

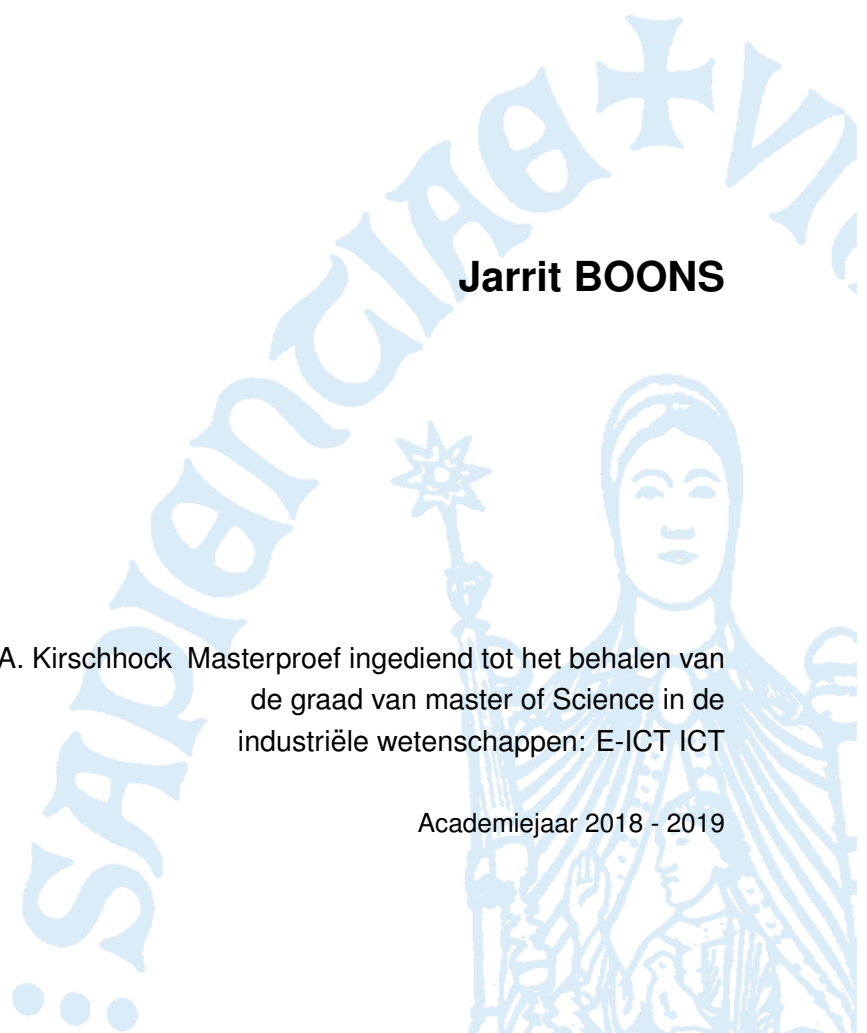
(Tussentijds Verslag) Wandelen door kristallen

Blender als kristallografische visualisatietool

Jarrit BOONS

Promotor: prof. Christine E. A. Kirschhock Masterproef ingediend tot het behalen van
de graad van master of Science in de
Promotor: lic. Ann Phillips industriële wetenschappen: E-ICT ICT

Academiejaar 2018 - 2019



©Copyright KU Leuven

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 Technologicampus De Nayer, Jan De Nayerlaan 5, B-2860 Sint-Katelijne-Waver, +32 15 31 69 44 of via e-mail iiw.denayer@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.

Inhoudsopgave

Inhoudsopgave	iii
Inhoud	iii
Lijst van figuren	iv
Lijst van tabellen	v
Lijst met afkortingen	vi
1 Inleiding	1
1.1 Situering van het onderzoek	1
1.2 Probleemstelling	2
1.3 Doelstelling	2
1.4 Onderzoeksvragen	3
1.5 Structuur van de tekst	3
1.6 Conclusie	4
2 Literatuurstudie	5
2.1 Kristallografie	5
2.2 Het Crystallographic Information File (CIF) formaat	9
2.3 Kristalvisualisatiesoftware	12
2.4 CIF-Parsers	14
2.5 Blender en de Blender API	16
2.6 Conclusie	19
3 Ontwerpproces	21
3.1 Van cif naar py	21
Bibliografie	23

Lijst van figuren

2.1	Voorstelling van een eenheidscel in de roosterruimte (Borchardt-Ott, 2011)	7
2.2	De 6 kristalfamilies	7
2.3	De 14 bravaisroosters en hun kristalsystemen	8
2.4	De user interface van VESTA3	13
2.5	De user interface van Crystalviewer9	13
2.6	De user interface van Olex ²	14
2.7	Het Blender startscherf	16
2.8	Het Blender project van de winnaar van een fotorealiteit wedstrijd, na rendering	18
3.1	Klassendiagramma van het ontwerp van deze thesis	22

Lijst van tabellen

2.1	Kristallen in wetenschappelijke takken	6
2.2	De submodules van de Blender bpy module (Conlan, 2017)	19

Lijst van afkortingen en begrippen

ASCII

American Standard Code for Information Interchange, standaard voor de codering van karakters

API

Application Programming Interface, een verzameling van definities en methodes

Open source

Een product dat vrij mag gebruikt worden

GUI

Graphic User Interface, een visuele interface tussen een programma en de gebruiker

Interface

Manier van interactie tussen twee zaken

Syntax

Vorm en structuur van een tekst

XML

Extensible Markup Language, standaardformaat met syntaxregels voor het opslaan van gegevens

Hoofdstuk 1

Inleiding

Dit hoofdstuk omschrijft de context waarin deze thesis zich zal afspelen en vormt het fundament van deze scriptie. Om de rest van deze thesis te begrijpen is het van belang dit hoofdstuk grondig te lezen.

De eerste sectie van dit hoofdstuk schetst onder welke onderzoeksdomeinen deze thesis kan worden ondergebracht. De tweede sectie beschrijft het probleem waarop deze thesis een oplossing zal trachten te bieden. Het algemene doel van deze thesis is het vinden en uitwerken van deze oplossing, dit doel zal verder worden opgedeeld in verschillende, kleinere, doelstellingen. In sectie vier worden deze doelstellingen geformuleerd als onderzoeksvragen bestaande uit een hoofdvraag en enkele deelvragen. Ten slotte is er nog een vijfde sectie waarin de opbouw van deze scriptie wordt beschreven. In een laatste sectie wordt dit hoofdstuk samengevat in een beknopte conclusie.

1.1 Situering van het onderzoek

Kristallografie is een tak van de wetenschap met als hoofdrol een eerder klein object, een kristal. Kristallen kunnen gezien worden als een puzzel van atomen. De stukjes van zo een puzzel kunnen verschillende chemische elementen zijn, maar evengoed allemaal dezelfde. Wat de kristallen zo uniek maakt is hoe deze zijn opgebouwd. De opbouw bepaalt allerhande eigenschappen van dat bepaalde kristal. De complexiteit van kristallen kan oplopen tot op het punt waarop het bijna onmogelijk wordt deze in woorden te beschrijven en zelfs nog moeilijker deze te interpreteren. Dit is waar computertechnologie aan te pas komt.

Sinds enkele decennia zijn wetenschappers in staat kristalstructuren om te zetten in een digitaal formaat dat leesbaar is door zowel mensen als computers, het CIF-formaat (zie later). Dankzij dit standaardformaat is het mogelijk geworden de structuur van kristallen te lezen en te delen zonder kans op misinterpretatie, en biedt het computers de mogelijkheid zelfs de meest complexe kristalstructuren naar een driedimensionaal beeld om te zetten. Het 3D visualiseren van kristallen geeft wetenschappers meer inzicht en geeft meer mogelijkheden om hun kennis over te brengen. Dit alles heeft geleid tot een nauwe samenwerking tussen kristallografie en computerwetenschappen om de wereld van de kristallografie tot leven te brengen.

1.2 Probleemstelling

De programma's die op dit moment door wetenschappers worden gebruikt bij het 3D visualiseren van kristallen bieden nog lang niet de vrijheid en aanbod aan features welke sommige hedendaagse 3D-software en game-engines te bieden hebben. Het van de grond opbouwen van 3D visualisatiesoftware of zelfs reeds bestaande software aanpassen is een enorme taak en slechts weinig wetenschappers hebben hier de tijd noch de technische knowhow voor. Daarnaast is het moeilijk om programmeurs te vinden die de nodige kennis bezitten over kristallografie en kristalstructuren of zich hierin willen verdiepen.

Het gebruiken van Blender als software om kristallen te visualiseren is een stap in de goede richting. Deze open source software laat, mits enige kennis van het programma, toe driedimensionale figuren te creëren en te bekijken. Het groot aantal features in Blender biedt de gebruiker veel vrijheid aan, wat ontbreekt in hedendaagse kristalvisualisatiesoftware. 3D objecten aanmaken en bekijken doet men via de grafische interface van Blender of met behulp van een script. Het gebruiken van scripts is erg interessant voor de gebruiker omdat deze hiermee langdradige of repetitieve taken kan laten uitvoeren.

In de context van kristallografie laat dit toe dat een kristal, bestaande uit een groot aantal atomen, niet meer manueel moet getekend worden. Helaas biedt dit geen volledige oplossing voor het probleem, voor elk kristal moet er nog steeds een apart script worden geschreven waarin alle informatie van dat kristal staat. Het scripten in Blender wordt gedaan aan de hand van Python en de API van Blender (zie verder). Hierdoor is het schrijven van scripts niet vanzelfsprekend en zonder enige voorkennis onbegonnen werk.

1.3 Doelstelling

Het algemene doel van dit werk is het ontwerpen van een interface die kristalstructuren kan visualiseren in het voornoemde open source programma, Blender. Eens dit bereikt is, zijn de mogelijkheden virtueel eindeloos.

Een van de problemen bij het visualiseren van een kristal met scripten in Blender is dat voor elke kristalstructuur een nieuw script moet worden geschreven. De interface moet gezien worden als een black box met als input de beschrijving van een kristalstructuur en de driedimensionale voorstelling hiervan als output. Dit stelt de nood aan een input die interpreteerbaar moet zijn door een computer en die in eerste instantie in eenzelfde formaat is om de omvang van het programma te beperken. Dit kan nadien nog uitbereid worden zodat verschillende formaten kunnen worden ingelezen. De eerste doelstelling is dus een keuze te maken van formaat dat er kan ingelezen worden door de interface.

De volgende stap in het ontwerpproces van de interface is het schrijven van een functie die in staat is het eerder gekozen formaat in te lezen en om te zetten naar bruikbare data. Afhankelijk van de complexiteit en striktheid van het formaat kan het ontwerpen van dit soort routines erg tijdrovend zijn. Het is mogelijk dit te vermijden door op zoek te gaan naar een reeds bestaande Python modules met een gelijkaardige werking en deze, indien mogelijk, te implementeren in de interface.

Eens het formaat kan worden ingelezen, moet nodige informatie worden opgeslagen. Python biedt de mogelijkheid aan om gebruik te maken van classes, wat toelaat de kristaldata op te slaan in zelfgecreëerde datastructuren, wat de uiteindelijke dataverwerking zal vereenvoudigen.

De Blender API geeft de gebruiker een keuze uit een immens aantal functies. Dit zorgt er echter voor dat scripten in Blender zonder de nodige voorkennis erg complex kan worden. Een belangrijk onderdeel van deze scriptie is dan ook het verdiepen in de API van Blender en alle mogelijkheden die deze biedt. Met de nodige kennis van de functies wordt het mogelijk de gemaakte datastructuren driedimensionaal te visualiseren in Blender. Enkele basisfeatures die de interface moet hebben is een manier om elementen te onderscheiden met behulp van kleur, automatisch bindingen maken tussen atomen op basis van hun onderlinge afstand en meerdere eenheidskristallen naast elkaar kunnen tekenen.

Ten slotte zullen de limieten van Blender getest worden in de context van het wetenschappelijk onderzoek rond kristallen.

1.4 Onderzoeksvragen

De doelstellingen uit vorige sectie kunnen worden geformuleerd als onderstaande onderzoeksvragen.

Hoofdvraag:

Is het mogelijk een interface te ontwerpen die in staat is kristalstructuren driedimensionaal te visualiseren in Blender?

Deelvragen:

Wat is de meest efficiënte methode om een kristalstructuur om te zetten in verwerkbare data?

Wat is de beste manier om kristaldata te visualiseren in Blender?

(extra) Hoe ver kan Blender gepushed worden als 3D visualisatiesoftware voor kristallen?

1.5 Structuur van de tekst

Het eerste hoofdstuk geeft een inleiding tot het algemene onderwerp van deze thesis. Hier zal onder andere de probleemstelling en het doel van dit onderzoek worden besproken. Dit hoofdstuk heeft als doel een kort overzicht te geven over de verdere inhoud van deze scriptie.

In het tweede hoofdstuk wordt de literatuurstudie besproken. Dit onderdeel van de tekst verdiept zich in kristallografie, het gebruikte formaat van kristalbeschrijving, reeds bestaande visualisatie software, de parser en tot slot Blender en de Blender API. Het doel van dit hoofdstuk is inzicht geven in het theoretische aspect van de thesis en de nodige kennis verschaffen over de gebruikte technologieën.

In het derde hoofdstuk wordt het ontwerpproces beschreven. Hier kunnen de stappen worden gevolgd die zijn ondernomen in het opbouwen van de interface. Er wordt op een oppervlakkige manier gekeken naar de opbouw van het programma om het globale overzicht te behouden. Dit hoofdstuk schetst de algemene structuur van het programma en de gedachtegang tijdens het ontwerpen hier-

van. De inhoud van het vierde hoofdstuk beschrijft de structuur van de ontworpen interface en hoe deze juist werkt. Deze tekst geeft een technische kijk op het programma en overloopt bepaalde onderdelen van de geschreven code in meer detail. Er wordt onder andere dieper ingegaan op de werking van de Blender API en de CIF-parser.

Hoofdstuk vijf kijkt in detail naar de output van het programma. Er wordt dieper ingegaan op de resultaten, complicaties, mijlpalen en gemaakte fouten. De eindconclusie van de thesis wordt besproken in hoofdstuk zes. Alle relevante informatie uit voorgaande hoofdstukken wordt hier opgesomd om een duidelijk overzicht te geven over het volledige onderzoek. De resultaten worden in dit hoofdstuk nogmaals kort overlopen om zo tot een uiteindelijk besluit te komen betreffende dit eindwerk.

1.6 Conclusie

Dit hoofdstuk gaf een overzicht over het onderwerp en de omvang van deze thesis, kaartte de probleemstelling aan en formuleerde enkele onderzoeksvragen die als rode draad doorheen deze tekst dienen. Met de kennis uit dit hoofdstuk zal de stap naar zowel de literatuurstudie als naar het meer praktische onderdeel van dit onderzoek minder groot zijn.

Hoofdstuk 2

Literatuurstudie

Vooraleer een oplossing kan worden ontworpen voor een probleem is het van belang de nodige informatie te verzamelen over de huidige staat van het probleem en eventuele reeds bestaande oplossingen. In dit hoofdstuk zal er onder andere worden gekeken naar de technologieën die op dit moment in de wetenschap gebruikt worden en welke technologieën mogelijk een oplossing kunnen bieden voor het probleem.

Het lezen van dit hoofdstuk is aangeraden aangezien het enkele belangrijke begrippen en technologieën beschrijft waarnaar zal worden verwezen in dit onderzoek. Het is echter mogelijk dit hoofdstuk over te slaan als deze informatie reeds gekend is of als de theorie achter het proces minder van belang is.

Omdat dit onderzoek zich deels afspeelt binnen de wereld van kristallografie, een specifieke tak van de wetenschap, zal de eerste sectie van dit hoofdstuk gewijd worden aan het overlopen van enkele begrippen die van belang zullen zijn in het verdere verloop van het onderzoek. Daar kristallografie een eerder complexe wetenschap is zal deze tekst een vereenvoudigde beschrijving zijn. Voor een meer accurate beschrijving wordt er verwezen naar de literatuur waarop deze sectie gebaseerd is. De tweede sectie zal het Crystallographic Information File of CIF-formaat beschrijven. Dit tekstformaat ligt aan de basis van het digitaliseren van kristalstructuren en zal gebruikt worden bij het inlezen van kristaldata. In de derde sectie wordt gekeken naar welke programma's op dit moment worden gebruikt bij het visualiseren van kristallen en hoe deze werken. De vierde sectie kijkt naar enkele reeds bestaande CIF-parsers en of deze al dan niet kunnen gebruikt worden in dit onderzoek. Sectie vijf zal zich verdiepen in de werking van Blender en de Blender API. In de zesde en laatste sectie wordt een conclusie getrokken uit de verkregen informatie.

2.1 Kristallografie

2.1.1 Wat is kristallografie

Kristallografie kan beschreven worden als de studie van materie in een kristallijne staat en houdt zich onder andere bezig de synthese en opbouw van kristallen en hun fysische en chemische ei-

Biologie	Chemie	Farmacologie	Geologie
proteïnes	Rubber	alle vaste medicijnen	alle mineralen
polysacharides	benzeen	vitamines	metalen
beenderen	naftaleen		

Tabel 2.1 Kristallen in wetenschappelijke takken

enschappen, de synthese van kristallen en hoe deze worden opgebouwd. Eerder in dit hoofdstuk werd kristallografie beschreven als een specifieke tak van de wetenschap, het is echter vermeldenswaardig dat dit een erg omvattende studie is. Kristallen komen namelijk voor in vrijwel elk andere wetenschappelijk domein. Tabel[2.1] toont enkele voorbeelden van kristallen en de respectievelijke tak van de wetenschap waar deze onder vallen.

In dit onderdeel zullen verder enkel de kristallografische begrippen worden beschreven die binnen de omvang van deze thesis vallen.

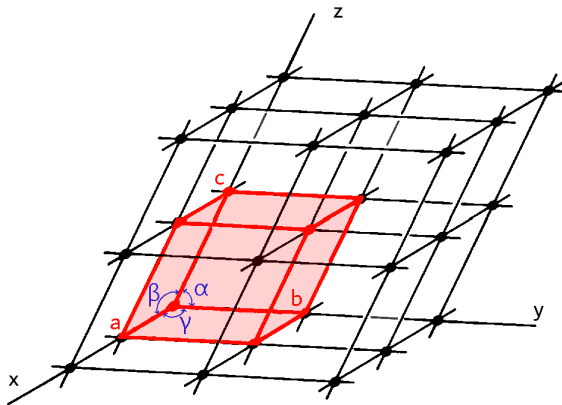
2.1.2 De eenheidscel

Alle kristallen zijn gedefinieerd als een periodische schikking van atomen, ionen of moleculen. Het opbouwen van een kristal wordt gedaan aan de hand van roosterpunten. Een aantal roosterpunten op een lijn met een gelijke onderlinge afstand wordt een roosterlijn genoemd, een verzameling van evenwijdige roosterlijnen met eenzelfde onderlinge afstand is een roostervlak. Ten slotte kan dit worden herhaald in een derde dimensie en wordt er een roosterruimte gevormd, welke in zwart getekend staat op figuur[2.1]. Een roosterruimte kan in essentie zo groot zijn als het beschreven kristal. Zoals eerder ook gezegd is een kristal periodisch, dit wil zeggen dat er enkel moet gekeken worden naar het kleinste unieke volume van een kristal. Dit uniek volume valt volledig binnen de roosterruimte beschreven door de eerste acht roosterpunten. Het volledige kristal kan worden opgebouwd door de translatie van deze eenheidscel in 3 richtingen in de ruimte. Een kristal bestaat gewoonlijk uit een groot aantal van deze eenheidscellen.

Geometrisch gezien bevat een eenheidscel 3 paar evenwijdige parallellogrammen als omhullende vlakken en kan het beschreven worden aan de hand van 6 variabelen, welke de roosterparameters worden genoemd. De parameters a , b en c bepalen de lengte tussen de roosterpunten volgens de drie assen van de roosterruimte. De drie resterende variabelen worden gebruikt om de hoeken tussen deze assen te beschrijven en worden γ , β en α genoemd. Deze variabelen staan ook aangeduid op figuur[2.1] met de eenheidscel in rood.

2.1.3 Classificatie van kristalroosters

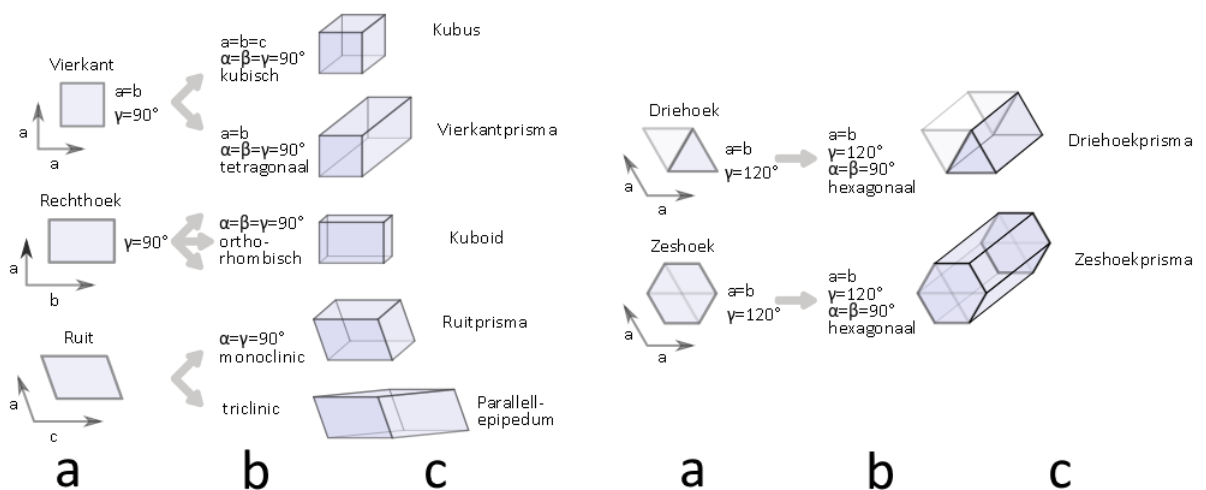
De vorm van de eenheidscel van een kristal is niet willekeurig, omdat dit volume de ruimte volledig moet kunnen vullen. Mogelijke types van eenheidscellen kunnen worden herleid op basis van de vorm van het vlak dat deze beschrijft, figuur[2.2.a]. De driedimensionale eenheidscel kan dan worden verkregen door de extrusie van dit vlak in de derde dimensie, figuur[2.2.c]. Zo bestaan er exact



Figuur 2.1: Voorstelling van een eenheidscel in de roosterruimte (Borchardt-Ott, 2011)

zes coördinaatsystemen, gebaseerd op de restricties van de roosterparameters. Deze vormen de basis voor de classificatie in kristalfamilies en krijgen de naam: kubisch, tetragonaal, orthorhombisch, monoclinisch, triclinisch, en hexagonaal. Deze worden weergegeven in figuur[2.2.b].

Vertrekkende van de mogelijke coördinaatsystemen en rekening houdend met mogelijke symmetrieën ontstaan er zeven kristalsystemen, figuur[2.3.a]. Hierbij wordt het hexagonale coördinatenstelsel opgesplitst in het trigonale en hexagonale kristalsysteem, naar mate er een drie- of een zesvoudige symmetrie aanwezig is. De zeven resulterende primitieve eenheidscellen beschrijven alle mogelijke eenheidstranslaties in richtingen a, b en c voor de zeven kristalsystemen. Een atoom of molecule op een positie x, y, en z zal ook altijd aanwezig zijn in richting a, b, en c als de afstand identiek is aan een veelvoud van de eenheidstranslaties in deze richtingen.



Figuur 2.2: De 6 kristalfamilies

De Franse natuurkundige Bravais (Britannica, 2018) geeft een extra indeling op basis van mogelijke translaties binnen de eenheidscel, welke in overeenkomst met de symmetrievorwaarden van de kristalsystemen zijn. Dit leidt tot de 14 Bravais roosters, ook wel gecentreerde roosters genoemd

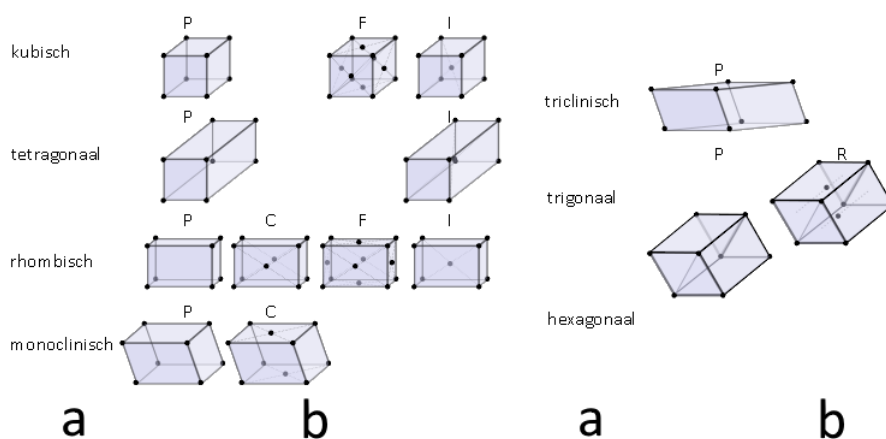
welke worden beschreven met een letter. De eerste wordt de primitieve genoemd en krijgt de letter P toegekend. Bij deze bestaan alleen de eenheidstranslaties. Het rooster bevat alleen roosterpunten op de 8 hoeken van de cel.

Het tweede gecentreerde rooster noemt men grondvlakgecentreerd. Deze heeft naast de eenheidstranslaties ook een centrering in een vlak. Elk deeltje kan ook teruggevonden worden over een translatie van een halve lengte langs twee assen. Er kan sprake zijn van A, B of C centrering als de verschuiving langs de diagonaal van respectievelijk het bc-, ac- of ab- vlak aanwezig is.

Wanneer er extra roosterpunten liggen in elk van voorgaande vlakken spreekt men van een vlakgecentreerd of F-rooster.

Ten slotte kan er zich, naast de roosterpunten op de hoeken, ook een in het centrum van het rooster bevinden, in dit geval wordt er van een ruimtegecentreerd of I-rooster gesproken.

In het geval van het trigonale kristalsysteem bestaat er een speciale vorm van centrering in overeenkomst met de drievoudige symmetrie, welke R-centrering genoemd wordt. Figuur[2.3.b] geeft de 14 resulterende Bravaisroosters weer, dit zijn alle mogelijke vormen dat een kristal kan aannemen.



Figuur 2.3: De 14 bravaisroosters en hun kristalsystemen

2.1.4 Ruimte- en puntgroepen

Dit concept is vrij complex en minder van belang voor dit onderzoek. Omdat gerelateerde termen in het verloop van deze tekst aan bod komen is het toch best deze beknopt toe te lichten. In deze literatuurstudie zal de exacte theorie hierachter niet in beschreven worden, maar enkel wat er van belang is voor het begrijpen van deze thesis.

Puntgroepen wijzen op de innerlijke symmetrie van een roosterpunt. Een puntgroep kan gezien worden als een verzameling van symmetrieoperaties. Een symmetrieoperatie wordt als een manipulatie van een object beschouwd welke het object in zijn oorspronkelijke vorm transformeert. Naast de identiteit van een object met zichzelf zijn er vier verschillende soorten van symmetrieoperaties: spiegeling in een vlak, rotatie rond een as, inversie en de combinatie van een spiegeling en een inversie, wat een draai-inversie wordt genoemd. In totaal bestaan er 32 kristallografische

puntgroepen, welke in overeenkomst zijn met de symmetrievoorwaarden van de 7 reeds besproken kristalssystemen.

Ruimtegroepen kijken niet enkel naar innerlijke symmetrie maar ook naar de symmetrie van de kristalstructuren en worden verkregen door de puntgroepen toe te passen op de 14 reeds gekende Bravaisroosters. Dit leidt tot in theorie 448 combinaties, maar door de gelijkheid tussen sommige is dit nummer terug te brengen naar een totaal van 230 ruimtegroepen. Dit heeft als gevolg dat er 230 verschillende manieren bestaan om deeltjes te ordenen in een eenheidscel. Als gevolg kan elk kristal altijd in een van deze ruimtegroepen beschreven worden.

2.1.5 Beschrijving van een kristal

Met voorgaande informatie in het achterhoofd is het mogelijk een kristalstructuur te beschrijven. De enige informatie die hiervoor nodig is, is de ruimtegroep, het gebruikte coördinaatsysteem en de mogelijke centrering dat alle symmetrieoperaties definieert. Vervolgens worden de roosterparameters van de eenheidscel bepaald en ten slotte wordt een lijst gegeven van alle atomen die niet door het uitvoeren van de gegeven symmetrieoperaties gegenereerd kunnen worden. Deze symmetrieonafhankelijke atomen in hun elementaire cel worden een asymmetrische eenheid genoemd en is voldoende om elk kristal af te beelden en te beschrijven.

2.2 Het Crystallographic Information File (CIF) formaat

2.2.1 Het STAR formaat

Om een beter idee te krijgen van de syntax van het CIF-formaat worden eerst de onderliggende concepten beschreven van het Self-Defining Text Archive and Retrieval (STAR) bestandsformaat, waarop het CIF-formaat gebaseerd is.

Het STAR-formaat was de eerste stap naar de digitalisering van data. (Hall et al., 1991) Dit formaat bestaat uit ASCII tekst, wat makkelijke aanpassing en leesbaarheid toelaat door zowel mensen als computers. De opbouw van dit formaat ziet er uit als volgt: een file kan bestaan uit een of meerdere datablokken, welke nogmaals onderverdeeld zijn in een sequentie van data-elementen. De identiteit van een data- element wordt bepaald door een unieke naam die ervoor wordt geplaatst in het bestand. Het kleine aantal regels in een STAR-bestand maakt dit een eenvoudig en veelzijdig formaat. Zo zijn er geen restricties op de volgorde waarin de data moet worden genoteerd en moet er geen kennis zijn van het datatype van een element. Het gebruik van een lus maakt het mogelijk de toekenning van data- elementen te herhalen.

2.2.2 De CIF-notatie

Het CIF-formaat bouwt verder op de syntax van het reeds besproken STAR-formaat met een aantal extra voorwaarden. Enkel de datanamen die beschreven worden in de CIF-dictionary [Bijlage A] worden toegelaten, deze woordenboek bevat dan ook alle nodige parameternamen die nodig zijn

om een kristal te beschrijven. De datanamen in het CIF-woordenboek zijn opgesteld door de Internationale Unie van Kristallografie (IUCr) en zijn dus algemeen aanvaard. Een lijn in het CIF-formaat mag niet langer zijn dan 80 karakters en een datanaam niet langer dan 32, doordat in de CIF-dictionary datanamen worden opgedeeld op basis van het item dat ze beschrijven is dit echter geen probleem. Net zoals bij de STAR-notatie is er geen verplichting tot het toekennen van datatypes, dit wordt bij het CIF-formaat echter wel aangeraden om de datawerking vlotter te laten verlopen. Zo worden data-elementen gezien als nummers wanneer ze beginnen met een cijfer. Een nummer kan een geheel getal, een kommagetal of als wetenschappelijke notatie worden gegeven. In het CIF-woordenboek worden ook de standaardeenheden van de data-elementen bijgehouden, de afmetingen van een eenheidscel worden verondersteld in Ångström te zijn. De Ångström is een eenheid die vaak gebruikt wordt in kristallografie en heeft de waarde van 0,1 nanometer. Data wordt beschouwd als tekst wanneer deze langer is dan één lijn. In het geval dat een data-element noch een nummer noch tekst is wordt het beschouwd als een karakter.

2.2.3 Voorbeeld van een CIF-bestand

Dit onderdeel wordt aan de hand van listings uit een CIF-bestand de betekenis van de vaak voorkomende termen uitgelegd. Dit is geen bestaand kristal en dient enkel als voorbeeld, het volledige bestand kan gevonden worden in de bijlages [Bijlage B].

data_CHA

1

Listing 2.1:

De naam van het datablok.

#

1

CIF taken from the IZA-SC Database of Zeolite Structures

2

Ch. Baerlocher and L.B. McCusker

3

#

4

Listing 2.2:

Alles achter een hashtag is commentaar en is enkel zichtbaar voor de lezer.

_cell_length_a

13.6750(0)

1

_cell_length_b

13.6750(0)

2

_cell_length_c

14.7670(0)

3

_cell_angle_alpha

90.0000(0)

4

_cell_angle_beta

90.0000(0)

5

<code>_cell_angle_gamma</code>	120.0000(0)	6
--------------------------------	-------------	---

Listing 2.3:

De roosterparameters, de lengtes van de ribbes in Ångström en de hoeken tussen de assen in graden, een getal tussen haakjes achteraan wijst naar de geschatte standaardafwijking van het getal.

<code>_symmetry_space_group_name_H-M</code>	'R ₃ m'	1
<code>_symmetry_Int_Tables_number</code>	166	2
<code>_symmetry_cell_setting</code>	trigonal	3

Listing 2.4:

De notatie van de ruimtgroep, het nummer van de ruimtgroep en het kristalstelsel van de eenheidscel.

<code>loop_</code>	1
<code>_symmetry_equiv_pos_as_xyz</code>	2
' x, y, z '	3
' $2/3+x, 1/3+y, 1/3+z$ '	4
' $1/3+x, 2/3+y, 2/3+z$ '	5

Listing 2.5:

Deze lus geeft de posities waarop de atoomgroep worden geplaatst, deze lijst is vaak erg lang.

<code>loop_</code>	1
<code>_atom_site_label</code>	2
<code>_atom_site_type_symbol</code>	3
<code>_atom_site_fract_x</code>	4
<code>_atom_site_fract_y</code>	5
<code>_atom_site_fract_z</code>	6
O1 O 0.9020 0.0980 0.1227	7
O2 O 0.9767 0.3101 0.1667	8
T1 Si 0.9997 0.2264 0.1051	9

Listing 2.6:

Deze lus beschrijft de groep van atomen en waar deze zich bevinden in het kristal. De eerste twee data-elementen bepalen respectievelijk het zelfgekozen symbool voor het element en de wetenschappelijke afkorting van het chemisch element. De drie laatste waarden geven de relatieve positie van deze atomen binnen de eenheidscel ten opzichte van de lengte van de ribben.

2.2.4 Kristallografische databanken

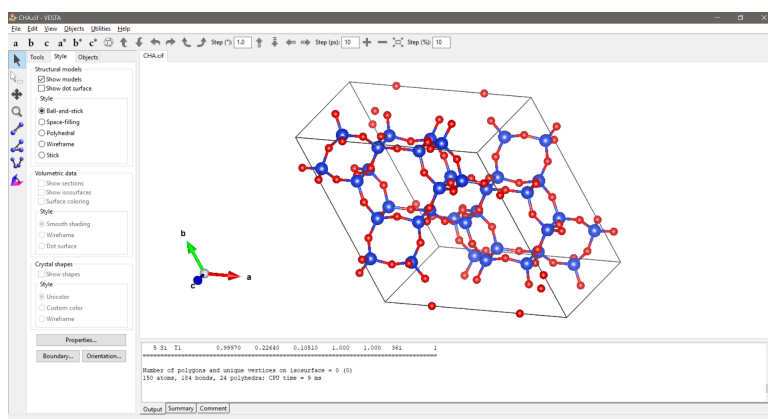
Aangezien het CIF-formaat gezien kan worden als een standaardformaat voor het beschrijven van kristallen bestaan er reeds CIF-bestanden voor vrijwel alle bestaande kristalstructuren. De ver-

zameling van deze bestanden, alsook andere formaten, worden kristallografische databanken genoemd en staan online ter beschikking op verschillende websites. De lijst met kristalstructuren blijft groeien en wordt bijgewerkt door verschillende organisaties en universiteiten, een voorbeeld hiervan is de Crystallography Open Database (COD) waar de kristalstructuren van onder andere organische, anorganische en metaalorganische samenstellingen kunnen teruggevonden worden. Het voorbeeldbestand in vorige sectie is een deel van de beschrijving van de kristalstructuur van het zeoliettype Chabaziet, dit CIF-bestand kan teruggevonden worden in de zeolietstructuurdata-bank op de website van de International Zeolite Association (IZA) welke zich uitsluitend bezighoudt met het onderzoeken en beschrijven van de verschillende types van het mineraal zeoliet.

2.3 Kristalvisualisatiesoftware

2.3.1 Visualization for Electronic and Structural Analysis (VESTA) 3

Als een van de meest frequent gebruikte 3D visualisatieprogramma's biedt VESTA 3 een grotere verscheidenheid aan features en voordelen dan zijn voorgangers en andere, gelijkaardige programma's doen. VESTA beschrijft zichzelf als een gratis te gebruiken 3D visualisatiesysteem voor structurele modellen, volumetrische data en kristalmorfologieën. (Momma and Izumi, 2008) De software wordt ondersteunt op Windows, Mac en Linux. Geschreven in de programmeertaal C met het gebruik van de OpenGL technologie en een modern C++ GUI framework biedt VESTA een gebruikersvriendelijke interface met de mogelijkheid tot het roteren, transleren en schalen van het getekende object alsook het hierop plaatsen van fysieke gegevens zoals elektronendichtheden, nucleaire dichtheden, golffuncties en elektrostatische potentialen. Hiernaast bestaan er verschillende manieren het object weer te geven. De omvang van deze objecten zijn in theorie oneindig en wordt slechts beperkt door de hardware van de computer, hoe groter het grafische rekenvermogen van deze hoe vlotter het zal functioneren. De grafische interface van VESTA wordt weergegeven op figuur[2.4]. VESTA is in staat verschillende bestandsformaten om te zetten naar een driedimensionale voorstelling, zoals het CIF-formaat. VESTA laat toe nieuwe modellen aan te maken of bestaande modellen aan te passen. Gecreëerde objecten kunnen ten slotte geëxporteerd worden in de meest voorkomende 3D formaten zodat ze in andere 3D software, zoals Blender, kunnen worden bekeken.

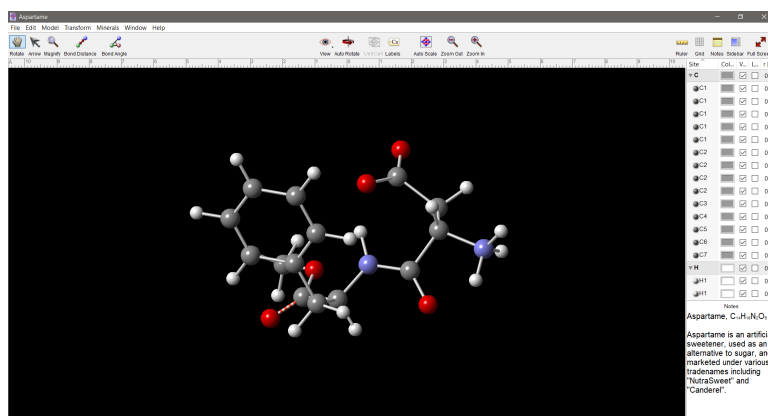


Figuur 2.4: De user interface van VESTA3

Omdat de broncode van VESTA niet openbaar beschikbaar is, is het vrijwel onmogelijk eigen features toe te voegen, en hangt de gebruiker vast aan de tools die deze software ter beschikking stelt. Deze beperking aan vrijheid heeft ertoe geleid dat wetenschappers op zoek gaan naar andere software die dit wel aanbiedt.

2.3.2 Andere 3D visualisatiesoftware

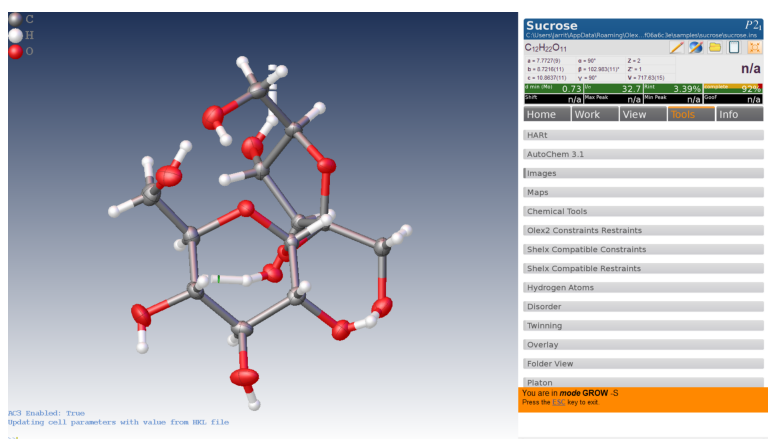
Naast VESTA bestaat er nog een groot aanbod aan andere software die in staat zijn kristallen driedimensionaal te visualiseren. Het Crystallmaker softwarepakket biedt het programma Crystalviewer, figuur[2.5], gratis aan voor persoonlijk gebruik. Het strak design en de gebruiksvriendelijke user interface geven de software een premium gevoel. Voor het gebruik van het volledig potentieel van dit programma en om toegang te krijgen tot de documentatie wordt de gebruiker ertoe gedwongen het eerder dure, volledige softwarepakket aan te kopen. Hiernaast biedt de gratis versie niet de mogelijkheid andere formaten in te lezen dan het .crystal – formaat, wat niet gezien wordt als standaardformaat en niet bestaat in de kristallografische databanken.



Figuur 2.5: De user interface van Crystalviewer9

Een andere bekende tegenhanger van VESTA is Olex², zie figuur[2.6]. Olex² biedt net zoals VESTA

een groot aantal features aan, welke duidelijk beschreven worden in een overzichtelijke documentatie. Het softwarepakket is volledig gratis en ter beschikking van iedereen. Olex² is volledig geschreven in Python en maakt gebruik van enkele bibliotheken die de gebruiker de mogelijkheid biedt verschillende kristalbeschrijvingsformaten, waaronder CIF, in te lezen te visualiseren. Ondanks de features en documentatie is het gebruiken van Olex² minder eenvoudig dan de eerdere geziene programma's. Bij het testen van dit programma trad er ook een probleem op bij het visualiseren van een CIF-bestand, wat deze software voor deze toepassing vrijwel onbruikbaar maakt.



Figuur 2.6: De user interface van Olex²

2.4 CIF-Parsers

2.4.1 Wat is een parser

Een parser wordt gezien als een computerprogramma of script dat in staat is een input om te zetten naar een datastructuur met als voorwaarden dat deze input volgens een bepaalde structuur, denk XML, is ingegeven. De parser gaat in de ingevoerde gegevens op zoek naar herkenbare elementen en slaat deze op volgens een eerder genoemde datastructuur, bijvoorbeeld een boomstructuur of een dictionary. Sommige parsers worden gebruikt om ingegeven data te visualiseren, deze data kan dan worden gevisualiseerd in de vorm van een boomstructuur of een grafiek. Een ander soort van parsers heeft als nut de gegevens van het bestand om te zetten in een dictionary, waardoor deze in een ander programma eenvoudig kunnen worden opgevraagd. In dit onderzoek is er nood aan dit tweede type van parser, zodat de gegevens in een CIF-bestand kunnen worden omgezet naar bruikbare data.

2.4.2 Het schrijven van een parser

Om een parser te schrijven voor een bepaald bestandstype of formaat is er eerst een goede kennis van dit formaat. Hoewel het op eerste zicht misschien niet logisch lijkt, wordt het schrijven van een parser eenvoudiger naarmate de striktheid van het formaat stijgt. Er moet minder rekening

gehouden met alle mogelijke variaties die in het bestand kunnen voorkomen. Zoals in sectie twee besproken werd bestaan er in het CIF-formaat weinig regels op vlak van tekststructuur. Wat ervoor zorgt dat het schrijven van een CIF-parser veel werk vergt. In dit onderzoek is er aanvankelijk geprobeerd een eigen parser te ontwerpen die kan gebruikt worden om CIF-bestanden om te zetten naar bruikbare data. Dit proces wordt in hoofdstuk drie in meer detail besproken. Omdat dit echter niet het doel is van dit onderzoek is er besloten te kijken naar de mogelijkheden die reeds bestaande CIF-parsers kunnen bieden. De bekeken parsers worden verder beschreven.

2.4.3 The Computational Crystallography Toolbox (cctbx)

Dit project heeft als voornaamste doel een brug te bouwen tussen kristallografische data en het gebruik hiervan in computer gerelateerde toepassingen. Het project is volledig open source, wat wil zeggen dat het volledig toegankelijk en gratis is voor iedereen die het wil gebruiken, hiernaast zorgt dit ervoor dat dit project constant wordt verbeterd en stijgt in functionaliteit. Deze toolbox ligt ook aan de basis van de visualisatiesoftware Olex², welke beschreven wordt in de derde sectie van dit hoofdstuk. De grote omvang van dit project en het grote aantal functionaliteiten maakt het, ondanks de duidelijke documentatie, ook meer ingewikkeld dit te gebruiken.

2.4.4 Python CIF Read/Write (PyCIFRW)

Naast het inlezen van CIF-bestanden kunnen deze met dit programma ook gecreëerd en aangepast worden. Ook dit programma is open source, omdat dit op een kleinere schaal gebeurt dan cctbx, zie vorige, zijn de mogelijkheden, hoewel meer beperkt, geschikter voor gebruik in dit onderzoek. Door de goede documentatie van dit project en de volledige ondersteuning van Python kan dit programma gecombineerd worden met de, in de volgende sectie besproken, Blender API. (IUCR, 2009) Voor deze thesis is de schrijffunctie minder van belang en zal niet worden bekeken in deze sectie.

```

import CifFile 1
2
#Het inlezen van een CIF-bestand in een variabele wordt gedaan met de 3
functie:
ciffile = CifFile.ReadCif("bestandsnaam.cif") 4
#Vervolgens kunnen datablokken worden ingelezen: 5
datablok = cf["a_data_block"] 6
#Of kan een data-element rechtstreeks worden aangesproken met 7
dataelement = cf["een_datablok"]["een_datanaam"] 8
#De data van loop of lusblok kan worden opgevraagd 9
lb = cb.GetLoop("een_datanaam uit de loop") 10

```

Listing 2.7: Werken met PyCIFRW in Python3

Listing[2.7] is een voorbeeld van hoe PyCIFRW kan worden gebruikt in Python om een CIF-file in te lezen. De voornaamste functies worden hier weergegeven. Een meer gedetailleerde beschrijving van de werking van PyCIFRW kan worden teruggevonden in hoofdstuk 4.

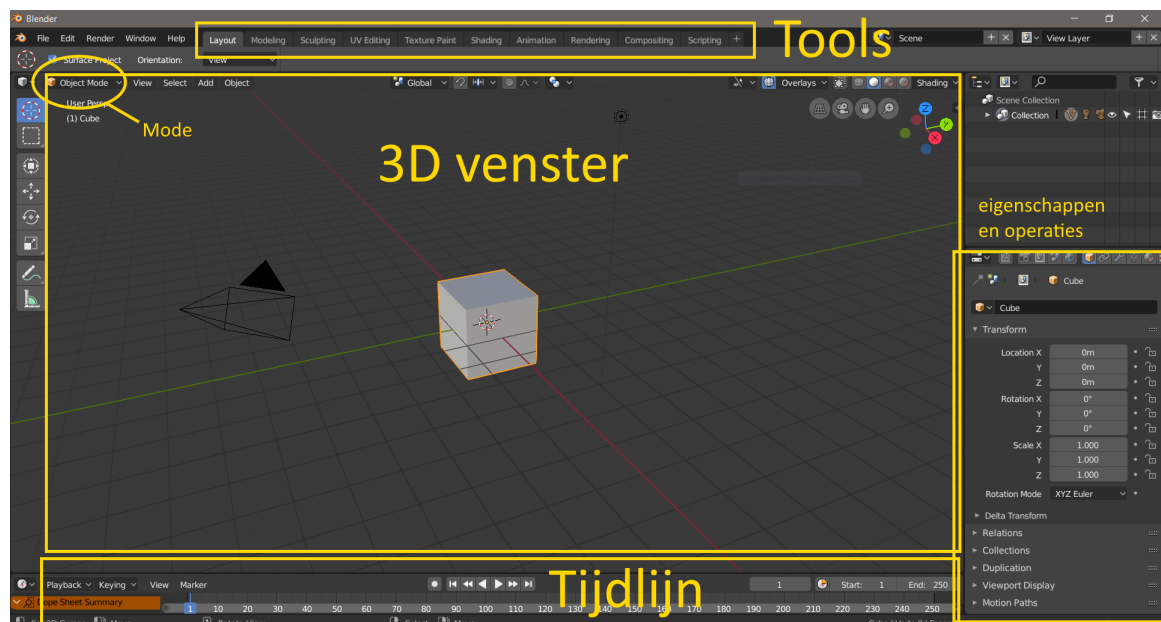
2.5 Blender en de Blender API

2.5.1 Gratis software

Blender beschrijft zichzelf als een „open-sourced, community development program”, en is het collectieve werk van softwareontwikkelaars overal ter wereld. (Chronister, 2011) De ontwikkeling en het correct functioneren van dit programma wordt overzien door The Blender Foundation, een Nederlandse, onafhankelijke, non-profit stichting. Dit alles leidt ertoe dat de Blender software gratis verkrijgbaar is voor iedereen die het wenst te gebruiken.

2.5.2 De Blender interface

Na het opstarten van Blender wordt de gebruiker begroet met de Blender interface. Een niet ervaren gebruiker zal naast, onder de indruk, vooral in de war zijn en niet weten waar te beginnen. In figuur[2.7] wordt het startscherm weergegeven en worden enkele onderdelen ervan beschreven. Veel van deze tools en hoe deze werken vallen niet binnen het bestek van deze tekst en worden weinig of niet uitgelegd, hiervoor wordt verwezen naar online handleidingen en de documenten waar deze tekst op gebaseerd is.



Figuur 2.7: Het Blender startscherm

In het midden van het scherm kan het 3D venster worden teruggevonden. Hier worden alle ob-

jecten weergegeven die zich op de huidige ingeschakelde layers bevinden. Het gebruik van layers laat toe objecten op een andere layer tijdelijk te verbergen om zo een duidelijk overzicht te behouden over het huidige werk. Wanneer een object wordt aangemaakt zal deze automatisch geplaatst worden op de huidige layer. De Blender interface heeft twee modes: object mode en edit mode.

In object mode kan een object in zijn geheel bewerkt worden. Transleren, roteren en herschalen zijn slechts enkele van het groot aantal aan objectoperaties dat Blender aanbiedt.

Edit mode laat de gebruiker onder andere toe de vorm van het object aan te passen door hoekpunten aan te duiden en te verplaatsen. Objecten toevoegen wordt gedaan met de Add knop linksboven in het venster. In Blender gebruikt men de rechtermuisknop om items te selecteren en te verplaatsen. Het klikken op de linkermuisknop verplaatst de 3D cursor, dit is de plaats waar nieuwe objecten terechtkomen wanneer ze worden aangemaakt. Ten slotte kan in het 3D venster worden bewogen door het indrukken van het muiswiel.

Met deze informatie kan er een object aangemaakt en verplaatst worden en kunnen er enkele basisoperaties op worden toegepast. Dit is voorlopig alles wat er over het bewerken van objecten gezegd zal worden. Hou echter in het achterhoofd dat dit slechts het topje is van de enorme ijsberg die Blender wordt genoemd.

2.5.3 Kleuren, texturen en materialen

Blender laat de gebruiker toe de gemaakte objecten, naast vorm, ook kleur te geven. Eerst krijgt het object een materiaal toegekend, waarna de eigenschappen van dit materiaal kunnen worden aangepast. Deze eigenschappen zijn onder andere de kleur, lichtuitstraling, weerspiegeling, schaduw en doorzichtigheid van het object. Om het object er nog realistischer te laten uitzien kan er een textuur op het object worden gezet. Deze geven het object een bepaalde oppervlaktestructuur of uiterlijk. Blender voorziet een aantal basistexturen maar de gebruiker kan zelf afbeeldingen gebruiken om het voorwerp een textuur te geven. Wanneer de objecten een kleur of textuur hebben kan de gebruiker deze renderen. Dit gaat alle lichtinteracties en schaduws berekenen, wat erg computerintensief kan zijn. In figuur[2.8] wordt de uitkomst van het renderen van een Blender project weergegeven, door slim gebruik van texturen en materiaaleigenschappen was de ontwerper in staat een 3D tekening levensecht te laten lijken. (Blenderguru, 2012)

2.5.4 Andere opmerkelijke tools in Blender

Naast het aanmaken, vervormen en inkleuren van objecten biedt Blender nog een handvol extra mogelijkheden aan. Met de Sculpting tool kan men objecten boetsen tot in de kleinste details. Dit kan leiden tot realistische beeldhouwwerken.

Onderaan de interface afgebeeld op figuur[2.7] kan het tijdlijn-venster gezien worden. Dit venster behoort tot de animation tool van Blender, door een aantal gerenderde frames in snelle successie achter elkaar te laten afspelen wordt een video verkregen. Kleine verschillen tussen deze frames geven de indruk dat het object beweegt, dit wordt een animatie genoemd.

Hoewel er betere software bestaat om dit te doen, is het in Blender, door gebruik te maken van



Figuur 2.8: Het Blender project van de winnaar van een fotorealiteit wedstrijd, na rendering

scripts, mogelijk volledig functionele 3D videogames te ontwerpen. Zo bestaan er reeds games die volledig in Blender zijn gemaakt en te koop staan op Steam, een bekende videogames-distributeur.

2.5.5 Scripten in Blender met de Blender API

Nog een belangrijke functie die Blender aanbiedt, en in de vorige sectie niet werd vermeld, is de mogelijkheid tot het scripten in Blender zelf, in tegenstelling tot de game-engine. Omdat deze feature in zekere mate de rode draad is in deze thesis wordt er een volledige subsectie gewijd aan dit eerder ingewikkelde onderdeel van Blender.

In de tweede sectie van dit hoofdstuk werd besproken hoe operaties in Blender kunnen uitgevoerd worden met behulp van de Blender interface. Er bestaat echter een andere manier om operaties uit te voeren, namelijk via scripts. Met de juiste kennis kan er met behulp van scripts zelfs meer gedaan worden dan met enkel de interface. Het schrijven van scripts wordt gedaan aan de hand van Python 3, verder gerefereerd als Python. Alle standaardmodules in Python kunnen dan ook gebruikt worden, hiernaast is het mogelijk zelf modules te importen, in het geval van dit onderzoek een CIF-Parser. Aan de hand van de Python logica is het eenvoudig loops te creëren die automatisch een aantal objecten aanmaakt in Blender. Hieronder wordt kort de Blender API beschreven.

De Blender API biedt een aantal modules aan die gebruikt kunnen worden, naast alle reeds bestaande functionaliteiten van Python. De voornaamste van deze modules is de bpy module, die alle functies bevat in verband met het aanmaken en aanpassen van objecten in Blender. De bpy

bpy.ops	Bevat alle operators voor het maken en aanpassen van objecten
bpy.context	Geeft de mogelijkheid specifieke data op te vragen
bpy.data	Bevat alle data van de objecten
bpy.app	Hulpmodule bij het schrijven van add-ons en extra functionaliteiten
bpy.types,bpy.utils,bpy.props	Hulpmodules bij het schrijven van add-ons
bpy.path	Vrijwel identiek aan de os.path submodule van Python

Tabel 2.2 De submodules van de Blender bpy module (Conlan, 2017)

module is zodanig groot dat deze nogmaals wordt opgedeeld in submodules, deze worden samen met hun taak weergegeven in tabel[2.2]. In de documentatie van de Blender API worden alle methodes van deze modules uitgelegd. In hoofdstuk vier wordt dieper ingegaan op het gebruik van de Blender API en de toepassing ervan in dit onderzoek.

2.6 Conclusie

Kristallografie is een specifieke tak in de wetenschap die zich bezighoudt met het onderzoeken van kristallen, hun structuur en hun eigenschappen. Een kristal wordt opgebouwd uit eenheidscellen. Het beschrijven van zo'n eenheidscel geeft alle informatie van het kristal. De vorm van een eenheidscel wordt bepaald door zes roosterparameters en een van de 14 Bravaisroosters. Deeltjes kunnen zich op 240 manieren in de eenheidscel bevinden, de ruimtegroepen genaamd.

In het CIF formaat is alle informatie over een kristalstructuur terug te vinden, het is leesbaar door zowel computers als mensen. CIF bestanden worden geschreven in ASCII wat eenvoudige leesbaarheid aanpasbaarheid toelaten. Gebaseerd op het STAR formaat heeft het CIF formaat enkele syntactische restricties waaraan moeten gehouden worden. De structuur van CIF bevat datablokken, data-elementen en lussen om de data weer te geven.

De bestaande 3D kristalvisualisatiesoftwarepakketten zijn vaak moeilijk in omvang, duur of biedt niet genoeg vrijheid. VESTA en Olex² zijn gratis programma's die kristalstructuren kunnen visualiseren en geven de gebruiker een aantal handige tools om onderzoek te vereenvoudigen. Ze hebben echter hun nadelen wat leidt tot de vraag naar andere programma's.

Parsers hebben als functie data te extraheren uit data met een bepaalde syntax. Ze worden gebruikt om data te visualiseren of bruikbaar te maken in andere toepassingen. De complexiteit van het schrijven van een parser is omgekeerd evenredig met het aantal syntaxregels van een formaat, hierdoor is het schrijven van een CIF parser eerder moeilijk. Cctbx en PyCIFRW zijn open source modules die in staat zijn CIF bestanden te parsen.

Blender is een gratis, open source project waarin driedimensionale tekeningen kunnen worden gemaakt. Met een groot aantal tools, waaronder animeren, sculpten en een eigen game-engine, geeft Blender de gebruiker veel vrijheid en mogelijke toepassingen. Door gebruik te maken van texturen en materialen kunnen er fotorealistische tekeningen gemaakt worden. Scripten in Blender

laat het automatiseren van tekenen toe en wordt gedaan aan de hand van Python scripts en de modules uit de Blender API.

Hoofdstuk 3

Ontwerpproces

In dit hoofdstuk worden de ondernomen stappen besproken bij het ontwerpen van een interface die kristalstructuren, in de vorm van een CIF-bestand, kan visualiseren in Blender. Een meer gedetailleerde beschrijving van dit proces kan worden gevonden in hoofdstuk vier. In de eerste sectie zal worden uitgelegd hoe, met behulp van een parser, een programma in Python kan geschreven worden dat de gegevens uit een CIF-bestand kan omzetten in verwerkbare data en deze opslaat in geschikte datastructuren.

3.1 Van cif naar py

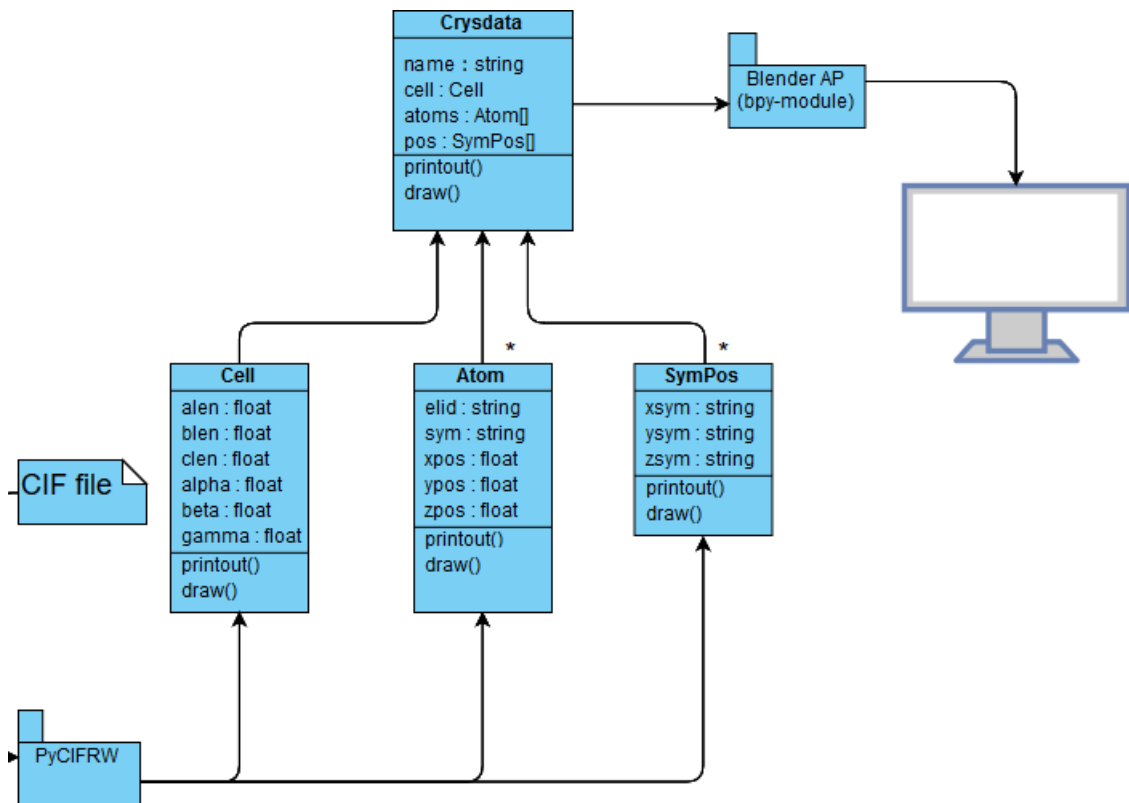
3.1.1 Zelf een parser ontwerpen

In de vierde sectie van hoofdstuk twee werd er reeds aangehaald wat er allemaal komt kijken bij het ontwerpen van een parser. Het is gelukt een programma[Bijlage C] te ontwerpen dat alle CIF-bestanden kan inlezen die te vinden zijn op de kristallografische databank van het IZA. Dit is omdat deze allemaal dezelfde textstructuur hebben. Bestanden uit databanken die werken met een andere structuur kunnen gemisinterpreteerd worden waardoor de parser nutteloos is. Het is mogelijk het programma aan te passen zodat, ongeacht de structuur van het CIF-bestand, een juiste interpretatie van de data kan gedaan worden. Dit zou echter veel tijd vergen en zal, in belang van dit onderzoek, niet worden gedaan. In de plaats zal er gewerkt worden met een reeds functionerende CIF-parser.

3.1.2 Inlezen van CIF

Het inlezen van een CIF-bestand zal worden gedaan worden met behulp van de PyCIFRW parser, die besproken wordt in het tweede hoofdstuk van deze tekst, en Python3 als programmeertaal. De volledige broncode van het programma[Bijlage D] en een voorbeeld van een CIF-bestand[Bijlage B] kan worden gevonden in de bijlages.

Vooraleer er data kan worden ingelezen dienen er datastructuren gemaakt te worden waarin deze



Figuur 3.1: Klassendiagramma van het ontwerp van deze thesis

kan worden opgeslagen. Zo een datastructuur noemt men een class en het werken hiermee wordt object georiënteerd programmeren genoemd. Enkele voordelen dat dit biedt over de gewoonlijke, imperatieve, methode van programmeren, zijn overzichtelijkheid, herbruikbaarheid van classes, een extra laag van veiligheid doordat de data niet rechtstreeks kan worden aangepast. Figuur[3.1] wordt een klassendiagram genoemd, en geeft de algemene opbouw het programma aan de hand van de gebruikte datastructuren, methodes, modules en hun onderlinge relaties. Uit dit diagram kan er geïnterpreteerd worden hoe de data uit het CIF-bestand met behulp van de PyCIFRW-parser wordt omgezet in de verschillende datastructuren welke nadien kunnen worden gevisualiseerd in Blender.

De broncode van het programma dat een CIF-bestand inleest en in datastructuren omzet kan gevonden worden tussen de bijlagen van deze tekst.[Bijlage D]

Bibliografie

- Bernstein, H. J., Bollinger, J. C., Brown, I. D., Gražulis, S., Hester, J. R., McMahon, B., Spadaccini, N., Westbrook, J. D., and Westrip, S. P. (2016). Specification of the crystallographic information file format, version 2.0. *Journal of Applied Crystallography*, 49(1):277–284.
- Blenderguru (2012). Photorealism competition results. <https://www.blenderguru.com/competitions/photorealism-competition-results>.
- Borchardt-Ott, W. (2011). *Crystallography: an introduction*. Springer Science and Business Media.
- Britannica (2018). Auguste bravais: French physicist. <https://www.britannica.com/biography/Auguste-Bravais>.
- Chronister, J. (2011). *Blender Basics: Classroom Tutorial Book*. Blender Nation.
- Conlan, C. (2017). *The Blender Python API: Precision 3D Modeling and Add-on Development*. Apress.
- Hall, S. R., Allen, F. H., and Brown, I. D. (1991). The crystallographic information file (cif): a new standard archive file for crystallography. *Acta Crystallographica Section A: Foundations of Crystallography*, 47(6):655–685.
- IUCR (2009). Programming with pycifrw and pystarw. https://www.iucr.org/__data/iucr/cif/software/pycifrw/PyCifRW-3.2/documentation.pdf.
- IZA (2018). Database of zeolite structures. http://europe.iza-structure.org/IZA-SC/ftc_table.php.
- Momma, K. and Izumi, F. (2008). Vesta: a three-dimensional visualization system for electronic and structural analysis. *Journal of Applied Crystallography*, 41(3):653–658.
- Theo Hahn, H. W. and Muller, U. (2005). *International Tables for Crystallography Volume A: Space-Group Symmetry*. by SPRINGER for THE INTERNATIONAL UNION OF CRYSTALLOGRAPHY.

Bijlagen (op USB)

- Bijlage A: De CIF-dictionary
- Bijlage B: Cif-bestand van de kristalstructuur van Chaziet
- Bijlage C: Ontworpen programma dat CIF-bestanden kan inlezen
- Bijlage D: Programma dat CIF-bestand kan inlezen en in een datastructuur kan omzetten

FACULTEIT INDUSTRIËLE INGENIEURSWETENSCHAPPEN
CAMPUS DE NAYER SINT-KATELIJNE-WAVER
J. De Nayerlaan 5
2860 SINT-KATELIJNE-WAVER, België
tel. + 32 15 31 69 44
iiw.denayer@kuleuven.be
www.iw.kuleuven.be



Grantt Map: Actuele Planning

	Sep	Okt	Nov	Dec	Jan	Feb	Maa	Apr	Mei	Jun
Literatuurstudie										
Kristalografie										
Blender										
Libraries										
Uitvoering										
Programmastructuur										
Schrijven scripts										
Debugging										
Optimaliseren										
Extra features (Add-ons)										
Vorbereiding Thesis										
Schriftelijk										
Mondeling										

CHECKLIST bij masterproef

 (bij tussentijds verslag)

Volgende checklist is een hulpmiddel om na te gaan of alle elementen in het tussentijds verslag opgenomen zijn. De ingevulde en ondertekende lijst wordt bij de promotor afgegeven samen met de twee kopijen van het tussentijds verslag.

- ✓ titel, naam van de student en promotor
- ✓ inhoudsopgave
- ✓ lijst met afkortingen (alfabetisch) en symbolen
- ✓ situering
- ✓ probleemstelling
- ✓ doelstelling
- ✓ onderzoeksvraag (en eventuele deelvragen)
- ✓ literatuurstudie van de betrokken state of the art
- ✓ reeds gerealiseerde elementen
- ✓ geactualiseerde planning (Gantt kaart)
- ✓ bibliografie (met correcte verwijzingen in de tekst)

De verantwoordelijkheid voor het correct invullen van deze checklist ligt bij de student.

Naam
Jarrit Boons

datum
20/12/2018

handtekening

