezen echter voor websockets vanwege hun lagere latency.

## 3.2 Algemene structuur

We hebben de webapp opgedeeld in drie onderdelen. Ten eerste is er een onderdeel voor de inputs. Ten tweede is er een onderdeel voor de verschillende outputs. Ten slotte is er een onderdeel waarin de gebruiker de gebruikte modellen kan bekijken en aanpassen. We zorgen ervoor dat de inputs altijd zichtbaar zijn voor de gebruiker, omdat dit essentieel is. We willen namelijk dat de gebruiker altijd met de inputs kan spelen om te leren. Het is minder belangrijk dat alle verschillende outputs altijd zichtbaar zijn. Daarom hebben we ervoor gekozen om voor elk type output een tabblad te voorzien. Daarnaast voorzien we een laatste tabblad voor het databaseheer. Hier kan de gebruiker de gebruikte modellen bekijken en aanpassen.

### 3.2.1 Inputs

In het inputgedeelte past de gebruiker verschillende componenten aan met behulp van input fields en sliders, en ziet de effecten hiervan in het outputgedeelte. De gebruiker kan kiezen uit drie standaardtypes inputs: LEDs, fosforen en quantum dots. De gebruiker kan zes van deze componenten selecteren, bijvoorbeeld drie LEDs, twee fosforen en één quantum dot. We kozen ervoor zes inputs aan te bieden omdat dit voldoende speelruimte laat, maar ook

realistisch blijft.

### 3.2.2 Outputs

De output is opgesplits in zes verschillende tabbladen.

- SPD
- Chromaticiteit
- Color rendering
- "realistic rendering"
- led visualisatie
- databeheer

In het SPD-venster bekijkt de gebruiker de spectra van LEDs via een grafiek. Daarnaast ziet de gebruiker hoe verschillende luminescente materialen deze spectra beïnvloeden. Dit venster dient als initieel inzicht in de gemaakte LED.

In het chromaticiteitsvenster bekijkt de gebruiker de chromaticiteit van de LEDs en eventuele luminescente materialen. Dit biedt inzicht in de kleur van de LEDs en de invloed van luminescente materialen op die kleur. Dit is cruciaal voor het leren over het ontwerpen van witte LEDs. Dit venster dient voor de gebruiker als een eerste inzicht in de kleur van de gemaakte LED.

In het color-renderingvenster ziet de gebruiker de color rendering metrieken (CRI of tm30) van de LEDs en luminescente materialen. Hierdoor krijgt de gebruiker een beter beeld van hoe goed de LEDs kleuren kunnen weergeven, wat essentieel is voor de perceptie van de belichting. Dit venster dient dus voor de gebruiker als een inzicht in de kleurweergave van de gemaakte LED.

In het "realistic rendering"-venster krijgt de gebruiker een realistische weergave van de verlichting in een ruimte. De app toont hierbij standaard een vergelijking met D65 (daglicht) om een beter beeld te geven van hoe de verlichting er in werkelijkheid uit zou zien. Deze vergelijking kan ook aangepast worden naar een aantal andere belangrijke lichtbronnen. Dit venster dient dus voor de gebruiker om te leren over hoe de verlichting met de gemaakte LED er in werkelijkheid uit zou zien.

In het led-visualisatievenster krijgt de gebruiker een idee van hoe de chip van de gemaakte LED eruitziet. De visualisatie toont LEDs als gekleurde rechthoeken, fosforen als gekleurde kristallen en quantum dots als gekleurde puntjes. In dit venster kan de gebruiker op een eenvoudige manier leren over hoe een LED chip eruit zou kunnen zien.

In het "data management"-venster beheert de gebruiker de data van de app. Hier bekijkt de gebruiker welke modellen worden gebruikt voor de spectra van LEDs en luminescente

materialen. Daarnaast kan de gebruiker eigen luminescente materialen toevoegen, het referentiespectrum voor color rendering aanpassen en het type color rendering wijzigen. Tot slot biedt de app een methode om een eigen formule te gebruiken voor de standaard spectra van luminescente materialen en het model voor de LEDs.

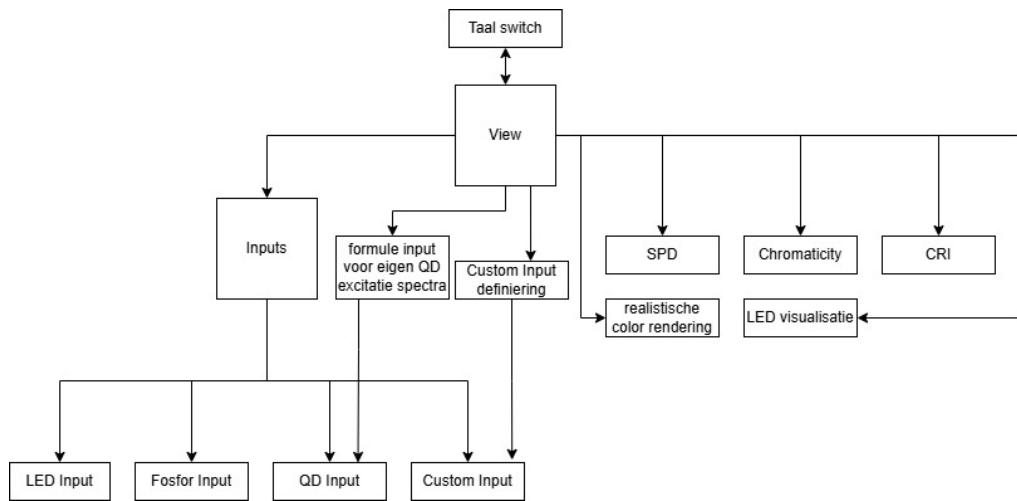
### 3.3 Structuur Vue

De webapp in Vue is opgebouwd met een structuur van componenten en views. De views vormen de verschillende pagina's van de app, terwijl de componenten de herbruikbare onderdelen zijn die binnen deze views gebruikt worden. We kozen ervoor slechts gebruik te maken van één view om de app zo eenvoudig mogelijk te houden.

Het werken met componenten biedt als voordeel dat ze herbruikbaar zijn. Zo gebruiken we bijvoorbeeld binnen het databaseheer tablad meerdere dezelfde componenten voor het tonen van modellen. Daarnaast kunnen componenten worden aangepast met "properties". Zo kan de gebruiker bijvoorbeeld zelf inputs definiëren, die vervolgens in de views worden toegepast.

De webapp bestaat uit een groot aantal componenten en één view. Omdat de app uit veel componenten bestaat, moeten we deze onderling kunnen laten communiceren. Zo moeten we bijvoorbeeld de CCT waarde die in het ene component berekend wordt versturen naar een ander component. We kunnen dit bijvoorbeeld doen aan de hand van "events", waarbij een component een event stuurt dat door een andere component wordt opgevangen.

Wanneer componenten die niet direct aan elkaar gekoppeld zijn data moeten delen, kan dit complex worden. Zo zou bijvoorbeeld als een component een event moet sturen naar een ver verbonden component een keten afgelegd moeten worden. Dit betekent dat één component een event ontvangt en dit moet doorsturen. Dit gaat zo verder tot de bestemming. Om dit te vereenvoudigen, maken we in de app gebruik van "pinia" stores. Deze globale structuren slaan data op en maken deze beschikbaar voor componenten, waardoor het delen van data tussen verschillende componenten eenvoudig blijft. Daarnaast kunnen we in pinia stores gebruik maken van de "persistent" state om caching toe te voegen. Het is echter niet altijd beter om stores te gebruiken, omdat dit de app complex kan maken. Daarnaast zijn stores ook minder efficient.

**Figuur 3.1:** Structuur Vue app

### 3.4 Implementatie inputs

De inputs vormen het belangrijkste onderdeel van de app. Via deze inputs zorgen we ervoor dat de gebruiker kan experimenteren en leren van de effecten van zijn aanpassingen. De app biedt, zoals eerder vermeld, drie standaardtypes inputs: inputs voor LEDs, fosforen en quantum dots.

Naast deze standaardinputs kan de gebruiker ook een input voor een luminescent materiaal maken met een csv-bestand en enkele scaling parameters. We bieden de mogelijkheid om eigen inputs te maken voor niet lerende gebruikers om te kunnen experimenteren. Zo kan bijvoorbeeld iemand die met een specifiek luminescent materiaal wil experimenteren dit doen. Deze input biedt echter wel minder flexibiliteit dan de standaardinputs, aangezien alleen de scaling parameters en de Photoluminescence quantum yield (PLQY)-waarde aangepast kunnen worden, dit omdat we bij deze input gebruik maken van vaste spectra en geen aanpasbare modellen.

Voor het werken met de inputs kiest de gebruiker via een selectiemenu. Hierbij maken we gebruik van de reactiviteit van vue om enkel de geselecteerde inputs te tonen. We doen dit aan de hand van conditional rendering, waarbij alle componenten wel ingeladen zijn, maar enkel de geselecteerde effectief getoond worden.

We slaan de data van de inputs op in een van de eerder vermelde "pinia" stores. We maken gebruik van deze store omdat we de data van de inputs moeten gebruiken in verschillende componenten. Zo wordt bijvoorbeeld de data gebruikt om de stralingsstroom te berekenen, maar ook om de chromaticiteit te berekenen.

### 3.4.1 LED Input

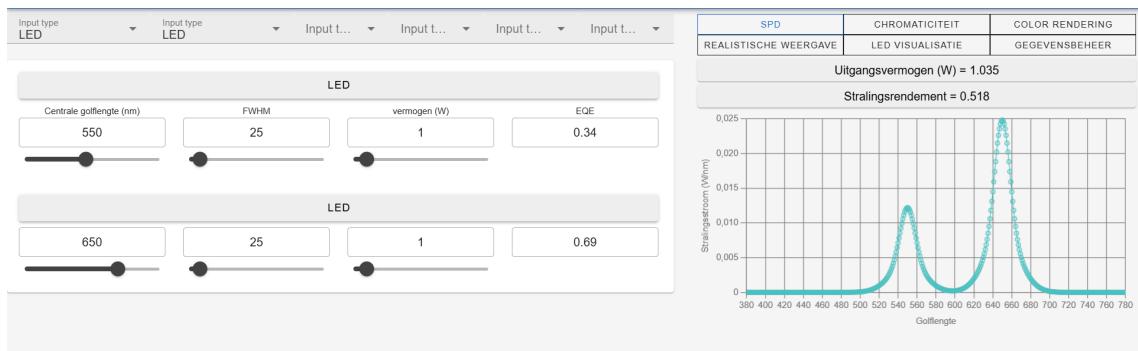
Om een LED toe te voegen, moet de gebruiker een LED selecteren uit de beschikbare inputtypes. Vervolgens kan de gebruiker de parameters waarmee hij/zij wil experimenteren aanpassen met behulp van input fields en sliders. We kozen ervoor de fields en sliders te gebruiken omdat dit de gebruiker zowel toelaat specifieke waarden in te geven en met waarden te spelen.

We kozen ervoor om de LED input te beperken tot de belangrijkste parameters.

- Centrale golflengte
- full width at half maximum (FWHM)
- Ingangsvermogen
- EQE

We kozen voor deze parameters omdat deze de belangrijkste zijn voor het maken van een LED. De centrale golflengte en FWHM bepalen de kleur van de LED, het ingangsvermogen bepaalt de lichtstroom en de EQE bepaalt hoe efficiënt de LED is in het omzetten van elektrische energie naar licht. Dit zijn dan ook de parameters die het belangrijkste zijn om al spelend te leren over de eigenschappen van LEDs.

We kozen ervoor de EQE-waarde onaanpasbaar te laten omdat deze waarde uit een model komt. De gebruiker kan echter wel de EQE-waarde aanpassen door een eigen model te gebruiken (Paragraaf 3.5.3).



**Figuur 3.2:** Voorbeeld LED inputs

### 3.4.2 Fosfor input

Net zoals bij de LED input, moet de gebruiker voor het werken met fosforen een fosfor selecteren uit de beschikbare inputtypes. Hierbij voorzien we opnieuw input fields en sliders om de parameters specifiek en spelend aan te kunnen passen.

We kozen ervoor de volgende parameters aan de gebruiker aan te bieden.

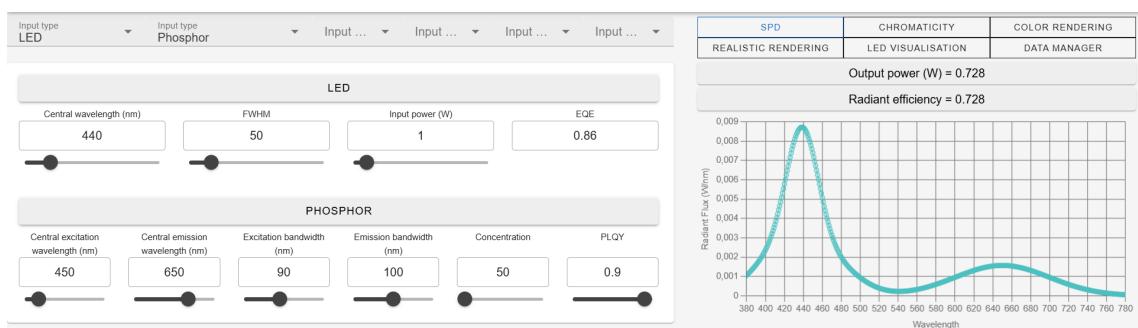
- Centrale excitatiegolfelngte
- Centrale emissiegolfelngte
- Excitatiebandbreedte
- Emissiebandbreedte
- concentratie
- PLQY

Deze parameters zijn de belangrijkste voor het maken van een fosfor. De centrale golfelngtes en bandbreedtes bepalen de kleur van de fosfor, de concentratie bepaalt de hoeveelheid energie die de fosfor kan absorberen en de PLQY bepaalt hoe efficiënt de fosfor is in het omzetten van geabsorbeerde energie naar licht. Daarnaast kan de gebruiker aan de hand van de centrale golfelngtes leren over de gevolgen van de stokes shift (Paragraaf 2.6.2).

We kozen ervoor om bij het selecteren van een fosfor de excitatiebandbreedte, emissiebandbreedte en PLQY standaardwaarden te geven.

- Excitatiebandbreedte: 90 nm
- Emissiebandbreedte: 100 nm
- PLQY: 0.9

Deze waarden zijn standaardwaarden die een realistisch beeld geven van een gewoonlijk fosfor.



**Figuur 3.3:** Voorbeeld fosfor inputs

### 3.4.3 Quantum dot input

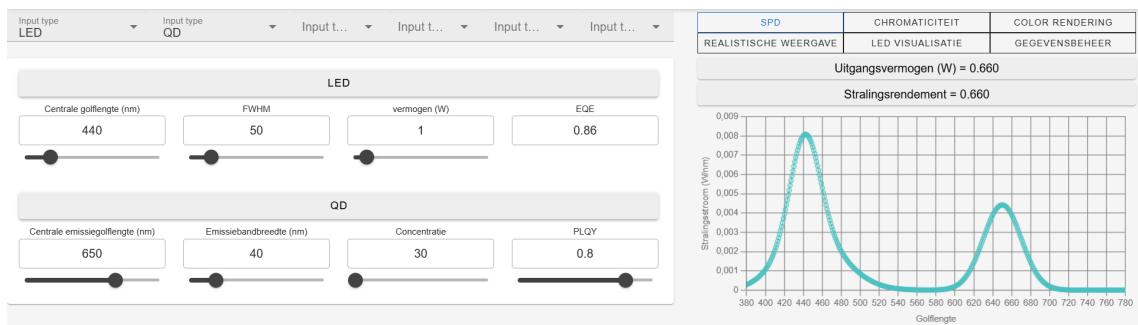
De quantum dot input lijkt op de fosfor input. De gebruiker past, net zoals bij de fosfor input, de centrale emissiegolfelngte, emissiebandbreedte, concentratie, en PLQY-waarden aan. De parameters voor de excitatie blijven echter onveranderd, omdat het standaardmodel voor het

excitatiespectrum geen centrale excitatiegolfleugte gebruikt. Wanneer echter een aangepaste formule wordt gebruikt, kan de gebruiker deze waarden wel aanpassen.

Voor de inputparameters die overeenkomen met de fosfor input gebruiken we achterliggend dezelfde variabelen. We doen dit op deze manier om zo het aantal parameters in de "pinia" stores zo gelimiteerd mogelijk te houden. Zo zorgen we ervoor dat de app zo efficiënt mogelijk blijft.

Net zoals bij de fosfor input, voorzien we standaardwaarden voor de emissiebandbreedte en PLQY.

- Emissiebandbreedte: 40 nm
- PLQY: 0.8

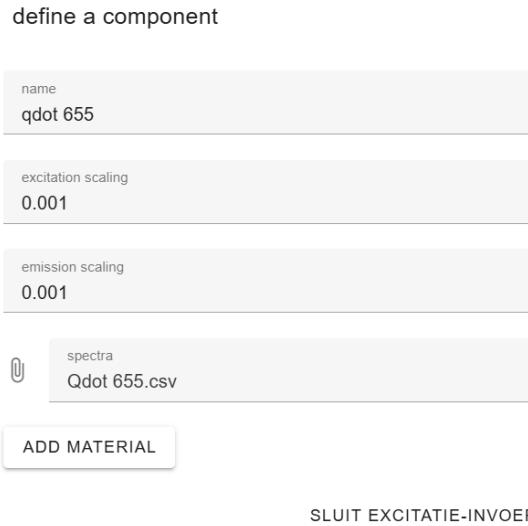


**Figuur 3.4:** Voorbeeld quantum dot inputs

### 3.4.4 Luminescent materiaal input

De gebruiker kan naast de standaard luminescente materialen ook eigen luminescente materialen toevoegen. Dit doet hij door een comma-separated values (CSV)-bestand te uploaden met de kolommen "Wavelength", "Excitation" en "Emission".

Daarnaast stelt de gebruiker een initiële schalingsfactor in en geeft hij een naam aan het materiaal. De schalingsfactor zorgt ervoor dat de geabsorbeerde en geëmitteerde energie logisch samenwerkt met de concentratieschalingsfactor.



**Figuur 3.5:** Voorbeeld upload luminescent materiaal

De naam van het materiaal wordt gebruikt om alle nodige informatie van het materiaal op te slaan in een store voor de toegevoegde materialen. De data wordt opgeslagen in een object waarbij de keys de naam van het materiaal zijn en de values een object bevatten voor de excitatie- en emissiespectra. Daarnaast worden ook schalingswaarden opgeslagen. Deze schalingswaarden worden voorzien omdat we niet kunnen voorspellen hoe de waarden van de gebruiker zullen zijn. De schalingswaarden zorgen ervoor dat de gebruiker de data kan aanpassen zodat realistische waarden bekomen worden in de simulaties.

We kunnen de gebruiker van deze inputs een stuk minder laten aanpassen omdat bepaalde zaken zoals de golflengtes inherent zijn aan het materiaal. We laten de gebruiker wel nog toe om de volgende parameters aan te passen.

- Schalingsfactoren
- concentratie
- PLQY

Omdat deze elementen ook geselecteerd moeten kunnen worden, hebben we componenten gemaakt die op basis van de geselecteerde waarde informatie ophalen uit de store en tonen aan de gebruiker. Hierbij moeten we rekening houden met de mogelijkheid dat de gebruiker een materiaal selecteert dat niet bestaat. In dat geval mag het component niet getoond worden. Dit kan eenvoudig gecontroleerd worden door te verifiëren of de naam behoort tot de beschikbare items en niet gelijk is aan LED, fosfor of QD.



**Figuur 3.6:** Voorbeeld luminescent materiaal input

## 3.5 Implementatie LEDs en SPD

### 3.5.1 Implementatie LED model

Het belangrijkste onderdeel van de app dat we implementeerden, is het model van de LED en de omzetting ervan naar de SPD. We starten met een model of spectrum dat het relatieve vermogen per golflengte weergeeft. Op basis hiervan zetten we dit om naar de SPD van de LED.

Oorspronkelijk gebruikten we een gaussiaans model met het centrum op de centrale golflengte van de LED en een breedte bepaald door het FWHM. Er zijn echter meerdere manieren om een LED te modelleren. Zo zouden we bijvoorbeeld ook een lorentziaans model kunnen gebruiken. Beide van deze modellen zijn echter relatief eenvoudig en geven geen volledig accuraat beeld van een LED.

Om de simulaties accurater te maken, schakelden we over naar een wiskundig model dat specifiek ontwikkeld werd voor multichip LEDs [24]. Dit model wordt, net als het gaussiaanse model, bepaald door de centrale golflengte en het FWHM van de LED. Het model bepaalt de SPD van de LED aan de hand van de volgende formule:

$$S_{\text{LED}}(\lambda, \lambda_0, \Delta\lambda_{0.5}) = \frac{g(\lambda, \lambda_0, \Delta\lambda_{0.5}) + 2g^5(\lambda, \lambda_0, \Delta\lambda_{0.5})}{3}, \quad (3.1)$$

waarin  $g$  gelijk is aan

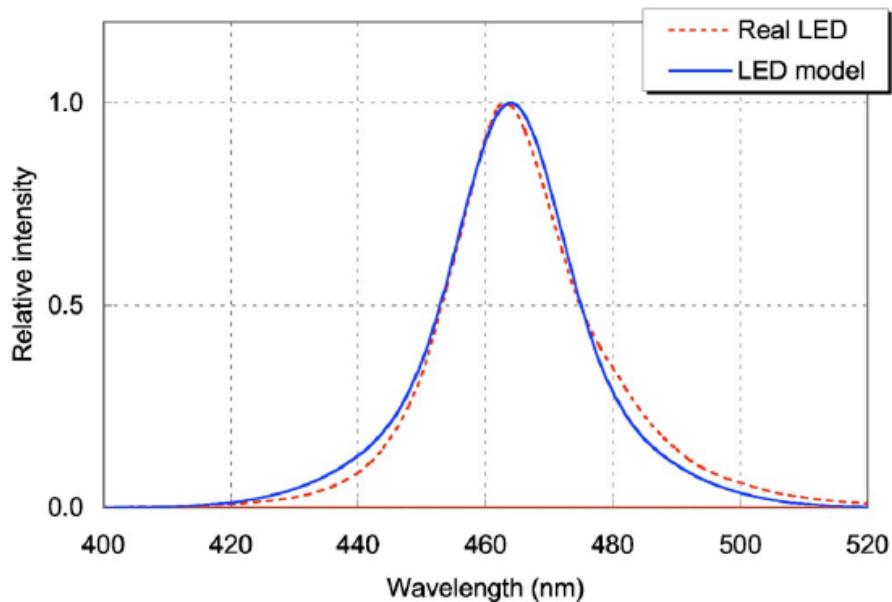
$$g(\lambda, \lambda_0, \Delta\lambda_{0.5}) = \exp\left(-\left[\frac{\lambda - \lambda_0}{\Delta\lambda_{0.5}}\right]^2\right). \quad (3.2)$$

met symbolen:

- $\lambda$ : De golflengte parameter (nm).
- $\lambda_0$ : De central golflengte van de LED emissie, dus de golflengte waarop het relatief vermogen het hoogste is.

- $\Delta\lambda_{0.5}$ : De FWHM, stelt de spectrale breedte van de LED voor.

Omdat dit model een relatief model is, waarbij de hoogste waarde 1 is en de laagste 0, passen we een normalisatie toe om verdere berekeningen correct te laten verlopen. We normaliseren eenvoudigweg door de som van alle datapunten als normalisatiefactor te gebruiken. Door vervolgens elk datapunt door deze factor te delen, bekomen we een model met een totaal standaard vermogen van één Watt. Omdat deze factor verschilt afhankelijk van de ingestelde parameters moeten we dit telkens opnieuw berekenen. Dit zorgt voor een lichte vertraging, we kiezen ervoor toch dit model te blijven gebruiken omdat de verhoogde accuraatheid belangrijker is dan de kleine vertraging. Dankzij deze aanpassing kunnen we het vermogen van een LED eenvoudig instellen door te vermenigvuldigen met het ingestelde vermogen. Daarnaast kunnen we de EQE op dezelfde manier direct vermenigvuldigen.



**Figuur 3.7:** Multichip led model (Ohno) [24]

### 3.5.2 Implementatie SPD

De overgang van een LED-model naar de SPD is vrij eenvoudig. Ten eerste maken we voor elke aanwezige LED de overgang van een relatief vermogen naar een effectief vermogen per nanometer. Door de eerder toegepaste normalisatie kunnen we dit doen door het relatieve vermogen van het model te vermenigvuldigen met het door de gebruiker ingestelde vermogen. Ten slotte vermenigvuldigen we ook de EQE van de LED. Door deze vermenigvuldigingen krijgen we een realistische simulatie van de SPD van een LED. Wanneer een gebruiker meerdere LEDs toevoegt, tellen we eenvoudigweg de verschillende bekomen SPD-waarden op om de totale SPD te verkrijgen.

Vervolgens tonen we het bekomen SPD-model aan de gebruiker in een grafiek. Hierin tonen

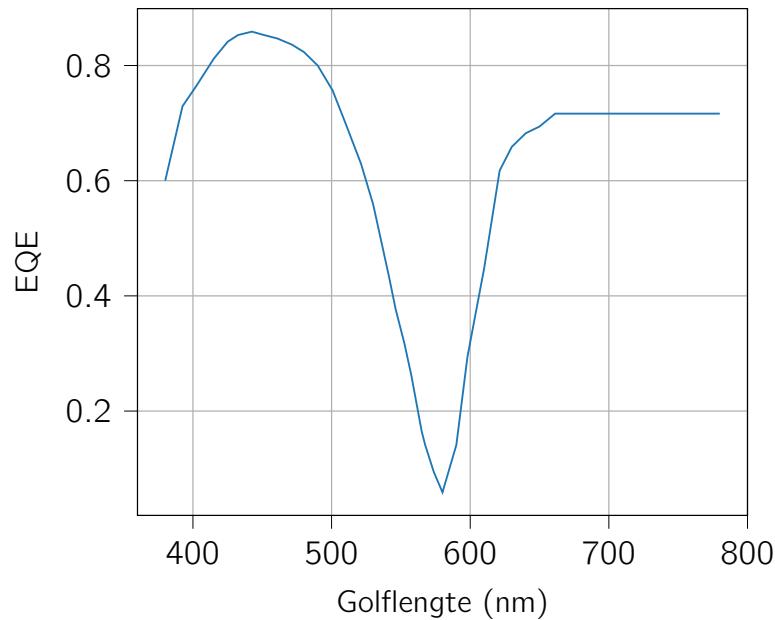
we de stralingsstroom tussen 380 en 780 nm (het zichtbare spectrum). Naast deze grafiek geven we ook het uitgaande vermogen en de "radiant efficiency" (uitgaand vermogen / ingaand vermogen) weer. Dit helpt de gebruiker te begrijpen welke effecten verschillende technieken hebben op het genereren van witte LEDs. Zo kan de gebruiker bijvoorbeeld zien waarom het combineren van rode, groene en blauwe LEDs niet noodzakelijk even efficiënt is als een blauwe LED met een gele fosfor.

Met deze grafiek kan de gebruiker zich al een eerste idee vormen over de eigenschappen van de gemaakte LED. Dit gebeurt aan de hand van de golflengtes van de pieken, de breedte van de pieken en de relatieve intensiteit van de pieken.

### 3.5.3 Model EQE

Omdat de berekening van de SPD gebruik maakt van de EQE van een LED, is er ook een model voor nodig. Een realistisch model van EQE is vooral belangrijk vanwege de "green gap". Dit is een kloof in de EQE-waarden van LEDs, die zich vooral bevindt bij de golflengtes die geassocieerd worden met groene LEDs. Dit verklaart waarom het maken van witte LEDs door simpelweg LEDs te combineren momenteel inefficiënt is. Het EQE-model leiden we af uit meetresultaten van LEDs uit 2017 en 2018 [8].

Omdat het echter niet zeker is dat deze "green gap" blijft bestaan of even ernstig blijft, bieden we de gebruiker de mogelijkheid om een CSV-bestand met EQE-waarden te uploaden. Dit bestand hoeft niet per se realistische waarden te bevatten. Zo kan het bijvoorbeeld ook worden gebruikt om te experimenteren met scenario's waarin de "green gap" niet bestaat of niet langer een groot probleem is. Dit maakt de app direct flexibeler en beter bestand tegen mogelijke veranderingen in de toekomst. Daarnaast laten we zo ook lerende gebruikers toe om te experimenteren met verschillende scenario's. Hierdoor kunnen ze beter begrijpen waarom de "green gap" een probleem is en hoe dit kan worden opgelost.

**Figuur 3.8:** Model EQE [8]

### 3.6 Implementatie chromaticiteit

Voor het implementeren van de chromaticiteit maken we gebruik van de in Paragraaf 2.3 beschreven formules. Er moet echter een kleine aanpassing worden gemaakt. We moeten namelijk overgaan van integralen naar sommen. Dit omdat de SPD van een LED een discrete functie is (één punt per nm). De chromaticiteit kan dan berekend worden aan de hand van de volgende formules voor de tristimuluswaarden:

$$X = 683 \cdot \sum_{\lambda} \bar{x}(\lambda) S(\lambda) \Delta \lambda \quad (3.3)$$

$$Y = 683 \cdot \sum_{\lambda} \bar{y}(\lambda) S(\lambda) \Delta \lambda \quad (3.4)$$

$$Z = 683 \cdot \sum_{\lambda} \bar{z}(\lambda) S(\lambda) \Delta \lambda \quad (3.5)$$

waarin:

- $\bar{x}(\lambda)$ ,  $\bar{y}(\lambda)$  en  $\bar{z}(\lambda)$ : de color matching functies.
- $S(\lambda)$ : de SPD van de LED.
- $\Delta \lambda$ : 1 nm.

De chromaticiteit kan dan berekend worden aan de hand van de volgende formules:

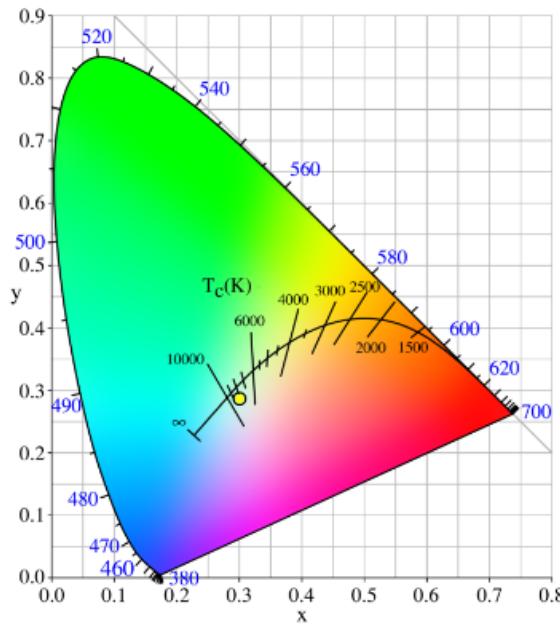
$$x = \frac{X}{X+Y+Z} \quad (3.6)$$

$$y = \frac{Y}{X+Y+Z} \quad (3.7)$$

$$z = \frac{Z}{X+Y+Z} \quad (3.8)$$

Voor de berekening van de chromaticiteit maken we gebruik van de color matching functies. Om deze te gebruiken, laden we een CSV-bestand in met de waarden van de color matching functies (CIE 1931 2° observer). Deze waarden worden vervolgens gebruikt om de chromaticiteit te berekenen.

Om het berekende ( $x, y$ )-punt op het chromaticiteitsdiagram te tonen, gebruiken we een canvas-element (HTML element voor tekenen) met een foto van een standaard chromaticiteitsdiagram. We tekenen het punt op dit diagram door de ( $x, y$ )-coördinaten te transformeren. Daarnaast voegen we de CCT-waarde toe, omdat het moeilijk voor de gebruiker is om deze waarde direct af te lezen van het diagram.



**Figuur 3.9:** Voorbeeld resultaat chromaticiteit

Het doel van de chromaticiteit is om de gebruiker een eerste idee te geven van de kleur van zijn gemaakte LED. Dit kan bijvoorbeeld de gebruiker helpen om een witte LED te maken. Daarnaast leert het de gebruiker ook dat chromaticiteit geen volledige beschrijving is van de kleur van een lichtbron. Zo kan een LED met dezelfde chromaticiteit toch een andere ervaring hebben voor een toeschouwer.

Naast de chromaticiteit worden ook de totale lichtstroom, het visueel rendement (totale lichtstroom / uitgaand vermogen) en de specifieke lichtstroom (totale lichtstroom / ingaand

vermogen) getoond. De totale lichtstroom geeft de gebruiker een idee van het licht dat de LED uitstraalt. Het visueel rendement geeft de gebruiker inzicht in hoe efficiënt de LED is in het omzetten van elektrische energie naar licht. De specifieke lichtstroom geeft de gebruiker een idee van hoeveel licht er wordt uitgestraald per watt elektrische energie. Deze waarden zijn belangrijk voor de gebruiker en kunnen bijvoorbeeld bepalend zijn bij het kiezen van een bepaalde manier om een kleur te verkrijgen (meerdere LED of luminescente materialen). Door ze hier toe te voegen krijgt de gebruiker dus een beter beeld van de eigenschappen van de gemaakte LED. Zo kan hij/zij bijvoorbeeld een LED maken met een bepaalde chromaticiteit, maar met een laag visueel rendement. Hierna kan dan bijvoorbeeld een LED bekomen worden met dezelfde chromaticiteit, maar met een hoger visueel rendement. Hierdoor leert de gebruiker over de voor en nadelen van de verschillende methoden om een kleur te bekomen.

## 3.7 Implementatie color rendering

### 3.7.1 CIE Ra-15

Het berekenen van de CRI en Ra-waarden is dankzij de Luxpy-bibliotheek zeer eenvoudig [23]. De gebruiker hoeft enkel een SPD mee te geven, waarna de R0-R9-waarden berekend kunnen worden. Luxpy berekent automatisch het referentiespectrum, maar omdat we dit ook aan de gebruiker willen tonen, zodat deze kan proberen een (volgens CRI) zo goed mogelijke lichtbron te verkrijgen, hebben we de Luxpy-functie licht aangepast. Het referentiespectrum wordt door deze aanpassing teruggegeven na een request aan de server. De CRI- en Ra-waarden worden vervolgens berekend en getoond aan de gebruiker in een staafdiagram.

We tonen de gebruiker de R0-R9-waarden, de CRI-waarde, het referentiespectrum en het eigen gemaakte spectrum. Aan de hand van de R-waarden kan de gebruiker zien welke kleuren goed worden weergegeven en welke niet. De CRI-waarde geeft de gebruiker een idee van hoe goed de kleuren over het algemeen worden weergegeven. Het referentiespectrum toont de gebruiker hoe een perfecte lichtbron eruitziet (volgens CIE Ra-15). We tonen al deze waarden zodat de gebruiker kan leren over de effecten van verschillende lichtbronnen op de kleurweergave.

Het is belangrijk dat de gebruiker leert hoe hij een lichtbron kan maken die volgens de CRI-waarde zo goed mogelijk is. Het is echter cruciaal dat de gebruiker begrijpt dat een hoge waarde niet noodzakelijk betekent dat de lichtbron effectief goed is. Zo kan bijvoorbeeld een score boven de 80 behaald worden, terwijl de waarde van R9 slechts rond de 20 ligt. Volgens CRI is dit een goede lichtbron, maar voor de mens is de R9-waarde erg belangrijk. In dit geval is de lichtbron waarschijnlijk niet ideaal. Hiervoor worden naast CIE Ra-15 ook andere methoden zoals IES TM-30 en "realistic color rendering" aangeboden.

### 3.7.2 ies-tm30

Naast CIE Ra-15 biedt Luxpy [23] ook de mogelijkheid om IES TM-30-berekeningen uit te voeren en rapporten te genereren. Omdat dit een recentere en officiële color rendering maat is, kozen we ervoor de gebruiker de mogelijkheid te geven om IES TM-30 te gebruiken in plaats van CIE Ra-15. Daarnaast is het ook een goede manier om te leren over de verschillen tussen verschillende maten voor kleurweergave. Een meer geavanceerde gebruiker kan dan weer kiezen welke color rendering methode het beste past bij zijn of haar toepassing. Net als bij CIE Ra-15 kunnen de benodigde berekeningen uitgevoerd worden met Luxpy.

We kozen ervoor om een eenvoudig TM-30-verslag te gebruiken met Rf, Rg, CCT en D(u,v), omdat dit de belangrijkste waarden binnen de TM-30 methode zijn. Daarnaast tonen we een gamutcirkel met daarop hue shift vectoren. Aan de hand van deze waarden kan een lerende gebruiker leren over het belang van de verschillende metrieken. Een geavanceerde gebruiker kan zich een idee vormen over de geschiktheid van zijn gemaakte lichtbron voor een bepaalde toepassing.

## 3.8 Implementatie luminescente materialen

Om de invloed van luminescente materialen te begrijpen, hebben we een berekening nodig die van de excitatie- en absorptiespectra naar de geabsorbeerde en geëmitteerde energie leidt. Zoals eerder vermeld, zijn er geen applicaties die dit reeds hebben gedaan. Bovendien zijn er geen formules beschikbaar die dit op een eenvoudige en algemene manier kunnen berekenen. Daarom besloten we zelf een formule te ontwikkelen.

### 3.8.1 Simple berekening

oorspronkelijk maakten we gebruik van een simpele berekening die ook in de KU Leuven Excel werd gebruikt. Deze werkt echter alleen volledig correct voor fosforen. Daarnaast is deze niet volledig theoretisch onderbouwd, maar volgt deze de verwachte effecten van fosforen (zoals de Stokes shift). De geabsorbeerde energie wordt in deze methode berekend aan de hand van een exponentieel verval.

$$E_{abs} = S \cdot (1 - \exp(-C \cdot 3 \cdot \sqrt{2 \cdot \pi} \cdot \text{norm}(\lambda, \lambda_{ex}, \text{excitatiebreedte}/2))) \quad (3.9)$$

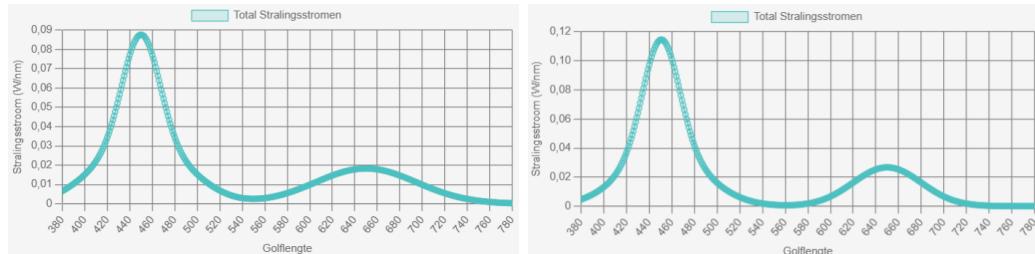
waarbij C de concentratie is en S de stralingsstroom.

Omdat deze formule gebaseerd is op het gebruik van de normale verdeling binnen het exponentiële verval, is deze alleen correct voor fosforen. Voor quantum dots en andere materialen waarbij de excitatiespectra geen normale verdeling volgen, is deze formule minder geschikt.

Om het geëmitteerde vermogen te berekenen, maken we gebruik van een simpele formule waarin we rekening houden met de verschillen in concentraties tussen de verschillende aanwezige materialen en de Stokes shift.

$$E_{em} = E_{abs}.QE \cdot \text{emissiespectrum}.C.(1 - (\lambda_{em} - \lambda_{ex}/\lambda_{ex})) \quad (3.10)$$

waarbij C het percentage van alle luminescente materialen is dat deze inneemt.



**Figuur 3.10:** Voorbeeld resultaten met simpel model (links = fosfor, rechts = quantum dot)

### 3.8.2 Geavanceerde berekening

Om een correctere berekening te maken zodat een lerende gebruiker de effecten van luminescente materialen correct kan leren kennen ontwikkelden we een geavanceerd model. We ontwikkelden dit geavanceerd model op een meer theorethische manier zodat alle effecten van de luminescente materialen correct gemodelleerd zijn. Deze formule baseren we op de verdeling van fotonen en de verdeling van energie. Daarnaast maken we gebruik van de relaties tussen excitatie- en emissiespectra en de geabsorbeerde en geëmiteerde energie. Deze formule kan gebruikt worden voor elk luminescent materiaal waarvan de excitatie- en emissiespectra gekend zijn en is dus meer geschikt voor de toepassing dan de simpele formule. We weten dat

$$f_1(\lambda) = \text{De distributie van geabsorbeerde energie en} \quad (3.11)$$

$$f_2(\lambda) = \text{De distributie van geëmiteerde energie.} \quad (3.12)$$

Hierbij weten we dat

$$f_1(\lambda) = f'_1(\lambda) \cdot SPD(\lambda) \quad (3.13)$$

met  $f'_1(\lambda)$  het excitatiespectrum van het gebruikte luminescente materiaal.

Nu moet echter nog  $f_2(\lambda)$  berekend worden. We weten dat de distributie van het aantal fotonen evenredig is met

$$f(\lambda) \cdot \lambda = g(\lambda) \quad (\text{de verdeling van de fotonen}) \quad (3.14)$$

want

$$E(\lambda) \sim 1/\lambda \quad (\text{de energie van een foton}) \quad (3.15)$$

Als we veronderstellen dat PLQY = 100% (het aantal geabsorbeerde fotonen/het aantal geëmitteerde fotonen van een materiaal). Dan weten we dat het aantal fotonen gelijk blijft en kunnen we stellen dat

$$\int g_1(\lambda) d\lambda = \int g_2(\lambda) d\lambda \quad (3.16)$$

Met  $g(\lambda)$  de verdeling van de fotonen.

uit 3.14 volgt dan dat

$$\int f_1(\lambda) \cdot \lambda d\lambda = \int f_2(\lambda) \cdot \lambda d\lambda \quad (3.17)$$

Met

Omdat  $f_1(\lambda)$  gekend is en we de vorm van  $f_2(\lambda)$  kennen. Deze heeft namelijk dezelfde vorm als het emissiespectrum  $f'_2(\lambda)$ . We kennen dus  $f_2(\lambda)$  op een schalingsfactor na dus

$$f_2(\lambda) = a \cdot f'_2(\lambda) \quad (3.18)$$

Met  $f'_2(\lambda)$  het emissiespectrum van het luminescente materiaal en  $a$  de schalingsfactor.

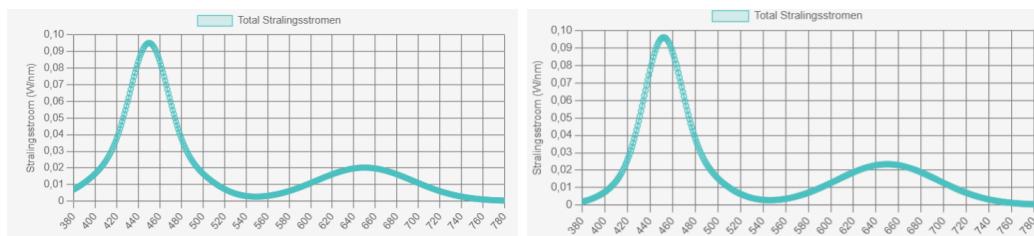
Dan kunnen we  $a$  op de volgende manier vinden

$$\int f_1(\lambda) \cdot \lambda d\lambda = \int a \cdot f'_2(\lambda) \cdot \lambda d\lambda \quad (3.19)$$

$$\Leftrightarrow a = \int f_1(\lambda) \cdot \lambda d\lambda / \int f'_2(\lambda) \cdot \lambda d\lambda \quad (3.20)$$

Tot nu toe hebben we verondersteld dat PLQY gelijk is aan 100%. Dit zal echter meestal niet het geval zijn. Om hier rekening mee te houden vermenigvuldigen we op het einde nog met PLQY en is dus

$$f_2(\lambda) = a \cdot PLQY \cdot f'_2(\lambda) \quad (3.21)$$



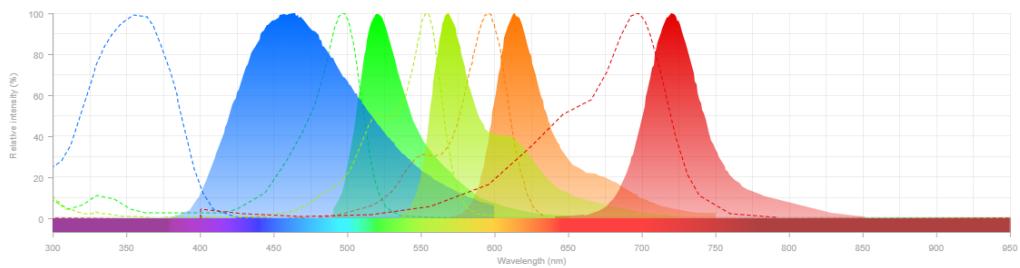
**Figuur 3.11:** Voorbeeld resultaten met geavanceerd model (links = fosfor, rechts = quantum dot)

Zoals te zien is in 3.10 en 3.11 zijn de resultaten bij het gebruik van fosforen inderdaad zeer vergelijkbaar. Bij quantum dots is er echter een groot verschil. Dit omdat nu correct rekening gehouden wordt met de niet normale verdeling van het excitatiespectrum.

### 3.8.3 Spectra

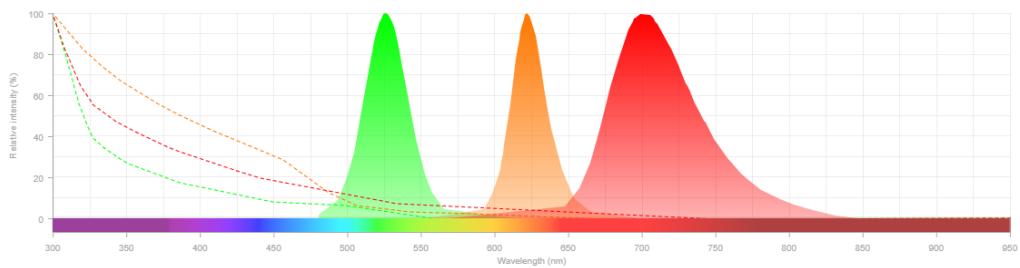
We maken gebruik van een algemeen model van de spectra zodat we de gebruiker kunnen laten zien wat de effecten zijn van de verschillende parameters van luminescente materialen. Hiervoor moeten we dus per standaard materiaal een model gemaakt worden.

Voor fosforen maakten we een model dat een normale verdeling volgt. Dit omdat de meeste fosforen een excitatie- en emissiespectrum hebben die ongeveer een normale verdeling volgen. Deze zijn meestal beiden gecentreerd rond de excitatie- en emissiegolfleugte met een bepaalde "skew". Het is dan ook voor deze reden dat we de mogelijkheid voorzien om het model aan te passen naar bijvoorbeeld een normaalverdeling met "skew" (Paragraaf 3.11).



**Figuur 3.12:** Voorbeeld fosfor spectra (stippellijn = excitatie, vol = emissie) [21]

Het modelleren van quantum dots is echter moeilijker. Ze hebben net zoals fosforen een normaal verdeeld emissiespectrum. Het excitationspectrum is echter niet normaal verdeeld. Ze volgen eerder een dalende lijn.



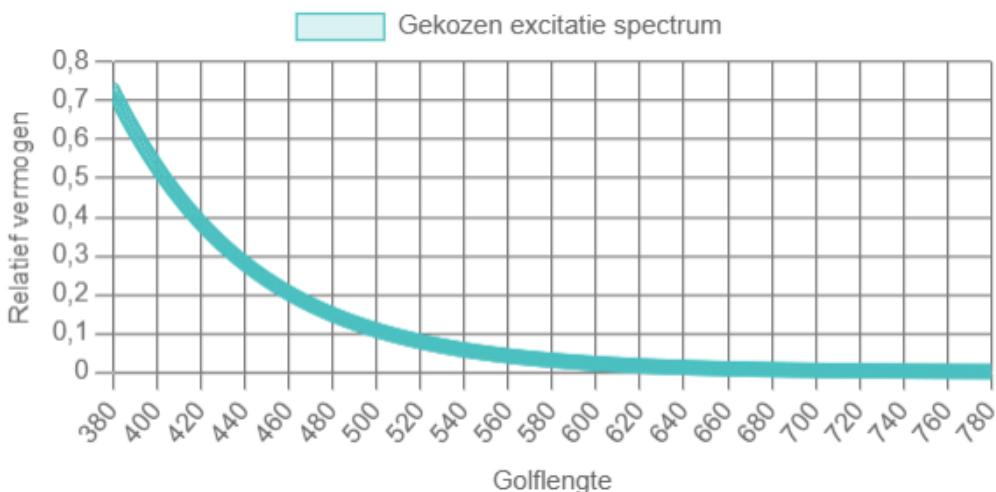
**Figuur 3.13:** Voorbeeld fosfor QD (stippellijn = excitatie, vol = emissie) [21]

Daarnaast zijn er ook in de lijn soms pieken te zien. Ten laatste varieert de lijn ook sterk afhankelijk van de configuratie van de quantum dot. Dit maakt het modelleren van quantum dots moeilijker dan fosforen. Zo bestaan er wel modellen voor quantum dots maar zijn deze vaak zeer specifiek afhankelijk van de configuratie van de quantum dot. Dit zijn dan bijvoorbeeld modellen die een empirische fit zijn van meetresultaten.

Omdat we een zo algemeen mogelijk model willen maken voor het excitationspectrum is dit type model dus geen goede optie. Ook zou het aanpassen van dit type model weinig bruikbaar zijn omdat het meestal gebruik maakt van moeilijkere parameters. Die ook weinig relevant zijn voor de gebruiker van onze app.

Uit deze vaststellingen besloten we om een simpel model te gebruiken voor het excitatiespectrum van de quantum dots. Dit model volgt een dalende exponentiële lijn. Dit model is echter niet perfect en dus voorzien we de optie om de gebruikte formule aan te passen. Daarnaast voorzien we hiervoor ook de mogelijkheid om eigen luminescente materialen toe te voegen. Dit bijvoorbeeld met gemeten data van een specifieke quantum dot voor volledige accuraatheid.

$$0.03 \cdot \exp\left(\frac{\log(0.1) \cdot (\lambda - 360)}{\lambda_{em} - 360}\right) \quad (3.22)$$



**Figuur 3.14:** Standaard excitatiespectrum qunatum dot

### 3.9 Implementatie LED visualisatie

De led visualisatie wordt geïmplementeerd aan de hand van Konva. Dit is een library die het eenvoudig maakt om op een canvas te tekenen. Ze zorgt ervoor dat we makkelijk mooiere figuren kunnen maken worden dan met de standaard canvas van HTML. Daarnaast is het ook eenvoudig om interactie aan de figuur toe te voegen. Dit omdat we kunnen gebruik maken van Konva voor Vue. Dit is een library die het eenvoudig maakt om Konva te gebruiken in combinatie met Vue en dus reactiviteit aan Konva toevoegt.

De visualisatie is vrij simpel (Figuur 3.15), ze bestaat uit een trapezium met hierin rechthoeken, punten en kristallen. De trapezium stelt de chip voor, de rechthoeken de LEDs, de kristallen de fosforen en de punten de quantum dots.

De ingestelde waarden van de gebruiker hebben invloed op de gemaakte figuur. Zo bepaald de golflengte/emissiegolflengte de kleur van de objecten in de chip. De concentratie van de luminescente materialen bepaalt dan weer de hoeveelheid van deze materialen in de chip.

Om de kleur van de objecten te bepalen maken we gebruik van een eenvoudige berekening om

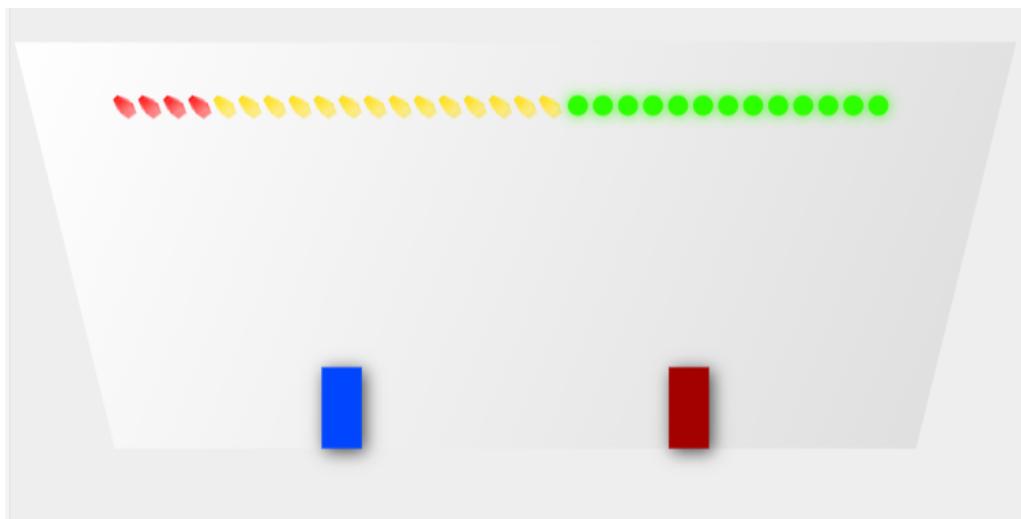
de golflengte om te zetten naar een hexadecimale kleurwaarde. De berekening werkt binnen bepaalde bereiken van golflengtes en zet hierin telkens de rode, groene en blauwe waarde. In het bereik tussen 380 en 440nm wordt bijvoorbeeld blauw op één gezet, groen op 0 en rood krijgt een waarde afhankelijk van de volgende formule:

$$\text{rood} = -(\lambda - 440)/(440 - 380) \quad (3.23)$$

Ten slotte wordt nog een factor en een gamma toegevoegd. De factor zorgt ervoor dat de intensiteit minder sterk wordt wanneer de grenzen van het zichtbare licht bereikt worden. Het gamma zorgt ervoor dat de kleuren er natuurlijker uitzien. Dit gamma wordt op 0.8 geplaatst. Door voor rood, groen en blauw de volgende berekening uit te voeren bekomen we de uiteindelijke kleurwaarde:

$$\text{kleur} = \begin{cases} 255 \cdot (\text{waarde} \cdot \text{factor})^{\text{gamma}} & \text{als waarde} > 0 \\ 0 & \text{anders} \end{cases} \quad (3.24)$$

We implementeren de visualisatie op deze manier om de gebruiker een eenvoudig beeld te geven over hoe een door hem gemaakte LED eruit zou zien. Dit kan de gebruiker helpen begrijpen hoe een chip ongeveer is opgebouwd en hoe de verschillende materialen in de chip zouden kunnen geplaatst zijn op een eenvoudige wijze.



**Figuur 3.15:** Voorbeeld LED visualisatie (2 fosforen, 1 QD en 2 LEDs)

### 3.10 Implementatie hyperspectral image rendering

Voor hyperspectral image rendering gebruiken we net zoals voor color rendering de luxpy bibliotheek [23]. Als we echter gewoon de standaard implementatie gebruiken zorgt het in dit geval voor problemen. Dit omdat het uitvoeren van de standaardimplementatie op de server

maar liefst 15 minuten duurt. Daarnaast gebruikt deze methode ook een grote hoeveelheid geheugen. Als dus meerdere mensen tegelijk deze standaardimplementatie zouden gebruiken zou de server al rap niet genoeg geheugen hebben en crashen. Daarnaast vinden we 15 minuten zeker een stuk te lang voor een gebruiker om te wachten op het resultaat van de rendering.

Om dit probleem op te lossen onderzochten we twee mogelijke oplossingen:

- Aanpassen van de gebruikte foto.
- Berekeningen die niet veranderen opslaan.

### 3.10.1 Foto

De standaard foto 3.16 die luxpy [23] gebruikt is een vrij complexe foto met een hoge resolutie. Deze combinatie van de hoge resolutie en de vele kleuren in de foto zorgt ervoor dat de berekeningen langer duren dan gewenst. Om dit op te lossen kijken we naar twee oplossingen:

- Resolutie verlagen
- Complexiteit foto verlagen



**Figuur 3.16:** Standaard foto luxpy [23]

Ten eerste onderzochten we het gebruik van een simpelere foto. We gebruikten hiervoor eerst een heel simpele foto van een blauwe bal 3.17 op een witte achtergrond. Dit verbeterde de

sneldheid van de berekening sterk, maar bracht een belangrijk probleem met zich mee: de foto geeft weinig informatie over de belichting.

Het grootste probleem is namelijk dat een gebruiker niet kan weten hoe de bal er normaal uitziet. Er bestaan namelijk zeer veel blauwe ballen in zeer veel verschillende tinten blauw. Het is dus hierdoor ook moeilijk af te leiden wat het effect van de gebruikte belichting is. De gebruiker kan dus niet leren over de effecten van de belichting op de kleur van het object.

REALISTIC VECTOR



**Figuur 3.17:** Foto blauwe bal

Om dit op te lossen keken we naar een eenvoudigere foto, die echter wel voorwerpen bevat waarvan de gebruiker de kleur zeker kent. Zo kwamen we op het idee om een verzameling fruit op een witte achtergrond te gebruiken, zoals te zien is in Figuur 3.18.

Hierdoor blijft de hoeveelheid kleuren beperkt en hebben waardoor de versnelling optreedt. Daarnaast hebben de meeste voorwerpen een vaste kleur wat ervoor zorgt dat de gebruiker de normale kleur van de voorwerpen kent. Deze verbetering zorgde echter nog niet voor genoeg winst in snelheid, dus onderzochten we ook de resolutie.



**Figuur 3.18:** Foto fruit 6,99MB

Ten tweede keken we naar de resolutie van de foto. Dit betekende simpelweg dat we de foto verkleinden. We deden dit systematisch totdat we een versie kregen die zowel een voldoende

snelle berekening als een scherp genoeg beeld opleverde. Uiteindelijk verkleinden we de foto van 6,99 MB naar 9,09 kB, zoals te zien is in Figuur 3.19. Het is mogelijk de foto nog verder te verkleinen, maar dit zou ten koste gaan van de kwaliteit van de foto. We vonden dat de kwaliteit van de foto in Figuur 3.18 nog voldoende was om de gebruiker een goed beeld te geven van de belichting van de foto. Bij kleinere foto's beginnen bepaalde details verloren te gaan, wat het moeilijker maakt voor de gebruiker om de belichting correct in te schatten.



**Figuur 3.19:** Foto fruit 9,09kB

Met een combinatie van deze twee aanpassingen op de foto bekomen we een foto die snel genoeg is ( $\pm$  5 seconden) om te renderen en toch voldoende informatie bevat voor de gebruiker.

### 3.10.2 Berekeningen

Om de berekening nog sneller te maken kijken we ook naar het opslaan van niet veranderende berekeningen. Zo zal de rendering van de verschillende beschikbare referentiespectra bijvoorbeeld nooit veranderen. We kunnen de rendering van de referentiespectra dus opslaan en deze opnieuw gebruiken bij elke berekening van de rendering.

Deze aanpassing zorgt ervoor dat de rendering zelf een stuk sneller kan verlopen doordat, maar half zo veel werk moet gebeuren. Doordat we echter reeds de tijd van de berekening sterk verlaagt hebben door de aanpassingen aan de foto, is het effect van deze aanpassing minder groot. Doordat de resolutie voor een grote verandering in tijd zorgt, treed er een probleem op bij het voordien berekenen van de foto. Door de lage resolutie is namelijk de tijd die nodig is voor het berekenen van het referentiespectra gedeeltelijk ongeveer gelijk aan de tijd die nodig is voor het inladen van de foto. Hierdoor is het effect van het voordien berekenen van het referentiespectra verwaarloosbaar. Moesten we echter in de toekomst een foto met een grotere resolutie willen gebruiken kan deze aanpassing mogelijk wel een voldoende groot effect hebben.

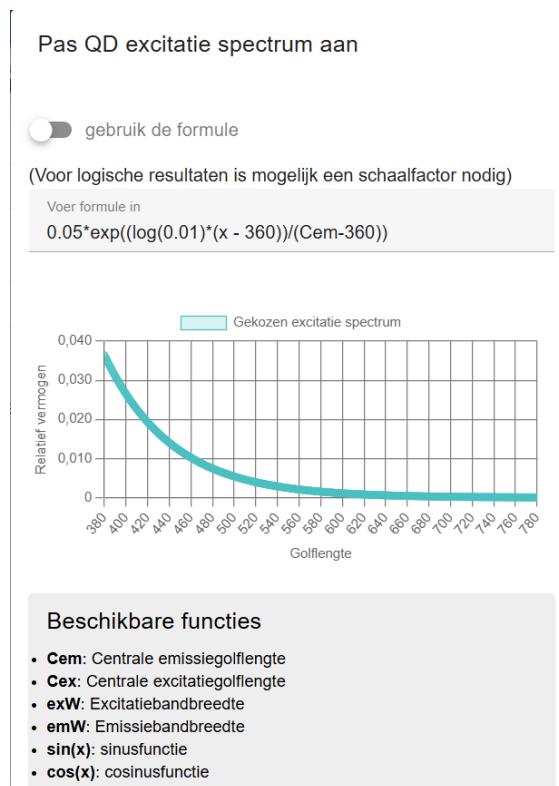
### 3.11 Implementatie data manager

We maken de app bruikbaarder voor geavanceerde gebruikers door een data manager toe te voegen. Hierin krijgt de gebruiker de mogelijkheid om gebruikte defaults aan te passen en te bekijken. Daarnaast kan de gebruiker eigen luminescente materialen toevoegen.

De eerste functie in de data manager is het vergrendelen van bepaalde variabelen. Met deze functionaliteit kan de gebruiker verschillende parameters van de inputs vergrendelen of ontgrendelen. Dit is vooral bedoeld voor educatieve doeleinden, bijvoorbeeld wanneer een docent leerlingen vraagt om een specifieke waarde, zoals de excitatiebreedte, vast in te stellen.

Daarnaast kan de gebruiker het referentiespectrum voor de realistic rendering aanpassen. Hierbij kan hij kiezen uit standaard spectra zoals D65, Tungsten, Clear Sky en Daylight. Voor color rendering kan hij bovendien het CRI-type instellen (CIE Ra-15, IES TM-30).

Ten derde kan de gebruiker de formules voor de standaard spectra van luminescente materialen en het LED-emissiemodel aanpassen. Hij kan gebruikmaken van standaard formules en parameters om een wiskundige formule te definiëren voor de spectra van de luminescente materialen. Deze formule kan vervolgens worden in- of uitgeschakeld. Tijdens het maken van de formule tonen we een grafiek van het gegenereerde spectrum om het beslissingsproces te vergemakkelijken.



**Figuur 3.20:** Voorbeeld input formule

Ten vierde krijgt de gebruiker in de data manager inzicht in de verschillende gebruikte standaardparameters. Zo kan hij de standaard EQE-waarden, het standaard LED-emissiemodel en de eigenschappen van luminescente materialen bekijken. Daarnaast tonen we bij de EQE- en LED-emissiemodeldata ook de bron van deze gegevens.

Zoals eerder vermeld, kan de gebruiker ook eigen luminescente materialen toevoegen 3.4.4. Tot slot kan hij de gebruikte EQE-waarden aanpassen. Dit gebeurt via een CSV-bestand, zoals beschreven in 3.5.3.

## 3.12 Hosting

De webapp wordt gehost op een combinatie van nginx (frontend) en systemctl (Flask-backend). Nginx fungeert niet alleen als host voor de frontend, maar ook als reverse proxy. Hierdoor kan het de requests van de frontend naar de backend sturen en tegelijkertijd rate limiting toepassen. Dit voorkomt overbelasting van de server, vooral bij rekenintensieve processen zoals hyperspectral image rendering, die veel tijd in beslag nemen.

# **Hoofdstuk 4**

## **Uitbreidingen**

De webapp bevat alle belangrijkste functionaliteiten. We zouden nog enkele concepten, zoals de "melanopic eye sensitivity", kunnen toevoegen. Dit concept meet hoe gevoelig het menselijk oog is voor licht. Toch hebben we besloten om dit en andere complexe concepten niet op te nemen, omdat ze de webapp onnodig ingewikkeld zouden maken. Ons doel is om de webapp zo eenvoudig en toegankelijk mogelijk te houden.

Naast het toevoegen van concepten kunnen we de webapp ook uitbreiden met extra functionaliteiten. Zo kunnen we bijvoorbeeld gebruikers de mogelijkheid geven om een account aan te maken. Met dit account kunnen zij instellingen, LED-configuraties en andere persoonlijke voorkeuren permanent opslaan.

Daarnaast kunnen we het educatieve aspect van de webapp verder uitbreiden. Momenteel ligt de focus vooral op interactiviteit en experimentatie. We kunnen echter extra educatieve elementen toevoegen, zoals uitleg over de gebruikte concepten en oefeningen om de leerervaring te verdiepen.

### **4.1 Gamification**

Zoals eerder vermeld, kan de webapp uitgebreid worden met meer educatieve aspecten. Gamification is een populaire methode om dit te realiseren [25]. Dit houdt in dat we game-elementen toevoegen aan een educatieve app, zonder dat het een volledige game wordt. In tegenstelling tot "Game-based learning", waarbij leren plaatsvindt door een spel te spelen, verhoogt gamification de motivatie en aandacht van de gebruiker door spelelementen te integreren. Hierdoor leert de gebruiker meer en beter.

Veelgebruikte game-elementen zijn:

- punten
- uitdagingen
- levels

- klassementen
- badges

Deze elementen houden niet alleen de gebruiker gemotiveerd, maar stimuleren ook interactie en discussie tussen gebruikers of studenten. Dit bevordert wederzijds leren en kan leiden tot een gezonde competitie binnen de gebruikersgroep.

## 4.2 Analyse model luminescente materialen

Het geavanceerde model (Paragraaf 3.8.2) voor de luminescente materialen is momenteel geïmplementeerd in de app, maar nog niet vergeleken met experimentele data. De app gebruikt dit model uitsluitend op basis van het beschikbare bewijs. Daarom is het belangrijk om het model te toetsen aan experimentele data. Dit bevestigt de correctheid en biedt de mogelijkheid om het model verder te optimaliseren.

Voor de excitatie- en emissiespectra gebruiken we eenvoudige formules, maar deze kunnen we uitbreiden met geavanceerdere formules die beter aansluiten op de realiteit. Daarnaast kunnen we meerdere standaardformules aanbieden, zodat zowel algemene als specifieke formules beschikbaar zijn. De gebruiker kiest vervolgens zelf welke formule hij of zij wil gebruiken.

## **Hoofdstuk 5**

### **Conclusie**

# Bibliografie

- [1] "LED Spectrum Simulator — Waveform Lighting", Waveform Lighting. adres: <https://www.waveformlighting.com/led-spectrum-simulator>. Geopend: 29.03.2025.
- [2] J. Selverian, *ColorCalculator Software*, JAHM Software Inc., 8 nov 2017. adres: [https://www.researchgate.net/publication/320930275\\_ColorCalculator\\_software](https://www.researchgate.net/publication/320930275_ColorCalculator_software), Geopend: 29.03.2025.
- [3] D. L. DiLaura, K. W. Houser, R. G. Mistrick en G. R. Steffy, *The lighting handbook*, 10th. IES Illuminating Engineering Society, 2011, isbn: 978-087995-241-9.
- [4] W. van Bommel, *Interior lighting, Fundamentals, Technology and Application*. Springer, 2019, isbn: 978-3-030-17194-0.
- [5] E. F. Schubert, *Light-Emitting Diodes*, 2nd. Cambridge University Press, 2006, isbn: 978-0-521-86538-8.
- [6] V. K. Khanna, *Fundamentals of Solid-State Lighting, LEDs, OLEDs, and Their Applications in Illumination and Displays*. CRC Press, 2014, isbn: 978-1-4665-6109-0.
- [7] "External Quantum Efficiency". adres: <https://www.sciencedirect.com/topics/engineering/external-quantum-efficiency>. (Geopend: 29.03.2025).
- [8] B. Hahn, A. Bauer en M. Binder, "Closing the Green Gap", *Osram Opto Semiconductors*, 29 jan 2019. adres: [https://www.energy.gov/sites/prod/files/2019/02/f59/hahn-gap\\_ssl-rd2019.pdf](https://www.energy.gov/sites/prod/files/2019/02/f59/hahn-gap_ssl-rd2019.pdf).
- [9] M. Krames, *Light-Emitting Diode Technology for Solid-State Lighting*. Soraa Inc., 2010. adres: <https://nap.nationalacademies.org/read/12821/chapter/12>, Geopend: 29.03.2025.
- [10] D. Zhu en C. J. Humphreys, "Solid-State Lighting Based on Light Emitting Diode Technology", in *Optics in Our Time*, M. D. Al-Amri, M. El-Gomati en M. S. Zubairy, red. Springer, 14 dec 2016, p. 87–118, isbn: 978-3-319-31903-2. doi: 10.1007/978-3-319-31903-2\_5. adres: [https://doi.org/10.1007/978-3-319-31903-2\\_5](https://doi.org/10.1007/978-3-319-31903-2_5), "Geopend: 29.03.2025".
- [11] Wikipedia, red. "CIE 1931 Color Space". adres: [https://en.wikipedia.org/wiki/CIE\\_1931\\_color\\_space](https://en.wikipedia.org/wiki/CIE_1931_color_space). Geopend: 29.03.2025.
- [12] Wikipedia, red. "Chromaticiteit". adres: <https://nl.wikipedia.org/w/index.php?title=Chromaticiteit&oldid=49823391>. Geopend: 29.03.2025.
- [13] "Color Rendering of Light Sources", DigiKey. adres: <https://www.digikey.pl/pl/articles/color-rendering-of-light-sources>. Geopend: 29.03.2025.
- [14] "What Is CRI R9 and Why Is It Important? — Waveform Lighting", Waveform Lighting. adres: <https://www.waveformlighting.com/tech/what-is-cri-r9-and-why-is-it-important>. Geopend: 29.03.2025.
- [15] C. Shen. "The Principle and Basic Calculation of CRI", LISUN GROUP. adres: <https://www.lisungroup.com/news/technology-news/the-principle-and-basic-calculation-of-cri.html>. Geopend: 29.03.2025.

- [16] Wikipedia, red. "Color rendering index". adres: [https://en.wikipedia.org/wiki/Color\\_rendering\\_index](https://en.wikipedia.org/wiki/Color_rendering_index). Geopend: 29.03.2025.
- [17] Wikipedia, red. "Chromatic Adaptation". adres: [https://en.wikipedia.org/w/index.php?title=Chromatic\\_adaptation&oldid=1188854965](https://en.wikipedia.org/w/index.php?title=Chromatic_adaptation&oldid=1188854965). Geopend: 29.03.2025.
- [18] A. D. et al., *IES Method for Evaluating Light Source Color Rendition*. IES Illuminating engineering society, 2018, isbn: 978-0-87995-379-9.
- [19] P. Sims. "What Is the Gamut Area Index (GAI) for Color Rendition?", Luminus Devices. adres: <https://luminusdevices.zendesk.com/hc/en-us/articles/4419618181389-What-is-the-Gamut-Area-Index-GAI-for-Color-Rendition>. Geopend: 29.03.2025.
- [20] M. Zollers. "Phosphor Modeling in LightTools". adres: <https://www.synopsys.com/content/dam/synopsys/optical-solutions/documents/datasheets/modeling-phosphors-in-lighttools.pdf>. (Geopend: 29.03.2025).
- [21] "Fluorescence SpectraViewer". adres: <https://www.thermofisher.com/order/fluorescence-spectraviewer/#!/>. Geopend: 29.03.2025.
- [22] I. Krukowski. "Vue 3 I18n: Building a Multi-Language App with Locale Switcher", Lokalise Blog. adres: <https://lokalise.com/blog/vue-i18n/>. Geopend: 29.03.2025.
- [23] K. A. G. Smet, "Tutorial: The LuxPy Python Toolbox for Lighting and Color Science", *LEUKOS*, jrg. 16, nr. 3, p. 179–201, 2 mei 2019. doi: 10.1080/15502724.2018.1518717. adres: <https://doi.org/10.1080/15502724.2018.1518717>, Geopend: 29.03.2025.
- [24] Y. Ohno, "Spectral Design Considerations for White LED Color Rendering", *Optical Engineering*, jrg. 44, nr. 11, 1 nov 2005, issn: 0091-3286. adres: <http://opticalengineering.spiedigitallibrary.org/article.aspx?doi=10.1117/1.2130694>, Geopend: 29.03.2025.
- [25] A. N. Saleem, N. M. Noori en F. Ozdamli, "Gamification Applications in E-learning: A Literature Review", *Technology, Knowledge and Learning*, jrg. 27, nr. 1, p. 139–159, issn: 2211-1670. doi: 10.1007/s10758-020-09487-x. adres: <https://doi.org/10.1007/s10758-020-09487-x>, Geopend: 29.03.2025.

## **Bijlage A**

### **Een aanhangsel**

sdfsffqsf

## **Bijlage B**

# **Beschrijving van deze masterproef in de vorm van een wetenschappelijk artikel**

The thesis should also contain a short scientific article. If you write your thesis in Dutch, you must write the article in English, and vice versa. We advise you to employ the IEEE Manuscript Templates for Conference Proceedings (<https://www.overleaf.com/latex/templates/ieee-conference-template/grfzhnncsfqn>). Compile the article in another project and include the generated pdf file as shown below:

# Conference Paper Title\*

\*Note: Sub-titles are not captured in Xplore and should not be used

1<sup>st</sup> Given Name Surname  
dept. name of organization (of Aff.)  
name of organization (of Aff.)  
City, Country  
email address or ORCID

2<sup>nd</sup> Given Name Surname  
dept. name of organization (of Aff.)  
name of organization (of Aff.)  
City, Country  
email address or ORCID

3<sup>rd</sup> Given Name Surname  
dept. name of organization (of Aff.)  
name of organization (of Aff.)  
City, Country  
email address or ORCID

4<sup>th</sup> Given Name Surname  
dept. name of organization (of Aff.)  
name of organization (of Aff.)  
City, Country  
email address or ORCID

5<sup>th</sup> Given Name Surname  
dept. name of organization (of Aff.)  
name of organization (of Aff.)  
City, Country  
email address or ORCID

6<sup>th</sup> Given Name Surname  
dept. name of organization (of Aff.)  
name of organization (of Aff.)  
City, Country  
email address or ORCID

**Abstract**—This document is a model and instructions for L<sup>A</sup>T<sub>E</sub>X. This and the IEEEtran.cls file define the components of your paper [title, text, heads, etc.]. \*CRITICAL: Do Not Use Symbols, Special Characters, Footnotes, or Math in Paper Title or Abstract.

**Index Terms**—component, formatting, style, styling, insert

## I. INTRODUCTION

This document is a model and instructions for L<sup>A</sup>T<sub>E</sub>X. Please observe the conference page limits.

## II. EASE OF USE

### A. Maintaining the Integrity of the Specifications

The IEEEtran class file is used to format your paper and style the text. All margins, column widths, line spaces, and text fonts are prescribed; please do not alter them. You may note peculiarities. For example, the head margin measures proportionately more than is customary. This measurement and others are deliberate, using specifications that anticipate your paper as one part of the entire proceedings, and not as an independent document. Please do not revise any of the current designations.

## III. PREPARE YOUR PAPER BEFORE STYLING

Before you begin to format your paper, first write and save the content as a separate text file. Complete all content and organizational editing before formatting. Please note sections III-A–III-E below for more information on proofreading, spelling and grammar.

Keep your text and graphic files separate until after the text has been formatted and styled. Do not number text heads—L<sup>A</sup>T<sub>E</sub>X will do that for you.

Identify applicable funding agency here. If none, delete this.

### A. Abbreviations and Acronyms

Define abbreviations and acronyms the first time they are used in the text, even after they have been defined in the abstract. Abbreviations such as IEEE, SI, MKS, CGS, ac, dc, and rms do not have to be defined. Do not use abbreviations in the title or heads unless they are unavoidable.

### B. Units

- Use either SI (MKS) or CGS as primary units. (SI units are encouraged.) English units may be used as secondary units (in parentheses). An exception would be the use of English units as identifiers in trade, such as “3.5-inch disk drive”.
- Avoid combining SI and CGS units, such as current in amperes and magnetic field in oersteds. This often leads to confusion because equations do not balance dimensionally. If you must use mixed units, clearly state the units for each quantity that you use in an equation.
- Do not mix complete spellings and abbreviations of units: “Wb/m<sup>2</sup>” or “webers per square meter”, not “webers/m<sup>2</sup>”. Spell out units when they appear in text: “. . . a few henries”, not “. . . a few H”.
- Use a zero before decimal points: “0.25”, not “.25”. Use “cm<sup>3</sup>”, not “cc”).

### C. Equations

Number equations consecutively. To make your equations more compact, you may use the solidus (/), the exp function, or appropriate exponents. Italicize Roman symbols for quantities and variables, but not Greek symbols. Use a long dash rather than a hyphen for a minus sign. Punctuate equations with commas or periods when they are part of a sentence, as in:

$$a + b = \gamma \quad (1)$$

Be sure that the symbols in your equation have been defined before or immediately following the equation. Use “(1)”, not

## **Bijlage C**

### **Poster**

## **Bijlage D**

# **GenAI code of conduct for students**

Generative AI (GenAI) assistance tools can be used to generate text, image, code, video, music, or combinations of these. It includes typical tools like (but this list is not limited to): ChatGPT, Google Gemini, MS Copilot, Midjourney, Claude.ai, Perplexity.ai, Dall-E, etc.

### **Student Information**

Student name: first name last name

This form is related to my Master thesis.

Title Master thesis: Title Masterproof Promoter: \_\_\_\_\_

Daily supervisor: \_\_\_\_\_

### **Use of GenAI Assistance**

Please indicate with "X" (possibly multiple times) in which way you were using GenAI:

- I did not use any GenAI assistance tool.
- I did use GenAI assistance. In this case, specify which ones (e.g., ChatGPT, . . .):  
\_\_\_\_\_

### **GenAI Assistance Used As/For**

- MS Copilot
- Language assistance

- Search engine
- Literature search
- Short-form input assistance
- Generating programming code
- Generating new research ideas
- Generating blocks of text
- Other (specify): \_\_\_\_\_

## **Code of Conduct for Different Uses**

### **Language Assistant for Reviewing or Improving Texts**

This use is similar to using spelling and grammar check tools, and you do not have to refer to the use of GenAI in the text. Be careful:

- Using GenAI tools on texts you did not write yourself to cover up plagiarism (by paraphrasing original texts) is not allowed.

### **Search Engine for Information or Existing Research**

This use is similar to a Google search or checking Wikipedia. If you then write your own text based on this information, you do not have to refer to the use of GenAI in the text. Be careful:

- Be aware that the output of the GenAI tool cannot be guaranteed as a 100% reliable source of information.
- The output may not be entirely correct and is limited due to the databases it uses. Knowledge evolves and may change over time, and it may be that the database of the GenAI tool is not up to date.

## Literature Search

This use is comparable to a Google Scholar search. Be careful:

- Be aware that the output is restricted to the database it is built on. After this initial search, look for scientific sources and conduct your own analysis.
- GenAI tools (like ChatGPT) may output no or wrong references. As a student, you are responsible for further checking and verifying the accuracy of references.

## Short-form Input Assistance

This use is similar to Google Docs powered by generative language models.

## Generating Programming Code

The use of GenAI for coding should be explicitly allowed by the teacher. If used for coding, correctly mention the use of GenAI assistance and cite it.

## Generating New Research Ideas

Further verify whether the idea is novel or not. It is likely that it is related to existing work, which should be referenced.

## Generating Blocks of Text

Inserting blocks of text without quotes and a reference to GenAI assistance in your report or thesis is not allowed. Be careful:

- When you literally copy elements from a conversation with a GenAI tool, quote them between quotation marks and refer to them according to the specified reference style.
- Describe the use of the GenAI tool (tool name, version, date, etc.) in the method section and optionally add the full conversation as an attachment.

## Other

Contact the professor of the course or the promoter of the thesis. Inform also the program director. Motivate how you comply with Article 84 of the exam regulations. Explain the use and added value of ChatGPT or another AI tool.

## **Further Important Guidelines and Remarks**

- GenAI assistance cannot be used related to data or subjects under Non-Disclosure Agreement.
- GenAI assistance cannot be used related to sensitive or personal data due to privacy issues.
- Take a scientific and critical attitude when interacting with GenAI assistance and interpreting its output.
- As a student, you are responsible for complying with Article 84 of the exam regulations: your report or thesis should reflect your own knowledge, understanding, and skills.

### **Exam Regulations Article 84**

"Every conduct individual students display with which they (partially) inhibit or attempt to inhibit a correct judgement of their own knowledge, understanding and/or skills or those of other students, is considered an irregularity which may result in a suitable penalty. A special type of irregularity is plagiarism, i.e., copying the work (ideas, texts, structures, designs, images, plans, codes, ...) of others or prior personal work in an exact or slightly modified way without adequately acknowledging the sources."

### **Additional Reading**

More information about being transparent on the use of GenAI assistance and about citing and referencing GenAI can be found on the student website.

### **A Few Final Words**

If you are uncertain whether or not you should declare your use of AI tools, discuss the matter with your teacher or promoter. It is safer to declare AI use when it is not needed than to withhold that declaration when it is required.

Finally, remember that advanced AI tools are new and that they can do things they could not do until recently. It is important to follow up on the most recent developments in AI technologies and communicate openly with your teachers, assistants, supervisors, and peers.

**KU LEUVEN - GENT**  
Faculteit Industriële Ingenieurswetenschappen  
Campus Rabot, Gebroeders De Smetstraat 1  
9000 Gent, België  
tel: + 32 9 265 86 10  
[iiw.gent@kuleuven.be](mailto:iiw.gent@kuleuven.be)  
[www.iiw.kuleuven.be](http://www.iiw.kuleuven.be)

