

Elektronenkristallographie unter dynamischen Bedingungen und
konvergenter Bestrahlung im Transmissionselektronenmikroskop:
Anwendung auf Ta_2P , $\text{Sc}_3\text{Sc}_{0.32}\text{Al}_{0.68}\text{O}_6$ und ScGaO_3

Dissertation

zur

Erlangung des Doktorgrades (Dr. rer. nat.)

der

Mathematisch-Naturwissenschaftlichen Fakultät

der

Rheinischen Friedrich-Wilhelms-Universität Bonn

vorgelegt von

Ulrich Wilke

aus

Clausthal-Zellerfeld

Bonn, Juli 2002

1	Einleitung	3
2	Allgemeiner Teil	9
2.1	Streuung an kristallinen Substanzen	9
2.1.1	Streuprozess	9
2.1.2	Grundformel der Elektronenbeugung	10
2.1.3	Braggsche Gleichung	11
2.1.4	Das reziproke Gitter und die Ewaldkugel	12
2.1.5	Lauzentrum	14
2.1.6	Kikuchilinien	15
2.1.7	Kinematische Streutheorie	16
2.1.7.1	Gitter- und Strukturamplitude	17
2.1.7.2	Der Atomformfaktor	19
2.1.7.3	Der Debye-Waller Koeffizient	20
2.1.8	Dynamische Theorie der Elektronenbeugung	21
2.1.8.1	Der Zweistrahlfall	23
2.1.8.2	Die Multislice Methode	26
2.1.9	Beugung unter konvergenten Einstrahlbedingungen	28
2.1.10	Das Programmpaket ELSTRU	29
2.1.11	Die Miller-Bravais Notation	31
2.2	Das Transmissionselektronenmikroskop	32
2.2.1	Die CCD-Kamera	35
2.2.1.1	Korrektur der Zählrate der CCD-Kamera	35
2.2.1.2	Abbildungseigenschaft der CCD-Kamera	36
2.2.2	Die energiedispersive Röntgenanalyse	37
2.2.2.1	Grundlagen der EDX	38
2.2.2.2	Aufbau und Funktionsweise der EDX	39
2.3	Probenpräparation	39
2.3.1	Vollpräparation	40
3	Spezieller Teil	43
3.1	Erweiterung des Programmpaketes ELSTRU	43
3.1.1	Ergebnisse mit einem unter konvergenten Einstrahlbedingungen simulierten Beugungsbild	45
3.1.2	Die Untergrundkorrektur	47
3.1.3	Ergebnisse mit einem unter konvergenten Einstrahlbedingungen erstellten experimentellen Beugungsbild	48
3.2	Ergebnisse mit der neuen Methode	51
3.2.1	Strukturverfeinerung von Ta ₂ P	51
3.2.1.1	Die zur Strukturverfeinerung benutzen Zonenachsen	52
3.2.1.1.1	Zonenachse [010] von Ta ₂ P	52
3.2.1.1.2	Zonenachsen [110], [310] und [230] von Ta ₂ P	55
3.2.1.1.3	Zonenachse [112] von Ta ₂ P	58
3.2.2	Strukturverfeinerung von Ta ₂ P	61
3.2.2.1	Diskussion der Strukturverfeinerung an Ta ₂ P	62

3.3	Strukturverfeinerung von $\text{Sc}_3\text{Sc}_{0.32}\text{Al}_{0.68}\text{O}_6$	71
3.3.1	Die zur Strukturverfeinerung verwendeten Zonenachsen von $\text{Sc}_3\text{Sc}_{0.32}\text{Al}_{0.68}\text{O}_6$	71
3.3.1.1	Zonenachse $[0001]$ von $\text{Sc}_3\text{Sc}_{0.32}\text{Al}_{0.68}\text{O}_6$	72
3.3.1.2	Zonenachse $[2\bar{1}\bar{1}0]$ von $\text{Sc}_3\text{Sc}_{0.32}\text{Al}_{0.68}\text{O}_6$	75
3.3.1.3	Zonenachse $[10\bar{1}1]$ von $\text{Sc}_3\text{Sc}_{0.32}\text{Al}_{0.68}\text{O}_6$	78
3.3.1.4	Zonenachse $[10\bar{1}0]$ von $\text{Sc}_3\text{Sc}_{0.32}\text{Al}_{0.68}\text{O}_6$	81
3.3.1.5	Zonenachsen $[01\bar{1}2]$ und $[20\bar{2}\bar{1}]$ von $\text{Sc}_3\text{Sc}_{0.32}\text{Al}_{0.68}\text{O}_6$	83
3.3.2	Strukturverfeinerung von $\text{Sc}_3\text{Sc}_{0.32}\text{Al}_{0.68}\text{O}_6$ mit den Datensätzen aus der Vorverfeinerung	87
3.3.2.1	Sc(1)-Koordination	89
3.3.2.2	Al(1)/Sc(2)-Koordination	91
3.3.2.3	Al(2)/Sc(3)-Koordination	92
3.3.2.4	Elementarzelle von $\text{Sc}_3\text{Sc}_{0.32}\text{Al}_{0.68}\text{O}_6$	93
3.4	Strukturverfeinerung von ScGaO_3	94
3.4.1	Die zur Strukturverfeinerung verwendeten Zonenachsen von ScGaO_3	95
3.4.1.1	Zonenachse $[100]$ von ScGaO_3	95
3.4.1.2	Zonenachse $[010]$ von ScGaO_3	98
3.4.1.3	Zonenachse $[001]$ von ScGaO_3	101
3.4.1.4	Zonenachse $[101]$ von ScGaO_3	104
3.4.1.5	Zonenachse $[120]$ von ScGaO_3	106
3.4.2	Strukturverfeinerung von ScGaO_3	108
3.4.2.1	Sc(1)-Koordination	111
3.4.2.2	Ga(1)/Ga(2)-Koordination	112
3.4.2.3	Sc(2)/Ga(3)-Koordination	113
3.4.2.4	Elementarzelle von ScGaO_3	114
3.5	Vergleich der gefundenen Bindungsabstände mit den Oxiden $\beta\text{-Ga}_2\text{O}_3$, Sc_2O_3 und α- bzw. $\gamma\text{-Al}_2\text{O}_3$	116
3.6	Vergleich der neu entwickelten mit der bisherigen Methode zur Aufnahme von Elektronenbeugungsdaten	118
3.7	Zusammenfassung	120
3.8	Ausblick	122
4	Literaturverzeichnis	123
5	Anhang	127
5.1	Mathematischer Zusammenhang zwischen dem Atomformfaktor der Elektronenbeugung mit dem der Röntgenbeugung	127
5.2	Programm <i>konv2.pro</i>	128

1 Einleitung

Die Elektronenkristallographie ist ein wertvolles Hilfsmittel für die Aufklärung von Kristallstrukturen. Darunter versteht man die Strukturanalyse von kristallinen Proben mit Hilfe von Elektronenbeugung und elektronenmikroskopischen Abbildungen. Der Vorschlag, unbekannte Strukturen mit Hilfe von Beugungsbildern in Kombination mit Abbildungen zu lösen, wurde in der 65-jährigen Geschichte der Elektronenkristallographie immer wieder kontrovers geführt, da es viele noch nicht gelöste Probleme mit dieser Arbeitsweise gibt. Sie ist aber dennoch häufig die Methode der Wahl für Verbindungen, die von ihrer Natur aus nur mikrokristalline Produkte hervorbringen können. Beispielsweise können röntgenamorphe Kristalle im Elektronenstrahl durchaus Beugungsbilder erzeugen. Andererseits liegt darin aber auch das Grundproblem der Elektronenkristallographie. Die starke und damit nach der dynamischen Theorie zu berechnende Wechselwirkung der Elektronen mit der Materie lässt eine Strukturlösung mit direkten Methoden aus Elektronenbeugungsdaten im Gegensatz zur Röntgenkristallographie (Berechnung der Beugungsintensitäten nach der kinematischen Theorie) eigentlich nicht zu. Unter bestimmten Bedingungen ist aber die Verwendung von Elektronenbeugungsbildern in Kombination mit Abbildungen zur direkten Strukturaufklärung möglich.

Ein Großteil der zurückliegenden Untersuchungen fokussierte die Bemühungen darauf, quasi-kinematische Daten mit Elektronenbeugungsbildern zu erzeugen. Dabei können auf mehrere Vorgehensweisen zurückgegriffen werden. Zum einen wird durch die Erhöhung der Beschleunigungsspannung der Elektronen im Transmissionselektronenmikroskop die Größe der Wechselwirkung der Elektronen mit der Materie verkleinert [1]. Auch Beugungsbilder von organischen Verbindungen liefern quasi-kinematische Daten. Diese enthalten mit Kohlenstoff, Wasserstoff und Sauerstoff Atome, die im Vergleich zu den schwereren Elementen kleine Atomformfaktoren besitzen. Durch die damit weniger wahrscheinliche Mehrfachstreuung verkleinern sich die auftretenden dynamischen Effekte. Weiterhin sollten Beugungsaufnahmen von möglichst dünnen Probenbereichen zur Strukturlösung verwendet werden, um damit ebenfalls dynamische Effekte zu unterdrücken. Selbst für Verbindungen mit schwereren Atomen ist es möglich, quasi-kinematische Daten zu erhalten. *Vincent et al.* [2-4] zeigten dieses anhand von Beugungsdaten aus höheren Laue Zonen, die unter konvergenten Einstrahlbedingungen aufgezeichnet wurden.

Dynamische Effekte in Elektronenbeugungsbildern sind aber nicht immer ein Nachteil bei der Strukturaufklärung von unbekanntem Verbindungen. Zu den Hauptgründen für eine nicht erfolgreiche Strukturaufklärung zählt die Bestimmung einer falschen

Raumgruppe. Die Auswirkungen der dynamischen Wechselwirkungen von vielen Strahlen auf die Auslöschung bestimmter Reflexe kann zur eindeutigen Bestimmung der Raumgruppe benutzt werden [5]. Gerade auch bei Beugungsbildern, die unter konvergenten Bedingungen aufgezeichnet werden, lässt die dynamische Wechselwirkung die unzweifelhafte Bestimmung der Raumgruppe zu [6].

Liegen quasi-kinematische Elektronenbeugungsdaten vor, stehen im Vergleich zur Röntgenkristallographie analoge Methoden zur Strukturlösung bereits zur Verfügung. Dabei muss das sogenannte Phasenproblem gelöst werden. Die Intensität eines gebeugten Röntgen- oder Elektronenstrahls ist dem Quadrat des Strukturformfaktors F_{hkl} proportional. Man erhält aus Beugungsaufnahmen daher nur den Betrag $|F_{hkl}|$ des Strukturformfaktors, was gleichbedeutend mit dem Amplitudenquadrat der gestreuten Welle ist. Die Phase der gestreuten Welle bleibt zunächst unbestimmt. Mit der Bestimmung der Phasen der einzelnen Beugungsreflexe (hkl) werden jedoch wichtige Informationen über die Atompositionen gewonnen.

Vainshtein [7] hat in seinen Arbeiten als erster versucht, leichtere Atome während einer Verfeinerung mit Elektronenbeugungsdaten aufzufinden. Er benutzte durch Röntgenkristallographie aufgefundene Atompositionen von schweren Atomen als Startwerte für die Phasenfindung von Beugungsamplituden. Es folgten zahlreiche Beispiele für diese sogenannte *Trial and Error*-Methode, die die beste Übereinstimmung eines Modells mit den experimentellen Elektronenbeugungsdaten zu finden versucht. *Voigt-Martin* et al. [8, 9] konnten beispielsweise die Struktur von kristallisierten flexiblen aromatischen Molekülen mit dieser Methode aufklären. Sie verifizierten die molekulare Packung der organischen Moleküle auch durch direkte Methoden zur Strukturlösung (s.u.). Die Konformation der Moleküle innerhalb der Elementarzelle wurde durch Energieminimierung der Molekülkonformation bestimmt.

Die in der Röntgenkristallographie oft angewendete Patterson-Methode zur Strukturlösung kann in der Elektronenkristallographie nicht so erfolgreich angewendet werden, da die Größenvariation der Atomformfaktoren für Elektronenstreuung im Vergleich zu den Röntgen-Atomformfaktoren geringer ausfällt [10]. Trotzdem wurde schon früh versucht, eine Pattersonprojektion von Aminosäuresalzen einschließlich der Lokalisierung der Metallome zu interpretieren [11]. Eine aktuellere Arbeit hat sich mit der Strukturlösung von Legierungen beschäftigt. In dieser Arbeit von *Vincent* und *Exelby* [12, 13] konnten mit Reflexen der höheren Lauezonen die Atomlagen in verschiedenen Schichten innerhalb der Elementarzelle lokalisiert werden.

Mit den Verbesserungen der magnetischen Linsen in den letzten Jahren ist man dem Ziel, separierte Atompositionen innerhalb eines hochaufgelösten Bildes sichtbar zu machen, näher gekommen. Auch die Objektwellenrekonstruktion hat wesentliches dazu beigetragen [14]. Hochaufgelöste TEM-Bilder müssen aber nicht zwingend mit

atomarer Auflösung aufgenommen werden, um als Startwerte der gefundenen Phasen für eine anschließende Verfeinerung zu dienen. *Weirich* et al. [15] haben die Struktur von $\text{Ti}_{11}\text{Se}_4$ auf diese Weise gelöst und verfeinert. Dabei wurde über eine Fourier-Transformation von HRTEM-Bildern zur primären Phasenfindung, die Phasen der Elektronenbeugungsreflexe (hkl) bestimmt. Durch eine Kombination von Abbildung und Beugung ist es bei der anorganischen Verbindung $\text{K}_2\text{O}_{0.7}\text{Nb}_2\text{O}_5$ ebenfalls gelungen, Startwerte für die Phasen von Reflexen zu erhalten und damit anschließend die Struktur aufzuklären [16].

Für die Strukturlösung mit direkten Methoden ist die Verwendung von quasi-kinematischen Elektronenbeugungsdaten unabdingbar. Untersuchungen mit simulierten Daten von repräsentativen organischen Molekülen zeigten, dass die experimentellen Bedingungen (Probenpräparation, Beschleunigungsspannung) so eingestellt werden können, dass experimentelle quasi-kinematische Daten erhalten werden können [17, 18].

In den letzten zehn Jahren haben sich drei Gruppen für die Strukturlösung mit direkten Methoden hervorgetan. *Dorset* et al. [20] benutzen schon 1975 die *Symbolic Addition*-Methode [19] zur Strukturlösung von organischen Verbindungen. Die wichtige Erkenntnis in weiteren Veröffentlichungen dieser Arbeitsgruppe war, dass texturierte Beugungsdaten von organischen Verbindungen benutzt werden können, um Kristallstrukturen ohne Vorkenntnisse aus der Röntgenkristallographie zu lösen [21]. Weitere Techniken, wie die Verwendung der *Tangens Formel* [22] oder die *Sayre-Gleichung* [23] zur Strukturlösung, wurden vielfach und erfolgreich von *Dorset* et al. angewendet. *Gilmore* et al. [24, 25] haben ihre Bemühungen auf die Anwendung der *Maximum Entropie*-Methode und *Maximum-Likelihood*-Prozeduren zur Strukturlösung mit Elektronenbeugungsdaten gerichtet. Das Problem, gute Phasenwerte für alle Reflexe zu erhalten, kann mit der *Maximum-Entropie*-Methode in manchen Fällen einfacher gelöst werden, da die für die direkten Methoden notwendigen rechnerischen Vereinfachungen bei der *Maximum-Entropie*-Methode nicht gemacht werden müssen. Zudem wird bei der Ermittlung einer neuen Phase die Ergebnisse aller bis dahin errechneten Phasen berücksichtigt [26]. *Fan* und *Li* fokussierten ihre Arbeiten auf die Bestimmung der Phasen von hochaufgelösten Hellfeldbildern [27] und die Erweiterung der *Sayre-Gleichung* auf die Elektronenbeugung [28, 29]. Diese Gruppe hat zudem eindrucksvolle Untersuchungen von der inkommensurablen Überstruktur des Minerals Ankanigite (= $\text{Ba}_{0.8}(\text{Ti}, \text{V}, \text{Cr})_8\text{O}_{16}$) durchgeführt [30]. Die von *Mo* et al. [31] gefundene Kristallstruktur eines Hochtemperatursupraleiters ist ebenfalls ein Beispiel für die Strukturlösung von anorganischen Verbindungen mit Hilfe der Elektronenkristallographie.

Die Kristallstrukturen von anorganischen Verbindungen, die aus leichten Atomen bestehen, können analog zu den organischen Verbindungen mit direkten Methoden aus

Elektronenbeugungsdaten aufgeklärt werden. *Cowley* [32] konnte auf diese Weise schon 1953 die Struktur der Borsäure aufklären. Eine neue Methode zur Datenaufnahme bei niedrigen Proben Temperaturen wurde von *Dorset* [33] zur erneuten Strukturaufklärung von Borsäure benutzt. Die Wasserstoffatome konnten aber auch mit dieser Arbeitsweise nicht mit letzter Sicherheit bestimmt werden. Schließlich können auch kompliziertere Kristallstrukturen wie die von $\text{Ba}_2\text{Fe}_2\text{O}_5$ mit Hilfe der Elektronenkristallographie aufgeklärt werden. Es handelt sich dabei um eine Perowskitstruktur, bei der sich 14 Formeleinheiten in der Elementarzelle befinden und damit im Bezug auf die ABO_3 Stöchiometrie eines leerstellenfreien Perowskits 14 Sauerstoffatome fehlen [34].

Den erforderlichen Abschluss einer Strukturlösung bildet die Verfeinerung der gefundenen Atompositionen, der Debye-Waller-Faktoren und gegebenenfalls der Besetzungsgrade der verschiedenen Positionen innerhalb der Elementarzelle. Fourier-Verfeinerungen werden besonders dann erfolgreich angewendet, wenn nicht alle Atome bei der Strukturlösung gefunden worden sind [35]. Sie benötigen allerdings bestimmte Vorgaben wie z.B. die Bewahrung von bestimmten Molekülgeometrien innerhalb einer kristallinen Phase oder das nicht Unterschreiten von bestimmten Kation-Anion-Abständen. Ansonsten ist es durchaus möglich, dass ein minimaler R-Wert von einer nicht realistischen Struktur gefunden wird.

Least-square-Verfeinerungen konnten bis 1998 ebenfalls nicht ohne bestimmte Vorgaben erfolgreich abgeschlossen werden. *Campbell et al.* [36] setzten verbundene Atom-Atom-Abstände und die entsprechenden Bindungswinkel innerhalb von Molekülen auf ideale Werte und suchten sowohl das Minimum der nicht verbundenen Atom-Atom-Potenziale als auch das Minimum des R-Wertes. Erfolgreich angewendet wurde das Programm LALS (= Linked-Atom Least Square) bei der Strukturverfeinerung von Polysacchariden und Polynukleotiden. Eine vorgabenfreie Verfeinerung mit Elektronenbeugungsdaten ist mit dem Programmpaket ELSTRU von *Jansen et al.* möglich [37]. MSLS als eigentliches Strukturverfeinerungsprogramm innerhalb des Programmpaketes ELSTRU ist der erste Algorithmus, der die Multi-Slice-Methode (MSLS) mit dem Least-Square-Verfahren (MSLS) verknüpft. Damit ist das der erste erfolgreiche Versuch mit dynamischen experimentellen Elektronenbeugungsdaten eine Strukturverfeinerung durchzuführen, da die Multi-Slice-Methode Beugungsintensitäten nach der dynamischen Theorie berechnet. Auf diese Weise ist es prinzipiell möglich, alle Atome des Periodensystems im Rahmen einer Strukturverfeinerung zu verfeinern, da die auftretenden dynamischen Effekte bei schwereren Atomen berücksichtigt werden. Die Restriktionen bezüglich der verwendbaren Kristalldicken entfallen bis zu einer gewissen Probendicke ebenfalls. Beugungsaufnahmen von Kristallbereichen, die je nach dem Streupotenzial der Atome in der Verbindung eine Dicke von bis zu 400 Å besitzen können, können zur Strukturverfeinerung herangezogen werden. Beugungsbilder von dickeren

Probenbereichen sollten allerdings von der Verfeinerung ausgeschlossen werden, da die Anzahl an lokalen Minima der R-Wert-Funktion mit wachsender Kristalldicke zunimmt, so dass das Auffinden des richtigen globalen Minimums erschwert oder sogar unmöglich ist. MSLS benötigt zur Strukturverfeinerung Punktbeugungsbilder, die durch einen feinen (\varnothing ca. 10 nm) parallelen Elektronenstrahl erzeugt werden. Das hat zum einen den Nachteil, dass der Primärstrahlreflex (000) unter den Aufnahmebedingungen übersteuert ist und damit nicht zur Strukturverfeinerung herangezogen werden kann. Des Weiteren ist der erreichbare minimale Durchmesser des Elektronenstrahls bei paralleler Beleuchtung mit ca. 10 nm häufig noch zu groß, da zur erfolgreichen Strukturverfeinerung eine homogene Kristalldicke beleuchtet werden muss (s. Kapitel 2.1.10).

Im Rahmen dieser Arbeit wurde eine neue Methode zur Datenauswertung von unter konvergenten Einstrahlbedingungen erstellten Beugungsbildern entwickelt. Der minimale Elektronenstrahldurchmesser lässt sich dadurch auf ca. 5 nm reduzieren, wodurch in der Regel die Bestrahlung einer homogen dicken Probenstelle ermöglicht wird. Dadurch kann die Anzahl an verwertbaren Aufnahmen im Vergleich zu der konventionellen Methode erhöht werden. Das Primärstrahlscheibchen (000) ist in Beugungsbildern, die mit einem konvergenten Elektronenstrahl auf der Probe erzeugt werden, nicht mehr übersteuert. Auf diese Weise kann der zusätzliche Informationsgehalt des Primärstrahls für die Strukturverfeinerung nutzbar gemacht werden. Das speziell dafür entwickelte Programm mit der Bezeichnung *konv2.pro* (s. Anhang), das unter IDL® [38] geschrieben wurde, ermöglicht das Auswerten von unter konvergenten Bedingungen erstellten Beugungsbildern. Diese neue Methode wurde an bekannten Systemen (Korund und Ta₂P) getestet und an zwei neuen Verbindungen (Sc₃Sc_{0.32}Al_{0.68}O₆ und ScGaO₃) mit unbekannter Kristallstruktur angewendet.

Zuerst wurde ein simuliertes [39] und ein experimentelles Beugungsbild von Korund in [0001], die jeweils unter konvergenten Einstrahlbedingungen errechnet bzw. erstellt sind, mit *konv2.pro* ausgewertet. Sowohl bei dem simulierten als auch bei dem experimentellen Beugungsbild wurde die *x*-Koordinate des Sauerstoffs innerhalb der Korundstruktur verfeinert.

Im Falle des Ta₂P wurden mehrere Kristallorientierungen zur Strukturverfeinerung herangezogen. Dabei wurde die Auswahl der einzelnen Zonenachsen auf solche beschränkt, bei denen sich die einzelnen Beugungsscheibchen innerhalb eines unter konvergenten Einstrahlbedingungen erstellten Beugungsbildes nicht überlappen. Als erstes werden die Ergebnisse der Vorverfeinerungen (s. Kap. 3.1) von den einzelnen Beugungsbildern gezeigt. Anschließend wird das Resultat der Strukturverfeinerung von Ta₂P beschrieben, das danach diskutiert wird.

Für die Verfeinerungen der beiden Strukturvorschläge von $\text{Sc}_3\text{Sc}_{0.32}\text{Al}_{0.68}\text{O}_6$ und ScGaO_3 wurden alle relevanten Kristallorientierungen zur jeweiligen Strukturverfeinerung verwendet. Im Falle des $\text{Sc}_3\text{Sc}_{0.32}\text{Al}_{0.68}\text{O}_6$ konnten alle Beugungsbilder unter konvergenten Bedingungen aufgezeichnet werden. Aufgrund der mit 19.004 Å langen a -Achse von ScGaO_3 wurden bei der Strukturverfeinerung dieser Verbindung auch punktförmige Beugungsbilder verwendet. Auch hier werden jeweils zunächst die Ergebnisse der Vorverfeinerung der einzelnen Beugungsbilder gezeigt. Das jeweilige Resultat der Strukturverfeinerung von $\text{Sc}_3\text{Sc}_{0.32}\text{Al}_{0.68}\text{O}_6$ und ScGaO_3 wird abschließend diskutiert und bewertet.

2 Allgemeiner Teil

2.1 Streuung an kristallinen Substanzen

In der Festkörperchemie ist es neben der Untersuchung der physikalischen und chemischen Eigenschaften von großer Bedeutung, die Struktur der zu untersuchenden Verbindung zu ermitteln. Etablierte Methoden wie die Röntgendiffraktometrie sind in ihren unterschiedlichen Varianten vielfach erfolgreich angewendet worden. Insbesondere mit der Vierkreis-Einkristallstrukturanalyse kann man seit längerem routinemäßig Kristallstrukturen lösen.

2.1.1 Streuprozess

Trifft Röntgenstrahlung auf ein Elektron, so können die Elektronen selbst zu Schwingungen angeregt werden und damit gleichzeitig als Quelle einer Strahlung mit gleicher Wellenlänge angesehen werden. Elektronen werden im Gegensatz zu den Röntgenstrahlen hauptsächlich an den Atomkernen gestreut. Diese kohärente Streuung ist Voraussetzung für die Beugung von Röntgen- bzw. Elektronenstrahlen an kristallinen Substanzen. Werden Elektronen bzw. Atomkerne regelmäßig angeordneter Atome von einer Röntgen- bzw. Elektronenwelle getroffen, so entsteht eine Vielzahl von kohärenten Strahlungsquellen. Da sich die kugelförmig ausbreitenden Wellenfronten der einzelnen Atome überlagern, kommt es zu Interferenzerscheinungen. Die physikalische Wechselwirkung der Probe mit der Welle ist von der Art der verwendeten Strahlung abhängig. Neutronen werden fast ausschließlich direkt am Atomkern gestreut. Röntgenwellen treten hingegen im wesentlichen mit der Elektronenhülle der Atome in Wechselwirkung. Ein maßgeblicher Unterschied besteht in der Größe der Wechselwirkung der unterschiedlichen Strahlung mit der Probe. Elektronen treten sehr viel stärker mit der Materie in Wechselwirkung als es Röntgenwellen tun. Das hat zur Folge, dass die kinematische Theorie, welche die Röntgenbeugung physikalisch korrekt beschreibt, für die Elektronenbeugung unzureichend ist, da sie Mehrfachstreuung nicht berücksichtigt. Für sie wurde die dynamische Theorie der Elektronenbeugung entwickelt. Durch die größere Wechselwirkung der Elektronen mit der Probe und die bessere Fokussierbarkeit von Elektronen im Vergleich zu Röntgen- oder Neutronenstrahlung ist es möglich, einhundert mal kleinere Kristalle als bei der Einkristallröntgenstrukturanalyse zu untersuchen.

2.1.2 Grundformel der Elektronenbeugung

Jedem Teilchen kann nach der Beziehung von de Broglie eine bestimmte Wellenlänge λ zugeordnet werden. Die Geschwindigkeit der Elektronen läßt sich aus der kinetischen Energie $E_{\text{kin}} = e \cdot U$ berechnen. Damit erhält man die Wellenlänge der Elektronen in der klassischen Näherung:

$$\lambda = \frac{h}{\sqrt{2 \cdot m_0 \cdot e \cdot U}}. \quad (1)$$

mit h = Plancksches Wirkungsquantum,
 m_0 = Ruhemasse des Elektrons,
 e = Elementarladung
und U = Beschleunigungsspannung.

Allerdings erreichen die Elektronen schon bei einer Beschleunigungsspannung von 100 kV knapp 55 % der Lichtgeschwindigkeit, weshalb relativistische Effekte berücksichtigt werden müssen. Tatsächlich ist die Masse des Elektrons bei einer Energie von 100 keV um den Faktor 1.2 größer als die Ruhemasse m_0 des Elektrons. Der Faktor liegt bei einer Beschleunigungsspannung von 300 kV bei 1.59, wodurch ein Korrekturfaktor benötigt wird, der die relativistischen Effekte berücksichtigt:

$$\lambda = \frac{h}{\sqrt{2 \cdot m_0 \cdot e \cdot U \cdot \left(1 + \frac{e \cdot U}{2 \cdot m_0 \cdot c^2}\right)}}, \quad (2)$$

mit der Lichtgeschwindigkeit c .

Für eine Beschleunigungsspannung von 300 kV ergibt sich damit eine Wellenlänge der Elektronen von 1.97 pm [63].

Setzt man diese Wellenlänge in die Braggsche Gleichung ein, so ergeben sich im Vergleich zur Röntgenbeugung sehr viel kleinere Beugungswinkel θ bis ca. 1.5° .

Wie in Abbildung 1 gezeigt ist, kann man in die Braggsche Gleichung für kleine Winkel $\theta \approx \sin \theta \approx \tan \theta$ den Wert

$$\frac{R}{2 \cdot L}$$

einsetzen.

Damit ergibt sich

$$\lambda \cdot L = d_{hkl} \cdot R = \text{const.}, \quad (\text{Kamerakonstante}) \quad (3)$$

mit

R = Abstand des Beugungspunktes von dem Primärstrahl,

L = Abstand der Probe von der Photoplatte.

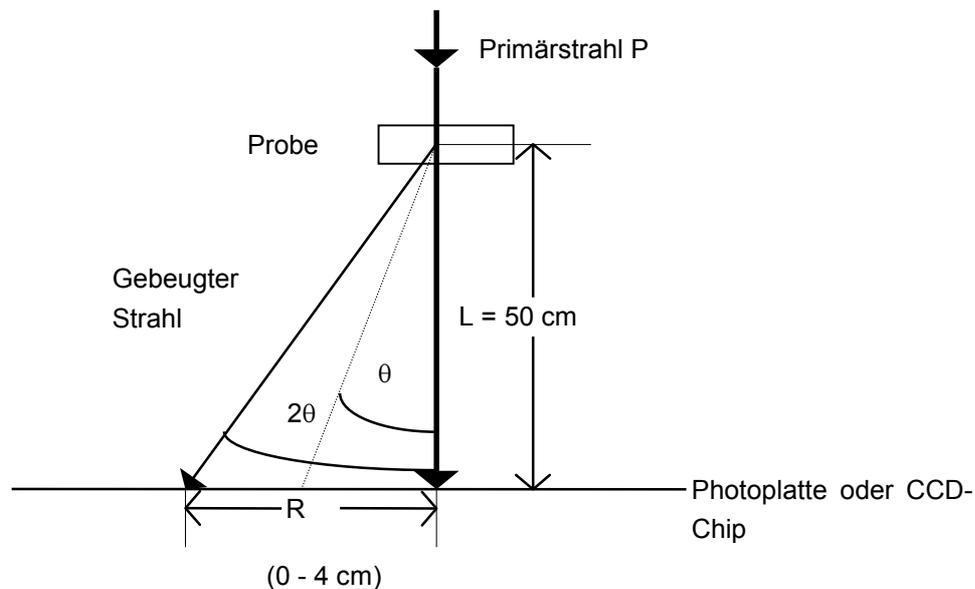


Abbildung 1: Schematische Darstellung der Elektronenbeugung zur Herleitung der Grundformel für die Kamerakonstante.

Die Kamerakonstante (3) läßt sich mit Hilfe von Standards bestimmen, deren d_{hkl} -Werte bekannt sind [60].

2.1.3 Braggsche Gleichung

William Lawrence Bragg hat sich einen kristallinen Festkörper aus Ebenen aufgebaut gedacht, die mit einer unterschiedlichen Anzahl an Atomen besetzt sein können, aber immer den gleichen Abstand, den sogenannten Netzebenenabstand (d_{hkl}), voneinander haben. Trifft ein Röntgen- oder Elektronenstrahl mit einem bestimmten Winkel θ auf eine Netzebene, so kann die Beugung als Reflexion aufgefasst werden.

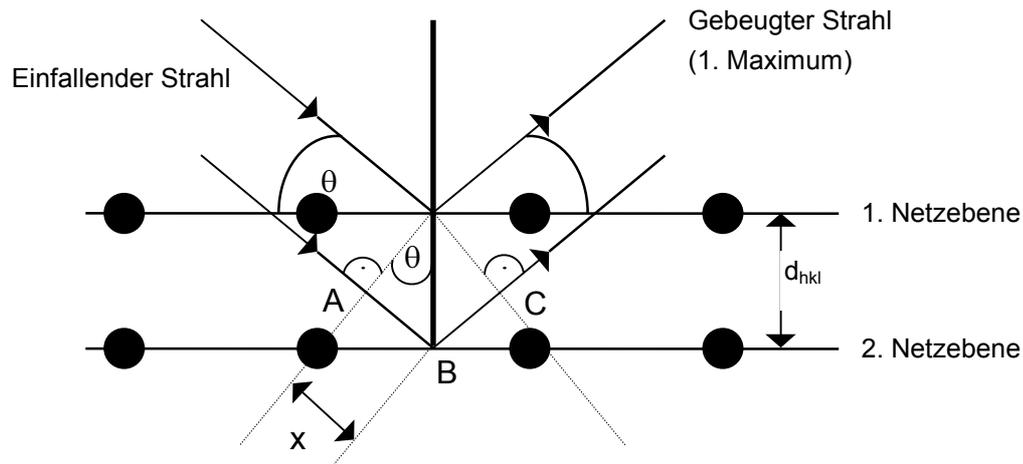


Abbildung 2: Herleitung zur Braggschen Gleichung.

Unter ganz bestimmtem Einfallswinkel θ treten positive und unter anderem Winkel negative Interferenzen der reflektierten Welle auf. Die Braggsche Gleichung für Intensitätsmaxima kann man wie folgt aus dem Gangunterschied paralleler Strahlen herleiten:

$$\overline{AB} + \overline{BC} = n \cdot \lambda.$$

Mit $\overline{AB} = \overline{BC}$ folgt $2 \cdot \overline{AB} = n \cdot \lambda,$

mit $\sin \theta = \frac{\overline{AB}}{d_{hkl}}$ folgt die Braggsche Gleichung

$$n \cdot \lambda = 2 \cdot d_{hkl} \cdot \sin \theta, \quad (4)$$

mit n = Beugungsordnung,

λ = Wellenlänge (z.B. der Röntgenstrahlung),

d_{hkl} = Netzebenenabstand

und θ = Einfallswinkel (z.B. der Röntgenstrahlung).

Durch Ausmessen der Beugungswinkel lassen sich die Netzebenenabstände und schließlich die Gitterparameter berechnen. Systematische Auslöschungen von Reflexen geben den Bravais-Typ an.

2.1.4 Das reziproke Gitter und die Ewaldkugel

Die Beschreibung der Beugung an einem Kristallgitter kann wesentlich erleichtert werden, indem man anstelle von zweidimensionalen Ebenenscharen im Kristall mit deren eindimensionalen Normalen arbeitet. Während die Richtung der Normalen die Lage der entsprechenden Ebene im Kristall charakterisiert, wird durch die Länge des Normalenvektors der reziproke Netzebenenabstand dargestellt. An den Endpunkten der Normalen erhält man das reziproke Gitter, welches analog zum realen Gitter von drei

Basisvektoren $\vec{a}^*, \vec{b}^*, \vec{c}^*$ und den drei eingeschlossenen Winkeln $\alpha^*, \beta^*, \gamma^*$ aufgespannt ist. Zwischen dem realen und dem reziproken Gitter besteht folgender mathematischer Zusammenhang:

$$\vec{a}^* = \frac{\vec{b} \times \vec{c}}{V}, \quad \vec{b}^* = \frac{\vec{c} \times \vec{a}}{V}, \quad \vec{c}^* = \frac{\vec{a} \times \vec{b}}{V}, \quad (5)$$

mit $V = \vec{a} \cdot (\vec{b} \times \vec{c}) = \vec{b} \cdot (\vec{c} \times \vec{a}) = \vec{c} \cdot (\vec{a} \times \vec{b}) =$ Volumen der realen Elementarzelle.

Die Ewaldsche Konstruktion veranschaulicht den Zusammenhang zwischen dem reziproken Gitter und den zugehörigen Reflexen. Ausgehend von dem Wellenvektor \vec{k}_0 mit der Länge $1/\lambda$, der den einfallenden Strahl repräsentiert und der im Ursprung des reziproken Gitters endet, erhält man den Punkt A. Um diesen Punkt wird eine Kugel mit dem Radius $1/\lambda$ konstruiert. Schneidet die Kugel weitere reziproke Gitterpunkte, so erscheint ein Reflex in einem so orientierten Kristall, d.h. die Braggsche Beugungsbedingung ist erfüllt. Der Beugungswinkel 2θ lässt sich ebenfalls aus der Ewaldschen Konstruktion ablesen.

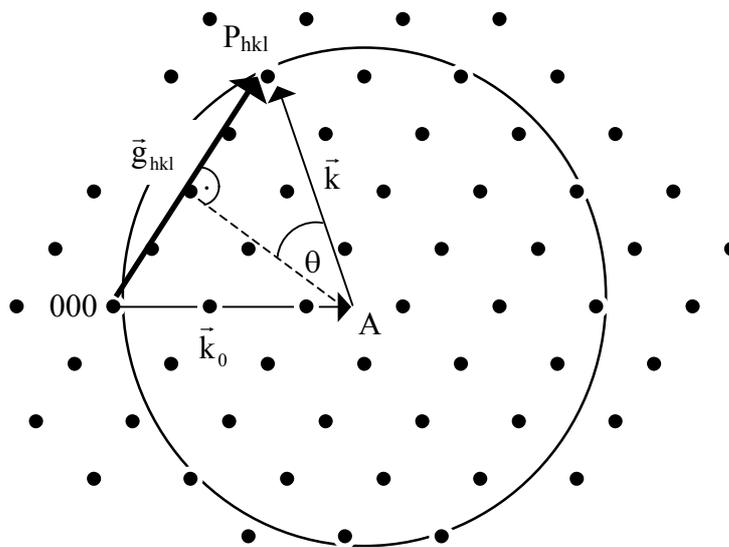


Abbildung 3: Die Ewaldsche Konstruktion im Fourierraum.

Für θ ergibt sich nach Abbildung 3:

$$\sin \theta = \frac{\text{Gegenkathete}}{\text{Hypotenuse}} = \frac{0,5 \cdot |\vec{OP}|}{|\vec{k}|} = \frac{|\vec{g}_{hkl}|}{2|\vec{k}|} = \frac{1/d_{hkl}}{2/\lambda} = \frac{\lambda}{2d_{hkl}}, \quad (6)$$

was der Braggschen Gleichung (4) entspricht.

Die Bedeutung der Ewaldschen Konstruktion liegt in der darin zum Ausdruck kommenden einfachen geometrischen Beziehung zwischen Primär- und Interferenzstrahl einerseits und dem reziproken Gitter andererseits.

Im Vergleich zur Röntgenbeugung ist der Radius der Ewaldkugel bei der Elektronenbeugung sehr viel größer. Für einen mit 300 kV beschleunigten Elektronenstrahl beträgt der Radius 51 \AA^{-1} , wohingegen er für Cu- K_{α} -Strahlung 0.65 \AA^{-1} beträgt. Dadurch sind in der Elektronenbeugung immer mehrere Reflexe einer Lauezone zu sehen, weil die Ewaldkugel immer mehrere reziproke Gitterpunkte schneidet. Bei der Elektronenbeugung werden Beugungsreflexe auch dann beobachtet, wenn die Ewaldkugel den reziproken Gitterpunkt nicht exakt schneidet. Dieser Sachverhalt wird im Kapitel Kinematische Streutheorie durch den Anregungsfehler s näher beschrieben [61].

2.1.5 Lauezentrum

Ist der Kristall perfekt in die Zonenachse einorientiert, so steht der einfallende Wellenvektor senkrecht auf einer bestimmten Ebene des reziproken Gitters (s. Abbildung 4a), und das Beugungsbild ist radialsymmetrisch um den Primärstrahl ausgeleuchtet.

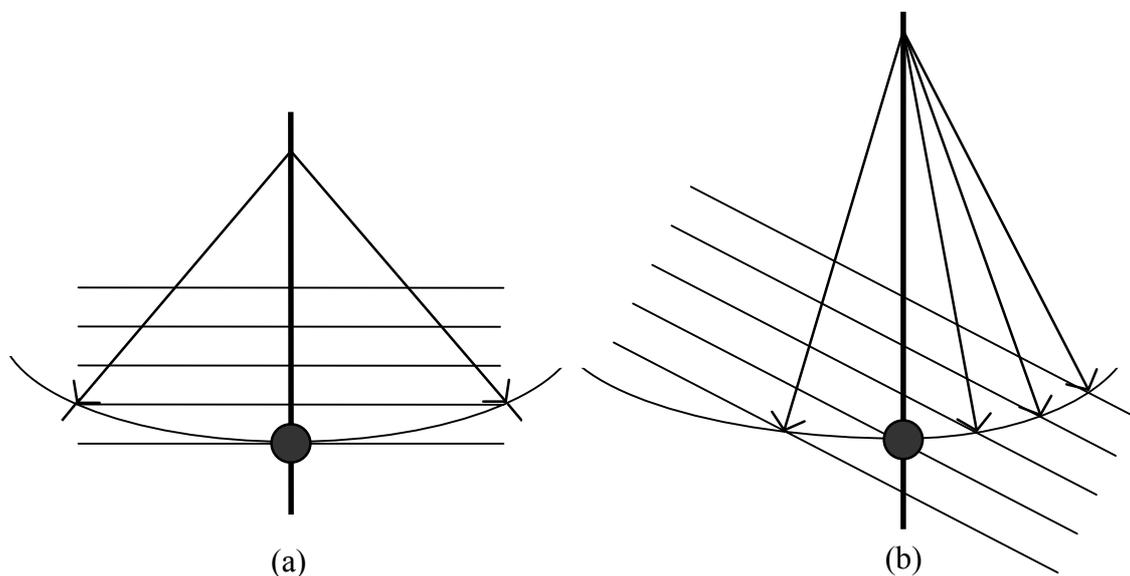


Abbildung 4: Ewaldsche Konstruktion im symmetrischen Lauefall (a) und nach einer Kristallverkipfung (b).

Ist die Probe nicht ideal in der Zonenachse, so steht der Wellenvektor der einfallenden Welle nicht mehr senkrecht zur Ebene des reziproken Gitters, sondern ist um einen Kippwinkel ihr gegenüber verändert. Das Beugungsbild ist nicht mehr symmetrisch um den Primärstrahl ausgeleuchtet. Es treten z. B. in Abbildung 4b rechts vom Primärstrahl

mehr Reflexe als im symmetrischen Lauefall auf. Zudem sind die Beugungsintensitäten von Fresnel Reflexen (z.B. $(h00)$ und $(\bar{h}00)$ in einer zentrosymmetrischen Raumgruppe) stark unterschiedlich.

2.1.6 Kikuchilinen

Die nach ihrem Entdecker benannten Kikuchilinen treten in den Probenbereichen auf, bei denen eine Mehrfachstreuung der Elektronen möglich ist. Sie werden zunächst inelastisch und anschließend elastisch gestreut. Dazu muß die Probe eine gewisse Mindestdicke aufweisen, da die Wahrscheinlichkeit für einen inelastischen Streuprozess ca. 10-mal kleiner ist als für einen elastischen Streuprozess. Beugungsreflexe lassen sich in diesen Bildern oft kaum mehr erkennen.

Die Maximalintensität inelastisch gestreuter Elektronen ist in Primärstrahlrichtung am größten (s. Abbildung 5). Sie nimmt mit zunehmendem Streuwinkel ab. Der Energieverlust der inelastisch gestreuten Elektronen ist dabei im Vergleich zur Primärenergie vernachlässigbar klein. Folglich bleibt die Wellenlänge der Elektronen nahezu konstant. Bei der sich anschließenden elastischen Streuung wirken die inelastisch gebeugten Elektronen als neue Primärstrahlen, die nun aus ihren jeweiligen Richtungen auf die Netzebenen treffen.

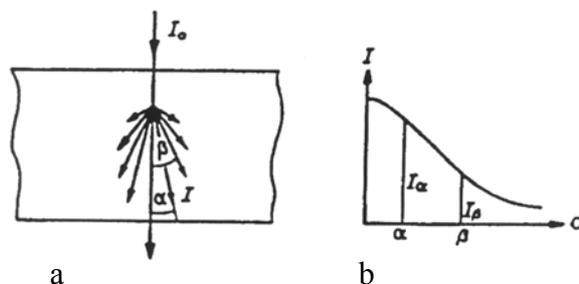


Abbildung 5: Inelastische Streuung

a) im Querschnitt der Probe b) in Abhängigkeit vom Streuwinkel.

Kikuchilinen werden von einer hellen und einer dunklen Linie begrenzt (s. Abbildung 6). Die inelastisch gestreuten Elektronen bilden einen Strahlkegel in einem Winkelbereich von $90^\circ - \theta$. Theoretisch schneiden die Streukegel den Beobachtungsschirm als schwach gekrümmte Hyperbeln, die aber aufgrund der kleinen Streuwinkel θ durch Geraden gut angenähert werden können und auch als solche beobachtet werden.

Durch Überlagerung mit der kontinuierlichen Streuintensität der inelastisch gestreuten Elektronen ergeben sich weiter vom Primärstrahl entfernt eine helle Linie bei H und eine dunkle Linie bei D (s. Abbildung 6). Die höhere Intensität der helleren Linie beruht auf dem geringeren Streuwinkel α bei der ersten Streuung. Damit einhergehend hat die dunklere Linie aufgrund ihres größeren Streuwinkels β bei der ersten Streuung eine geringere Intensität [60].

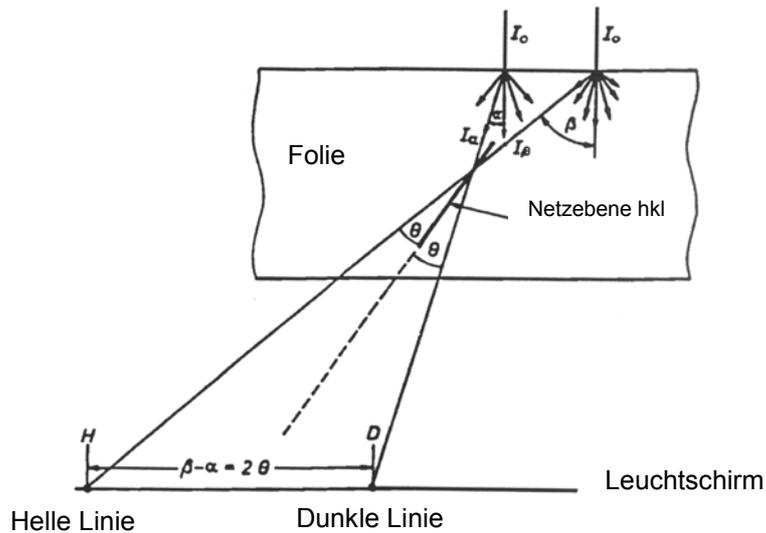


Abbildung 6: Entstehung von Kikuchilinien.

2.1.7 Kinematische Streutheorie

Die kinematische Theorie beschreibt allgemein die Intensität einer einfach gebeugten Welle. Dabei geht sie von zwei Voraussetzungen aus. Zum ersten sind alle Streuprozesse elastisch, und zum zweiten ist die Intensität der abgebeugten Strahlen im Vergleich zu der des Primärstrahls so klein, dass Sekundärbeugung der bereits einmal gebeugten Wellen nicht mehr stattfindet. Diese Vereinfachungen gelten im allgemeinen für die Röntgen- und die Neutronenbeugung. Für die Elektronenbeugung muss für quantitative Aussagen die dynamische Theorie herangezogen werden, da die Wechselwirkung der Elektronenwellen mit der Materie 10^7 -mal größer ist als die von Röntgenstrahlung. Aufgrund der sehr viel größeren Wechselwirkung tritt bei Elektronenstrahlung Sekundärbeugung auf. Auch die Intensität der abgebeugten Strahlen ist größer, als man es nach der kinematischen Theorie erwarten würde. Für qualitative Aussagen ist die kinematische Theorie aber auch bei der Elektronenbeugung ein bewährtes Mittel.

2.1.7.1 Gitter- und Strukturamplitude

Die Strukturamplitude F_{hkl} spiegelt den atomaren Aufbau der Elementarzelle wider. Sie trägt auch am meisten zur Intensität der Reflexe in Bragg-Beugung bei, da diese hauptsächlich von der Art der Atome und von deren Lage in der Elementarzelle abhängt. Summiert man über alle Atome in der Elementarzelle im Kristall, so erhält man die Strukturamplitude (7).

$$F_{hkl} = \sum_{k=1}^n f_R \cdot \exp[-2\pi i(h \cdot x_k + k \cdot y_k + l \cdot z_k)], \quad (7)$$

mit f_R = Atomformfaktor des k-ten Atoms,
 h, k, l = Millersche Indizes des betrachteten Reflexes
 und x_k, y_k, z_k = Atomkoordinaten in der Elementarzelle.

Wenn die Strukturamplitude den Wert Null annimmt, ist ein Beugungsreflex ausgelöscht.

Die Streukraft der einzelnen Atome, die sich in f_R widerspiegelt, wird mit zunehmender Ordnungszahl Z der Atome größer und mit größeren Beugungswinkeln θ kleiner.

Da der Atomformfaktor bei der Elektronenbeugung ca. um den Faktor 100 größer ist als bei der Röntgenbeugung, besitzen die gebeugten Strahlen bei der Elektronenbeugung relativ zum Primärstrahl eine größere Intensität als bei der Röntgenbeugung. Dadurch können die bereits gebeugten Strahlen, wie oben erwähnt, zum zweiten Mal gebeugt werden. Damit kann es bei ausreichender Kristalldicke zum Auftreten von kinematisch verbotenen Reflexen kommen, da diese Reflexe „dynamisch“ angeregt sind.

Neben der Strukturamplitude trägt auch die Gitteramplitude G einen entscheidenden Anteil an der Intensität der gebeugten Strahlung bei. In G wird die äußere Form des Kristalls berücksichtigt, und für einen quaderförmigen Kristall gilt:

$$G = \sum_{m=1}^{M_1} \sum_{n=2}^{M_2} \sum_{o=3}^{M_3} \exp[-2\pi i \cdot (\vec{k} - \vec{k}_0) \cdot \vec{r}_g], \quad (8)$$

mit $M_{1,2,3}$ = Anzahl der Elementarzellen in Richtung der drei Gittervektoren,
 \vec{k} = Vektor des gebeugten Strahls,
 \vec{k}_0 = Vektor des einfallenden Strahls,
 \vec{r}_g = Translationsvektor zum Ursprung der Elementarzelle.

Bei sehr dünnen Proben können auch solche Netzebenen Beugungspunkte erzeugen, die die Braggsche Gleichung nicht exakt erfüllen. Dieser Sachverhalt kann mit der Einführung des Anregungsfehlers \vec{s} erklärt werden, der ein reziproker Gittervektor in

Richtung des Primärstrahls ist. Schneidet die Ewald-Kugel den Vektor \vec{s} , so tritt ein Beugungsreflex auf (s. Abbildung 7).

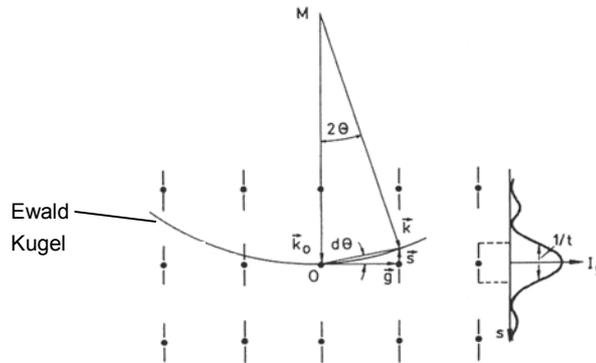


Abbildung 7: Zur Definition des Anregungsfehler \vec{s} .

Für die Reflexionsbedingung gilt nun die Vektorgleichung:

$$\vec{g} + \vec{s} = \vec{k} - \vec{k}_0. \quad (9)$$

Unter Berücksichtigung von \vec{s} ergibt sich für die Gitteramplitude:

$$|G|^2 = \frac{\sin^2(\pi \cdot M_1 \cdot \vec{s}_x \cdot \vec{a}_1)}{(\pi \cdot \vec{s}_x \cdot \vec{a}_1)^2} \cdot \frac{\sin^2(\pi \cdot M_2 \cdot \vec{s}_y \cdot \vec{a}_2)}{(\pi \cdot \vec{s}_y \cdot \vec{a}_2)^2} \cdot \frac{\sin^2(\pi \cdot M_3 \cdot \vec{s}_z \cdot \vec{a}_3)}{(\pi \cdot \vec{s}_z \cdot \vec{a}_3)^2} \quad (10)$$

$$= G_x \cdot G_y \cdot G_z$$

Der Beitrag der Gitteramplitude für eine Raumrichtung verschwindet damit erst, wenn

$$s_i M_i a_i = 1 \text{ oder } s_i = \frac{1}{M_i a_i} \left(= \frac{1}{t}; \text{ mit } t = \text{Probendicke} \right) \text{ ist.}$$

Der Anregungsfehler ist somit umgekehrt proportional zur Probendicke entlang der beobachteten Richtung. Für die in der TEM notwendigen dünnen Proben ergibt sich daraus die Elongation der reziproken Gitterpunkte (s. Abbildung 7). Neben dem Hauptmaximum bei $\vec{s}_z = 0$ treten bei $|G|^2$ Nebenmaxima auf (s. Abbildung 8).

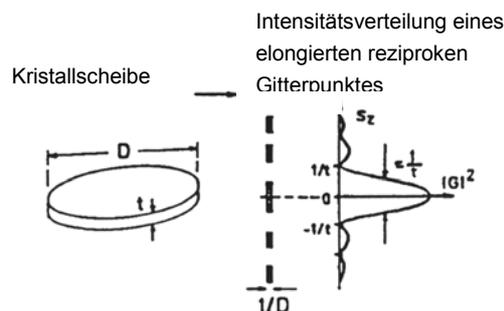


Abbildung 8: Intensitätsverlauf als Funktion des Anregungsfehlers in Richtung von z.

2.1.7.2 Der Atomformfaktor

Das Streuvermögen von Atomen für Elektronen- und Röntgenstrahlung nimmt mit ihrer Ordnungszahl zu. Atome sind aber keine mathematischen Punkte, sondern besitzen eine räumliche Ausdehnung. Es treten Phasendifferenzen zwischen den gestreuten Strahlen eines Atoms auf, da die räumliche Ausdehnung der Atome im Bereich der Wellenlänge der Elektronen liegen. Deshalb kommt es zu einer Schwächung der Gesamtintensität der Streustrahlung. Die Elektronenstreuung ist winkelabhängig und nimmt mit zunehmendem Streuwinkel ab. Dieser Zusammenhang gilt sowohl für die Röntgen- als auch für die Elektronenbeugung. Für die Röntgenbeugung ist der Atomformfaktor folgendermaßen definiert:

$$f_R(s) = \int \rho(r) \cdot \exp[-4 \cdot \pi \cdot i \cdot s \cdot r] dr, \quad (11)$$

mit ρ = Elektronendichte im Atom

$$\text{und } s = \frac{\sin \theta}{\lambda} = \frac{1}{2d_{hkl}}.$$

In Analogie zur Röntgenbeugung ist der Elektronenstreufaktor $f^e(s)$ definiert als

$$f^e(s) = \int V(r) \cdot \exp[-4 \cdot \pi \cdot i \cdot s \cdot r] d\tau. \quad (12)$$

Der mathematische Zusammenhang zwischen dem Atomformfaktor für die Röntgenbeugung mit dem für die Elektronenbeugung ist in Gleichung 13 wiedergegeben (Herleitung s. Anhang nach [62]):

$$f^e(s) = \frac{|e|}{16\pi^2 \epsilon_0 s^2} \int [Z - f^x(s)] \quad (13)$$

mit Z = Ordnungszahl des Atoms.

Für einen Kristall mit mehreren Atomen pro Elementarzelle ergeben sich die Potenzialkoeffizienten als Summe über alle atomaren Streufaktoren einer Elementarzelle:

$$V_g = \frac{1}{\Omega} \sum_i f^e(s) \exp(-2 \cdot \pi \cdot i \cdot g \cdot r_i) \quad (14)$$

mit Ω = Volumen der Elementarzelle.

Die Mott-Bethe-Gleichung verknüpft die Kristall-Strukturfaktoren V_g mit den atomaren Röntgen-Streufaktoren f_i^x :

$$V_g = \frac{e}{16 \cdot \pi^3 \epsilon_0 \Omega} \sum_i \frac{Z_i - f_i^x(s)}{s^2} \exp(-B_i s^2) \exp(-2 \cdot \pi \cdot g \cdot r_i) \quad (15)$$

$$= \frac{h^2}{8 \cdot \pi \epsilon_0 m_e |e| \Omega} F_g^B$$

- mit r_i = Atomposition in der Elementarzelle,
 m_e = Ruhemasse des Elektrons,
 F_g^B = Elektronenstrukturfaktor, der mit Hilfe der Born Näherung erhalten wurde (International Tables vor 1990)
 und B_i = Debye-Waller-Koeffizient, der im folgenden Kapitel näher betrachtet wird [62].

2.1.7.3 Der Debye-Waller Koeffizient

Der Atomformfaktor beschreibt das Streuvermögen von ruhenden freien Atomen oder Ionen. Tatsächlich führen die Atome eines Festkörpers aber Schwingungen um ihre Ruhelage aus. In Abbildung 9 ist die Energie eines Atoms als Funktion der interatomaren Abstände gezeigt.

Nur am absoluten Nullpunkt wird ein Energieminimum erreicht, das dem Abstand R_0 entspricht. Wird den Atomen Energie z. B. in Form von Wärme zugeführt, so existieren ein innerer und ein äußerer Umkehrpunkt der Schwingungsbewegung des Atoms. Durch diese hochfrequente Bewegung entstehen zusätzliche Phasendifferenzen zwischen den gestreuten Wellen, die eine asymmetrische Ausdehnung des Intensitätsbereiches eines Reflexes bewirken.

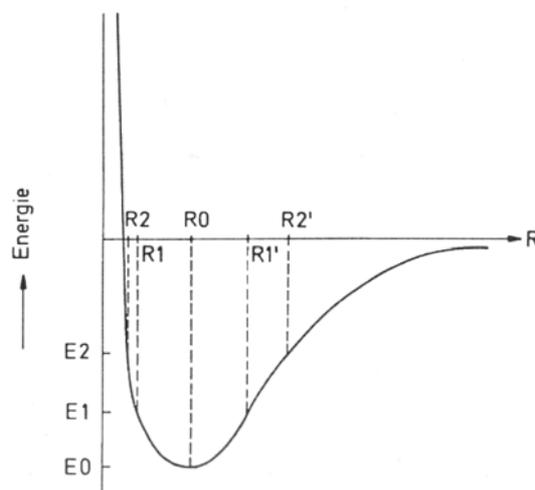


Abbildung 9: Energie der Atome in Abhängigkeit von ihrem Abstand.

Dieser Sachverhalt lässt sich nicht einfach beschreiben, da er sowohl eine Gittereigenschaft als auch eine Atomeigenschaft ist. Man ist daher auf Näherungsverfahren angewiesen. Debye und Waller haben einen Korrekturfaktor entwickelt, der trotz einiger Näherungen gute Ergebnisse liefert.

Der Faktor B stellt ein zusätzliches Dämpfungsglied für die atomaren Streufaktoren dar. Er bewirkt, dass die Streukurven mit steigendem Beugungswinkel rascher abklingen und die Intensität damit geringer wird. B lässt sich durch folgende Beziehung mit der mittleren Auslenkung \bar{u} senkrecht zur reflektierenden Netzebene in Verbindung setzen:

$$B = 8 \cdot \pi^2 \cdot \bar{u}^2. \quad (16)$$

Die mittlere quadratische Auslenkung \bar{u}^2 der Atome erhält man aus dem Debye-Modell:

$$\bar{u}^2 = \frac{3 \cdot h^2}{4 \cdot \pi^2 \cdot M \cdot k \cdot \Theta_D} \cdot \left[\frac{1}{4} + \frac{T^2}{\Theta_D} \cdot \int_0^{\Theta_D/T} \frac{x \cdot dx}{\exp(x) - 1} \right], \quad (17)$$

mit M = Atommasse,
 Θ_D = Debyetemperatur,
 T = Temperatur
 und h = Plancksches Wirkungsquantum [61].

2.1.8 Dynamische Theorie der Elektronenbeugung

Die kinematische Theorie ist nur für die Röntgen- und Neutronenstrahlung physikalisch korrekt und quantitativ anwendbar (s. a. Kap. Kinematische Streutheorie). Der Elektronenstrahl erfährt aufgrund Braggscher Reflexion eine ständige Abschwächung beim Durchlauf durch den Kristall. Dies hat zur Folge, dass tiefer liegende Netzebenen einen kleineren Beitrag zur Reflexintensität leisten. Phänomene wie Umweganregung können die Intensität des Primärstrahls wieder verstärken.

Um eine korrekte quantitative Beschreibung zu erhalten, ist eine quantenmechanische Behandlung des Beugungsproblems durchzuführen, die schließlich zu einer dynamischen Theorie führt.

Da es sich um stationäre Elektronenzustände im Kristall handelt, lassen sich die Elektronen im gitterperiodischen Potenzial durch die zeitunabhängige Schrödingergleichung beschreiben:

$$\left[-\frac{\hbar^2}{2 \cdot m} \cdot \nabla + V(\vec{r}) \right] \cdot \Psi(\vec{r}) = E \cdot \Psi(\vec{r}), \quad (18)$$

mit $\hbar = \frac{h}{2\pi}$; h = Plancksches Wirkungsquantum,
 m = Elektronenmasse,
 ∇ = Laplacescher Differentialoperator,
 $\Psi(\vec{r})$ = Wellenfunktionen der Elektronen,

wobei die potenzielle Energie der Elektronen im Kristallfeld durch

$$V(\vec{r}) = U(\vec{r}) \cdot \frac{h}{2 \cdot m} \quad (19)$$

mit $U(\vec{r})$ = Gitterpotenzial

und die Gesamtenergie oder Energieeigenwerte der Elektronen durch

$$E = \frac{h^2 \cdot k_0^2}{2 \cdot m} \text{ gegeben ist.} \quad (20)$$

Nach Einsetzen von Gleichungen (19) und (20) in die Schrödingergleichung (18) ergibt sich:

$$\Delta \Psi(\vec{r}) + 4 \cdot \pi^2 \cdot [k_0^2 - U(\vec{r})] \cdot \Psi(\vec{r}) = 0. \quad (21)$$

Da das Gitterpotenzial $U(\vec{r})$ periodisch ist, kann es in einer Fourierreihe entwickelt werden:

$$U(\vec{r}) = \sum_n U_n \cdot \exp[2 \cdot \pi \cdot i \cdot (\vec{g}_n \cdot \vec{r})], \quad (22)$$

mit n = Index der betrachteten Netzebene,
 U_n = Fourierkomponente, die der Streukraft der Netzebene entspricht,
 \vec{g}_n = der zur Netzebene zugehörige reziproke Gittervektor
 und \vec{r} = Vektor im Ortsraum.

Mit dem Blochwellenansatz $b(\vec{r})$ läßt sich die Eigenfunktion eines Elektrons in einem periodischen Potenzial darstellen:

$$b(\vec{r}) = \sum_n C_n(\vec{K}_0) \cdot \exp[2 \cdot \pi \cdot i(\vec{K}_0 + \vec{g}) \cdot \vec{r}], \quad (23)$$

mit \vec{K}_0 = Wellenvektor des ungebeugten Elektronenstrahls im Kristall

und $\vec{K}_0 + \vec{g}_n$ = Wellenvektor des gebeugten Elektronenstrahls im Kristall.

Eingesetzt in die Schrödingergleichung (21) ergibt sich

$$-\exp(2 \cdot \pi \cdot i \cdot \vec{K}_0 \cdot \vec{r}) \cdot \left[\sum_n \left(C_n \cdot (\vec{K}_0^2 - \vec{k}_0^2) + \sum_{n'} U_{n-n'} \cdot C_{n'} \right) \cdot \exp(2 \cdot \pi \cdot i \cdot \vec{g}_n \cdot \vec{r}) \right] = 0. \quad (24)$$

Eine nicht triviale Lösung ergibt sich nur dann, wenn für alle n die Summe in der inneren runden Klammer gleich Null wird:

$$C_n \cdot (\vec{K}_n^2 - k_0^2) + \sum_{n'} (U_{n-n'} \cdot C_{n'}) = 0. \quad (25)$$

Dies stellt ein System mit n Gleichungen dar, das als Säkulargleichung bezeichnet wird [63]. Der sogenannte n-Strahl-Fall berücksichtigt den Primärstrahl und alle gebeugten Strahlen. Er gibt die Verhältnisse quantitativ korrekt wieder, ist aber aufgrund der Vielzahl an Gleichungen (in manchen Fällen treten mehr als 100 Reflexe auf) sehr umfangreich. Zum qualitativen Verständnis der Beugungsphänomene ist es jedoch oft ausreichend, neben dem Primärstrahl nur einen gebeugten Strahl zu betrachten.

2.1.8.1 Der Zweistrahlfall

Der Zweistrahlfall ist die einfachste Näherung, mit dem man die Säkulargleichung (25) lösen kann. Hier sind nur zwei Blochwellen angeregt, so dass sich die gesamte Intensität auf den Primärstrahl und einen gebeugten Strahl verteilt. In der Gleichung (25) werden damit nur die Komponenten n bzw. $n' = 0$ und 1 berücksichtigt. Damit reduziert sich die Säkulargleichung auf zwei Gleichungen der Form:

$$(\vec{K}_0^2 - \vec{k}_0'^2) \cdot C_0 + U_{-1} \cdot C_{-1} = 0, \quad (26)$$

$$U_1 \cdot C_0 + (\vec{K}_1^2 - \vec{k}_0'^2) \cdot C_1 = 0, \quad (27)$$

mit $\vec{k}_0'^2 = \vec{k}_0^2 - U_0$.

Die beiden Gleichungen (26) und (27) lassen sich unter der Voraussetzung lösen, dass sich die Wellenvektoren \vec{K}_n an der Eintrittsfläche des Kristalls nur um den Betrag δk in z-Richtung ändern:

$$\delta k^{1,2} = \frac{1}{2} \cdot s \pm \frac{1}{2} \cdot \sqrt{\sigma^2 + s^2}, \quad (28)$$

mit $s =$ Anregungsfehler,

$$\sigma = \frac{1}{\xi_g}, \quad \xi_g = \text{Extinktionslänge des Reflexes } \vec{g}.$$

Mit $\sqrt{\sigma^2 + s^2} = p$ ergeben sich folgende Werte für die C_n :

$$C_0^1 = \sqrt{\frac{p-s}{2 \cdot p}}, \quad C_1^1 = \sqrt{\frac{p+s}{2 \cdot p}}, \quad (29)$$

$$C_0^2 = -\sqrt{\frac{p+s}{2 \cdot p}}, \quad C_1^2 = -\sqrt{\frac{p-s}{2 \cdot p}}. \quad (30)$$

Damit erhält man als Lösung der Schrödingergleichung die beiden Blochwellen

$$b^1(\vec{r}) = \exp\left[2 \cdot \pi \cdot i \cdot (\vec{k}_0' \cdot \vec{r})\right] \cdot \exp(2 \cdot \pi \cdot i \cdot \delta k^1 \cdot z) \cdot \left[C_0^1 + C_1^1 \cdot \exp(2 \cdot \pi \cdot i \cdot \vec{g} \cdot x)\right] \quad \text{und} \quad (31)$$

$$b^2(\vec{r}) = \exp\left[2 \cdot \pi \cdot i \cdot (\vec{k}_0' \cdot \vec{r})\right] \cdot \exp(2 \cdot \pi \cdot i \cdot \delta k^2 \cdot z) \cdot \left[C_0^2 + C_1^2 \cdot \exp(2 \cdot \pi \cdot i \cdot \vec{g} \cdot x)\right]. \quad (32)$$

Man kann die Entwicklung der Elektronenwellen im Zweistrahlfall folgendermaßen zusammenfassen:

Beim Eintritt einer Elektronenwelle in den Kristall werden zwei Blochwellen angeregt. Jede Blochwelle setzt sich ihrerseits aus zwei Teilwellen zusammen, wobei der Wellenvektor der einen Teilwelle in Richtung des Primärstrahls und der andere Wellenvektor in Richtung des gebeugten Strahls zeigt. An der Probenunterseite, wo der Elektronenstrahl wieder austritt, zerfällt jede Blochwelle in jeweils eine ebene Welle parallel zum Primärstrahl und parallel zum gebeugten Strahl.

Innerhalb der Probe treten die Blochwellen unterschiedlich mit den Atomen in Wechselwirkung, da sich die eine Blochwelle vorwiegend am Ort der Atome und die andere Blochwelle sich überwiegend zwischen den Atomen aufhält. Dadurch kommt es zu unterschiedlichem Absorptionsverhalten der beiden Blochwellen. Diejenige, deren größte Aufenthaltswahrscheinlichkeit am Ort der Atome ist, wird „anomal“ stark absorbiert. Daher werden auch die beiden Amplituden der beiden Blochwellen mit zunehmender Dicke unterschiedlich groß. Die Blochwelle am Ort der Atome befindet sich in einem Potenzialloch, wodurch sie stärker beschleunigt wird. Bei gleicher

Gesamtenergie der beiden Blochwellen ist damit die kinetische Energie der einen Blochwelle größer und die potenzielle Energie dieser Blochwelle kleiner als die der anderen Blochwelle. Damit unterscheiden sich die beiden Wellenvektoren der Blochwellen.

Wenn man diese Tatsachen berücksichtigt, lassen sich die beiden Intensitäten des Primärstrahls I_P und des abgelenkten Strahls I_G folgendermaßen darstellen [63]:

$$I_P = \frac{\exp(-\mu_0 \cdot D)}{2 \cdot (1 + w^2)} \left[(1 + 2w^2) \cosh \frac{\mu \cdot D}{\sqrt{1 + w^2}} + 2w\sqrt{1 + w^2} \sinh \frac{\mu \cdot D}{\sqrt{1 + w^2}} + \cos \left(2\pi \frac{\sqrt{1 + w^2}}{\xi_g} D \right) \right]$$

$$I_G = \frac{\exp(-\mu_0 \cdot D)}{2 \cdot (1 + w^2)} \left[\cosh \frac{\mu \cdot D}{\sqrt{1 + w^2}} - \cos \left(2 \cdot \pi \cdot \frac{\sqrt{1 + w^2}}{\xi_g} \right) \cdot D \right], \quad (33) \text{ und } (34)$$

mit μ_0 = Normale Absorption,

μ = Anomale Absorption,

$w = \frac{s}{\sigma}$ = Normalisierter Anregungsfehler,

D = Probendicke

und ξ_g = Extinktionslänge.

Wie aus den beiden Gleichungen ersichtlich ist, liegt die gesamte Intensität an der Probenoberseite ($D = 0$) im Primärstrahl. Mit zunehmender Dicke verlagert sich die Intensität vom Primärstrahl zum gebeugten Strahl. Dabei durchläuft sie beim Primärstrahl ein Minimum und beim gebeugten Strahl ein Maximum. Nachdem diese Extremwerte erreicht worden sind, kehrt sich der ganze Vorgang um. Man spricht dementsprechend von einer Pendellösung, da das System in der Braggschen Beugungsbedingung ($w = 0$) zwischen Minima und Maxima mit einer Periodizitätslänge von ξ_g oszilliert. Die Pendellösung der Intensität wird von Absorption in der Probe gedämpft, so dass bei ausreichender Dicke schließlich kein Kontrastunterschied im Bild mehr zu sehen ist (s. Abbildung 10) [63].

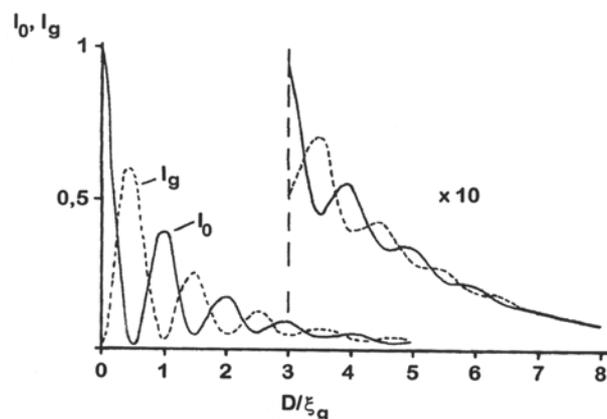


Abbildung 10: Intensitätsverlauf der Primärstrahls I_0 und des gebeugten Strahls I_g in Abhängigkeit von der Probendicke D unter Bragg-Bedingung.

2.1.8.2 Die Multislice Methode

Die Elektronenbeugungsintensitäten können auch mit der Multislice Methode [64], welche ein iteratives Verfahren ist, quantitativ berechnet werden. Dabei wird der Kristall in kleine Scheiben senkrecht zur Einstrahlrichtung der Elektronen unterteilt angenommen, wobei deren Anzahl und Dicke abhängig von den Gitterparametern und von der Probendicke gewählt wird.

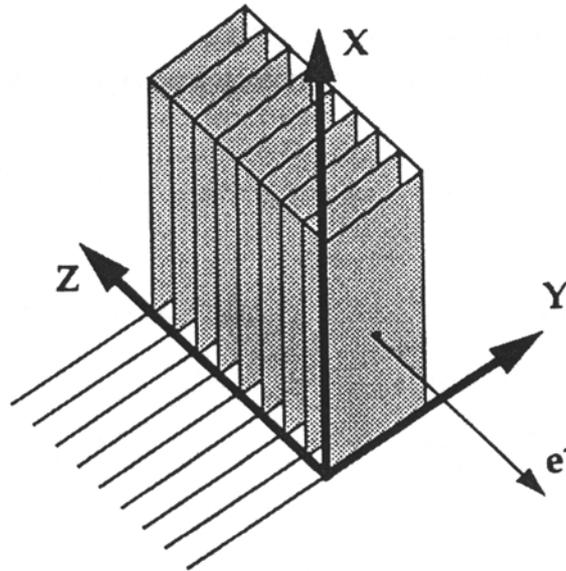


Abbildung 11: Einteilung eines Kristalls in Scheiben nach [39].

Wie bereits erwähnt ändern sich die Phase und die Amplitude einer Elektronenwelle beim Durchgang durch eine Probe. Hingegen kann für dünne Kristalle die Absorption und die damit verbundene Amplitudenänderung vernachlässigt werden. Man spricht hier von der Näherung des schwachen Phasenobjektes. Die Phasenänderung ist vom Brechungsindex und damit von dem Potenzial des Objektes abhängig. Sie wird für jede einzelne Scheibe durch die Transmissionsfunktion (35) berechnet, indem das Potenzial q auf eine Ebene projiziert wird:

$$q_n(x, y) = \exp[-i \cdot \sigma \cdot \Phi_n(x, y) \cdot \Delta z], \quad (35)$$

mit $\sigma = \left[2 \cdot \pi \cdot m_0 \cdot e \cdot \lambda \cdot \left(1 + \frac{e \cdot U}{m_0 \cdot c^2} \right) \right] \cdot h^{-2}$ = Relativistisch korrigierte Wechselwirkungskonstante

und $\Phi_n(x, y) = \frac{1}{\Delta z} \cdot \int_{z_n}^{z_n + \Delta z} \Phi(x, y, z) dz$ = gemittelttes Potenzial der Scheibe.

Die z-Achse liegt dabei parallel zur Einstrahlrichtung der Elektronen.

Da der Abstand der einzelnen Scheiben sehr klein ist und als Vakuum betrachtet wird, müssen bei der Ausbreitung der Welle zwischen den Ebenen Fresnelsche Beugungseffekte berücksichtigt werden. Mathematisch bedeutet dies eine Faltung der Wellenfunktion mit der Propagationsfunktion p :

$$p_n(x, y) = \exp\left[\frac{-\pi \cdot (x^2 + y^2)}{\Delta z \cdot \lambda}\right]. \quad (36)$$

Die Propagationsfunktion p beinhaltet eine parabolische Näherung der eigentlich kugelförmigen Ausbreitung der Wellenfunktion.

Die einfallende ebene Welle kann folgendermaßen dargestellt werden:

$$\Psi_0(\vec{r}) = \exp\left[-2 \cdot \pi \cdot i \cdot (\vec{k} \cdot \vec{r})\right]. \quad (37)$$

Nach der n -ten Scheibe hat die Wellenfunktion folgende Gestalt:

$$\Psi_n(\vec{r}) = [\Psi_{n-1}(\vec{r}) \otimes p_n(\vec{r})] \otimes q_n(\vec{r}) \quad (38)$$

mit Ψ_{n-1} = Wellenfunktion nach der $(n-1)$. Schicht.

Wegen der Periodizität von q_n bei kristallinen Proben ist bei einer numerischen Lösung die Iteration im reziproken Raum durchzuführen:

$$\Psi_n(\vec{u}) = [\Psi_{n-1}(\vec{u}) \cdot P_{n-1}(\vec{u})] \cdot Q_n(\vec{u}) \quad (39)$$

Da Q_n nur an den Punkten des reziproken Gitters ungleich Null ist, vereinfacht sich die Faltung zum Produkt.

Mit abnehmender Scheibendicke konvergiert die Methode zwar gegen die richtige Lösung, aber in der Praxis muss die Scheibenzahl begrenzt werden, da die zur Verfügung stehende Rechengeschwindigkeit von Computern die Anzahl der Scheiben limitiert. Werden hinreichend dünne Schichten angenommen, so ist die Multislice-Theorie exakt und äquivalent zur Blochwellenmethode.

2.1.9 Beugung unter konvergenten Einstrahlbedingungen

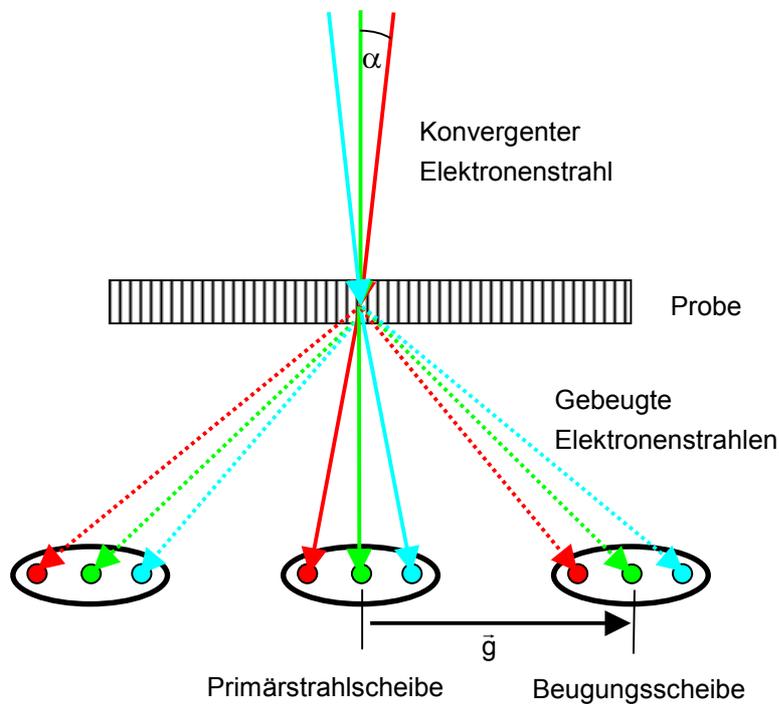


Abbildung 12: Entstehen von konvergenten Beugungsscheiben.

Unter konvergenten Einstrahlbedingungen erstellte Elektronenbeugungsbilder entstehen durch ein Zusammenziehen des Elektronenstrahls auf der Probe. Dazu wird die 2. Kondensorlinse am Mikroskop so angeregt, dass der Elektronenstrahl punktförmig auf der Probe zusammengezogen ist. In diesem Fall entstehen keine Beugungspunkte, wie es bei einer parallelen Beleuchtung der Fall wäre, sondern in Abhängigkeit von der Wahl der C2 Kondensorblende und damit dem Konvergenzwinkel α unterschiedlich große Beugungsscheiben in der hinteren Brennebene der Objektivlinse. Diese Beugungsbilder bestehen aus einer Summe von Punktbeugungsbildern mit mehreren Einstrahlrichtungen (s. Abbildung 12). Aufgrund von dynamischen Effekten können die einzelnen Beugungsscheiben Intensitätsmodulationen aufweisen. Diese Kontrastunterschiede spiegeln die Orientierung und die Symmetrie des beugenden Kristalls wider [6]. Dadurch ist diese Art von Elektronenbeugung ein wertvolles Hilfsmittel in der Elektronenkrystallographie zur Bestimmung von Raumgruppen, da sich Aussagen gewinnen lassen, die über diejenigen von Punktdiffraktogrammen hinausgehen (s. a. Einleitung).

2.1.10 Das Programmpaket ELSTRU

Das Programmpaket ELSTRU beinhaltet die Programme PFMONITOR Version 6.2, XATOMS Version 1.5, GREED Version 2.0 und das Strukturverfeinerungsprogramm MSLS Version 10.0. Es wurde in Delft (Niederlande) von *Jansen et al.* entwickelt und uns zur Verfügung gestellt [37]. Das Programm PFMONITOR ist die Benutzeroberfläche, unter der man die anderen Programme bedienen kann.

Die mit der CCD-Kamera aufgenommenen Punktbeugungsbilder werden mit dem Datenreduktionsprogramm GREED bearbeitet. Zunächst müssen alle übersteuerten Bereiche des Beugungsbildes von der weiteren Bearbeitung ausgeschlossen werden, welches GREED mit einer Routine ermöglicht, deren Parameter einstellbar sind. Weiterhin bietet das Programm eine Suchroutine, mit der die Intensitätsmaxima gefunden werden können. Eine manuelle Markierung der nicht aufgefundenen Reflexe zur späteren Indizierung des gesamten Beugungsbildes ist ebenfalls möglich. Intensitäten im Beugungsbild, die durch Röntgenquanten entstehen und von GREED als Reflexe erkannt werden, können manuell entfernt werden, da diese ein falsches Indizierungsgitter erzeugen. Sind alle nötigen Reflexe gefunden, wird die Beugungsaufnahme zweidimensional indiziert und die Indizierung verfeinert. Nach anschließender Transformation in die dreidimensionale Indizierung erfolgt die Datenreduktion, bei der jedem Reflex (hkl) seine zugehörige Intensität zugeordnet wird. Dadurch wird die ursprünglich 8 MB große Beugungsbilddatei auf eine 2 kB große Eingabedatei für MSLS reduziert.

Mit dem Programm XATOMS können die Atompositionen und deren Besetzungsgrad festgelegt werden. Hier werden auch die Parameter gekennzeichnet, die mit dem Programm MSLS verfeinert werden sollen.

MSLS ist das eigentliche Programm für die Strukturverfeinerung. Es berechnet quantitativ Elektronenbeugungsintensitäten mit der Multislice Methode [64] und vergleicht diese mit den gemessenen Beugungsintensitäten. MSLS kann kristallographische Parameter wie Atomkoordinaten, Besetzungsgrade der Atompositionen, Temperaturfaktoren und für die Intensität in Elektronenbeugungsbildern mitentscheidende Parameter wie Kristalldicke und Lauezentrum des Beugungsbildes verfeinern. Der Verfeinerungsalgorithmus beruht auf der Methode der kleinsten Fehlerquadrate (least squares). Er ist ein linearisierter, nicht-linearer Algorithmus, wie er in der Kristallographie für Strukturverfeinerungen gewöhnlich genutzt wird [65]. Dieser Algorithmus minimiert den Residualwert R , der gleichzeitig die Güte eines Verfeinerungszyklus ausdrückt. R berechnet sich im Programm nach:

$$R = \frac{\sum_m (I_m^{beo} - I_m^{ber})^2}{\sum_m (I_m^{beo})^2}, \quad (40)$$

mit I_m^{beo} = Gemessene Intensität des m-ten Reflexes

und I_m^{ber} = Berechnete Intensität des m-ten Reflexes.

Je kleiner der R-Wert ist, desto besser ist die Übereinstimmung der gemessenen mit den berechneten Intensitäten aus dem vorgegebenen Modell.

Das kleinste Fehlerquadrat-Verfahren berechnet den Wert um den ein freigegebener Parameter variiert wird.

Der Hauptteil des Verfeinerungsalgorithmus ist ein Satz von linearen Gleichungssystemen für die Veränderung der Parameter s :

$$v = M_{ij} \cdot s, \quad (41)$$

mit $M_{ij} = \sum_m w_m \cdot \frac{\delta I_m^{ber}}{\delta p_j} \cdot \frac{\delta I_m^{ber}}{\delta p_i}$ = Verfeinerungsmatrix,

s = Parameterveränderung,

p_i = der i-te zu verfeinernde Parameter,

I_m^{ber} = berechnete Intensität des m-ten Reflexes

und w_m = Wichtungsfaktor des betrachteten Reflexes, der sich so berechnen lässt:

$$w_m = \frac{1}{(\sigma(I))^2}; \text{ mit } \sigma(I) = \text{Standardabweichung der berechneten Intensität.}$$

Der Vektor v_i ist gegeben durch:

$$v_i = \sum_m w_m \cdot (I_m^{ber} - I_m^{beo}) \cdot \frac{\delta I_m}{\delta p_i}, \quad (42)$$

mit den gleichen Variablen wie oben und

I_m^{beo} = beobachtete Intensität des m-ten Reflexes.

Da die Ableitung $I'(p)$ analytisch nicht gelöst werden kann, wird auf das folgende numerische Verfahren zurückgegriffen:

$$\lim_{\delta \rightarrow 0} \frac{I(p + \delta) - I(p)}{\delta} = I'(p) \quad (43)$$

Der in jeder Verfeinerung übliche Skalenfaktor S , der, ohne dass physikalische und strukturelle Parameter geändert werden, die berechneten den gemessenen Intensitäten anpasst, wird in MSLS folgendermaßen definiert:

$$S = s \cdot \frac{\sum_{H \neq 0} I_H^{obs}}{\sum_{H \neq 0} I_H^{cac}} \quad (44),$$

mit s = Parameterveränderung.

Damit erhält man für den Skalenfaktor idealerweise den Wert 1.0.

MSLS kann Verfeinerungen nur in der Miller-Notation durchführen, was im hexagonalen Kristallsystem eine Umrechnung von der dort eindeutigen Miller-Bravais Notation notwendig macht. Für eine erfolgreiche Strukturverfeinerung sind Aufnahmen in mehreren Zonenachsen nötig. In den Hauptzonenachsen der zu untersuchenden Verbindung sind Aufnahmen von mindestens zwei unterschiedlichen Kristalldicken erforderlich.

2.1.11 Die Miller-Bravais Notation

In einem hexagonalen Gitter sind Richtungen kristallographisch gleichwertig, die in der Dreier-Notation unterschiedliche Miller-Indizes besitzen. Mit der Definition eines zusätzlichen Basisvektors in der Basalebene (a_3 in Abbildung 13), der die hexagonale Symmetrie der Basisfläche berücksichtigt, kann die Äquivalenz aller Richtungen und Ebenen in einem hexagonalen System mit gleichen Miller-Indizes in der Vierer-Notation gezeigt werden.

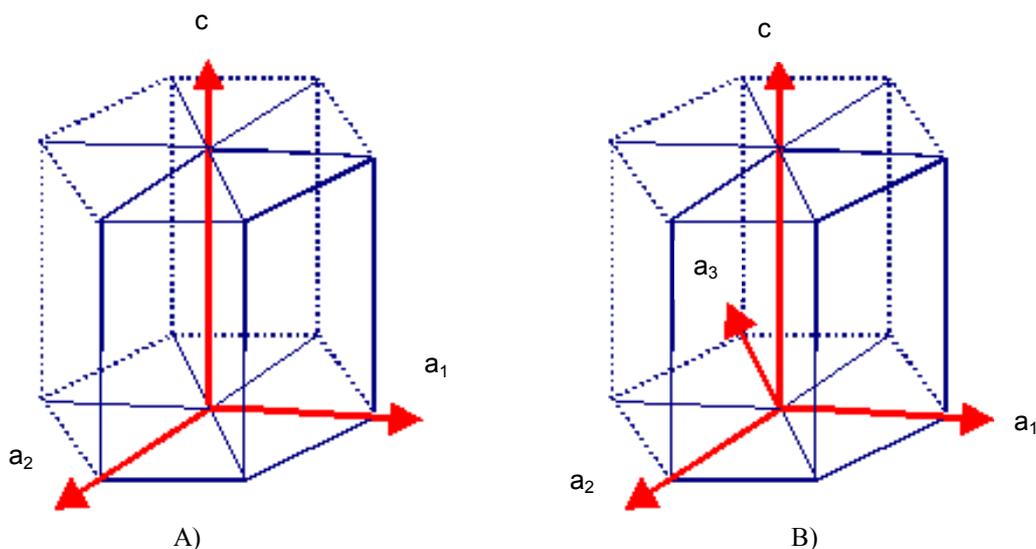


Abbildung 13: Zur Herleitung der Miller-Bravais-Indizierung. A) = ursprüngliche Elementarzelle der Miller-Notation. B) = Neue Elementarzelle mit der Miller-Bravais-Notation.

Die Indizes der Vierer-Notation berechnen sich aus denen der Dreier-Notation nach einem bestimmten Schema. Kristallrichtungen (= Zonenachsen) werden nach folgender Regel berechnet:

$$\begin{array}{ccc} [uvw] & \longrightarrow & [UVTW] \\ U & = \frac{1}{3} \cdot (2 \cdot u - v) & [45] \end{array}$$

$$V = \frac{1}{3} \cdot (2 \cdot v - u) \quad [46]$$

$$T = -(u + v) \quad [47]$$

$$W = w. \quad [48]$$

Das Umrechnen der Indizierung von Netzebenen ist einfacher, da sich die drei ursprünglichen Indizes (hkl) nicht ändern und sich der vierte Index (i) in (hkil) nach der Gleichung $i = -(h+k)$ berechnen lässt.

2.2 Das Transmissionselektronenmikroskop

Während z.B. in der Röntgenbeugung ein pulverförmiges Präparat nur in seiner Gesamtheit untersucht werden kann, können in der Transmissionselektronenmikroskopie (TEM) Beugungsinformationen von einzelnen Kristallen desselben Pulvers erhalten werden. Durch diese lokale Methode ist es auch möglich, gezielt Informationen von einer Phase aus einem Phasengemisch zu erhalten. Die TEM liefert dabei nicht nur Strukturinformationen aus Beugung und Abbildung, sie kann auch über weitere Methoden, wie z.B. EDX (s. Kap. 2.2.2) Aufschlüsse über die chemische Zusammensetzung einer Phase liefern.

Alle hier vorgestellten experimentellen Arbeiten wurden an einem Transmissionselektronenmikroskop der Firma Philips vom Typ CM300 FEG mit einer Ultra-Twin-Linse durchgeführt. Es besitzt eine Feldemissionsquelle, arbeitet mit einer Beschleunigungsspannung von 300 kV und ist sowohl mit einer energiedispersiven Röntgenanalytik (EDX) als auch mit einem abbildenden Energiefilter der Firma Gatan ausgestattet.

Ein Transmissionselektronenmikroskop kann wie ein Lichtmikroskop aus folgenden funktionalen Einheiten aufgebaut beschrieben werden:

1. Beleuchtungssystem, bestehend aus Elektronenquelle, Kondensorlinsen und Kondensorblenden
2. Probenbereich
3. Abbildendes System, bestehend aus Objektivlinse und -blenden und Projektivlinsen zur Nachvergrößerung
4. CCD-Kamera (s. Kap. 2.2.1) o.ä. Detektor zur Aufzeichnung von Daten

In Abbildung 14 ist der schematische Aufbau eines Transmissionselektronenmikroskops wiedergegeben. Es bestehen geringfügige Abweichungen zum benutzten Transmissionselektronenmikroskop CM 300, wo noch eine weitere Kondensorblende (C2) existiert. Die Wahl dieser C2-Kondensorblende hat bei dem CM300 einen wesentlichen Einfluss auf den erreichbaren minimalen Elektronenstrahldurchmesser, der für die quantitative Elektronenbeugung sehr wichtig ist.

Die Elektronen, die von der Kathode im Wehneltzylinder ausgehen, werden zur Anode hin beschleunigt. Als Elektronenemitter dient im CM300 eine Schottky-Emitter-Kathode, welche aus mit Zirconiumoxid beschichtetem Wolfram besteht. Die Beschichtung mit Zirconiumoxid setzt die Austrittsarbeit der Elektronen aus Wolfram, das in [100] orientiert ist, auf 2.8 eV herab. Im Gegensatz zu einer kalten Emissionsquelle wird der Schottky-Emitter bei einer Temperatur von 1800° C betrieben. Durch eine anliegende Extraktionsspannung von ca. 4 kV treten die Elektronen unterstützt von der thermischen Energie aus der Kathode aus. Die Vorteile einer Feldemissionsquelle gegenüber konventionellen Glühkathoden liegen zum einen in der geringen Breite der Energieverteilung der Elektronen von ca. 0.8 eV. Zum anderen besitzt die Feldemissionsquelle eine geringe Ausdehnung der virtuellen Elektronenquelle, deren Größe durch den extrapolierten Schnittpunkt der elektrischen Feldlinien bestimmt wird, welche innerhalb der Kathode ($\varnothing = 0.5 \mu\text{m}$) liegt und wenige nm betragen kann. Dadurch erhält man einen sehr kleinen, aber trotzdem intensiven Elektronenstrahl. Schließlich ist auch die Kohärenz der Elektronen bei einer Feldemissionsquelle im Vergleich mit einer Glühemissionsquelle größer.

Das System von Kondensoren und Kondensorblenden erlaubt es, die Probe nach den jeweiligen experimentellen Anforderungen zu beleuchten. Die Probe wird durch eine Schleuse (side entry) in das Transmissionselektronenmikroskop eingebracht und befindet sich zwischen den beiden Objektivlinsen. Der verwendete Doppelkipp-Probenhalter ermöglicht es, die Probe erstens senkrecht zum Elektronenstrahl hin und her zu bewegen und zweitens sie um die beiden Achsen, die senkrecht zueinander und jeweils senkrecht zum Elektronenstrahl liegen, zu kippen.

Die Objektivlinse ist die wichtigste Linse im Transmissionselektronenmikroskop, da diese den größten Vergrößerungsschritt realisiert. Das Auflösungsvermögen in der TEM ist nicht wie in der Lichtmikroskopie von der benutzten Wellenlänge sondern von den Linsenfehlern der Objektivlinse und der Kohärenz der Elektronen limitiert. Alle Linsen im Transmissionselektronenmikroskop sind elektromagnetische Linsen, die ein inhomogenes, rotationssymmetrisches Magnetfeld besitzen. Sie bestehen aus gewickelten Spulen mit Eisenpolschuhen. Durch Variation der Linsenströme werden verschieden starke Magnetfelder erzeugt, wodurch sich die Elektronenbahnen verändern und woraus verschiedene Brennweiten der Linsen resultieren.

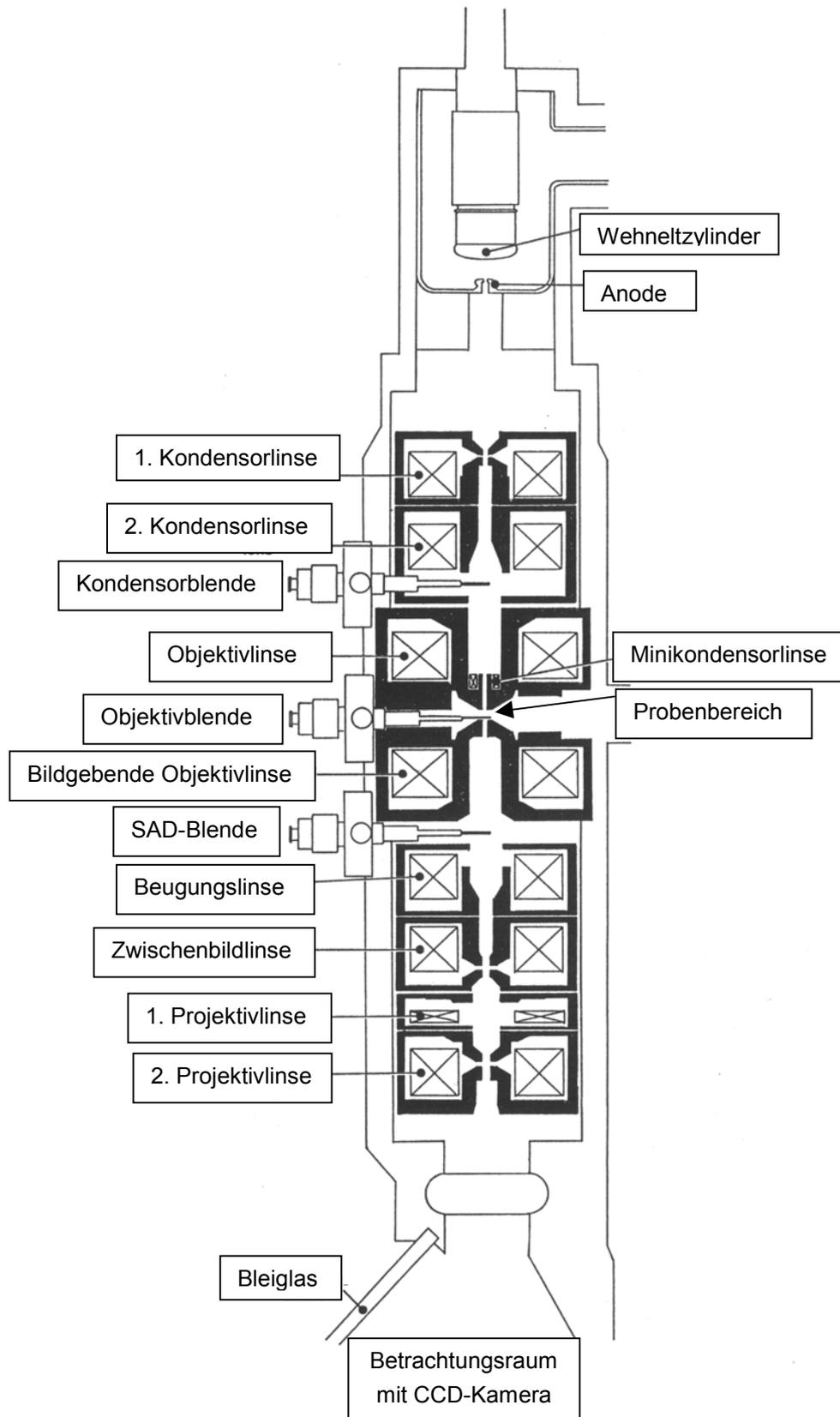


Abbildung 14: Schematischer Aufbau eines Transmissionselektronenmikroskops nach [66].

In der hinteren Brennebene der Objektivlinse entsteht das Beugungsbild, d.h. Elektronenstrahlen, die von der Probe unter einem bestimmten Winkel gebeugt werden,

werden in einem Punkt vereint. Hier ist es durch die Objektivblende möglich, Bilder zu erzeugen, die nur von bestimmten gebeugten Strahlen herrühren. Wird die Objektivblende z.B. um den Primärstrahl gesetzt, so wird ein Hellfeldbild mit einem höheren Kontrast erhalten als ohne Blende.

Das Projektivlinsensystem kann in zwei Modi betrieben werden. Im Abbildungsmodus wird das Zwischenbild, im Beugungsmodus das Beugungsbild in das weitere Linsensystem übertragen. Durch Einbringen einer Feinbereichsblende (SAD = selected area diffraction) in die erste Zwischenbildebene ist es möglich, aus bestimmten Bereichen der Probe Beugungsinformationen zu erhalten. Nur von dem ausgewählten Bereich wird dann Beugungsinformation erhalten.

Zur Visualisierung der vergrößerten Elektronenstrahldichte dient schließlich ein fluoreszierender Beobachtungsschirm. Die Datenaufnahme erfolgt mit einer $2k \times 2k$ CCD-Kamera, welche im folgenden Kapitel beschrieben wird.

2.2.1 Die CCD-Kamera

Es stehen am Transmissionselektronenmikroskop CM300 zwei CCD-Kameras zur Verfügung. Die Multi-Scan-Camera vom Typ MSC 794 der Firma Gatan besteht aus einer dünnen mit Fluoreszenzmaterial, welche über lichtleitenden Fasern mit einem in 2048×2048 unterteilten Halbleiterchip verbunden sind. Jedes quadratische Pixel hat eine Größe von $30 \times 30 \mu\text{m}$. Damit ist die Kamera ca. $6 \times 6 \text{ cm}$ groß. Solch eine CCD-Kamera hat gegenüber fotografischen Filmen im wesentlichen fünf Vorteile:

- Direkter Zugriff auf digitale Daten,
- ausreichende Empfindlichkeit für hochenergetische Elektronen,
- hohes Auflösungsvermögen,
- ein lineares Ansprechverhalten bis zur Sättigung und
- mit 2^{14} (= 16384) Graustufen eine recht hohe Dynamik.

Die Fiberoptik führt das Licht, das durch die einfallenden Elektronen auf die Szintillatorschicht entstanden ist, auf den auf ca. -30°C gekühlten CCD-Chip. Dort erzeugt ein ankommendes Photon Elektronen-Loch-Paare. Diese werden auf dem Chip pixelweise ausgelesen und dann in ein digitales Signal umgewandelt. Die Kühlung des CCD-Chips minimiert die thermische Anregung von Elektronen im Halbleitermaterial. Die Slow-Scan-Kamera (SSC) der Firma Gatan besitzt im Gegensatz zur Multi-Scan-Kamera 1024×1024 Pixel.

2.2.1.1 Korrektur der Zählrate der CCD-Kamera

Trotz der Kühlung fließt auf dem CCD-Chip noch ein Dunkelstrom, der zu einer falschen gemessenen Intensität im experimentellen Bild führt. Zur Korrektur wird die

Zählrate des CCD-Chips ohne Exposition mit Elektronen aufgenommen, als Dunkelstrombild gespeichert und von den Rohdaten einfach subtrahiert. Darüber hinaus ist aber noch eine weitere Korrektur des experimentellen Bildes nötig. Diese berücksichtigt die unterschiedliche Empfindlichkeit der Pixel auf dem CCD-Chip bei gleicher Belichtungszeit.

Das korrigierte experimentelle Bild wird mit folgender Formel erhalten:

$$I_{korr} = (I_{roh} - I_{dunkel}) \cdot \frac{I_{gain} - I_{dunkel}}{I_{gain} - I_{dunkel}} \quad (49)$$

mit I_{roh} = Rohdaten des CCD-Chips,

I_{dunkel} = Intensitäten, die ohne Elektronenexposition aufgenommen wurde

und I_{gain} = Intensität der Pixel, welches bei homogener Ausleuchtung und gleicher Belichtungszeit aufgenommen wurde.

2.2.1.2 Abbildungseigenschaft der CCD-Kamera

Ein wichtiges Merkmal einer CCD-Kamera ist, wie punktgenau sie ein ideal punktförmiges Signal, welches kleiner ist als ein Pixel der CCD-Kamera selbst, wiedergeben kann, ohne dieses wesentlich zu verbreitern. Eine Antwort auf diese Frage liefert die sogenannte Punktbildfunktion. Eine horizontale Pixelreihe der Punktbildfunktion ist in Abbildung 15 dargestellt, die eine Intensitätsverteilung eines punktförmigen Signals auf dem CCD-Chip zeigt.

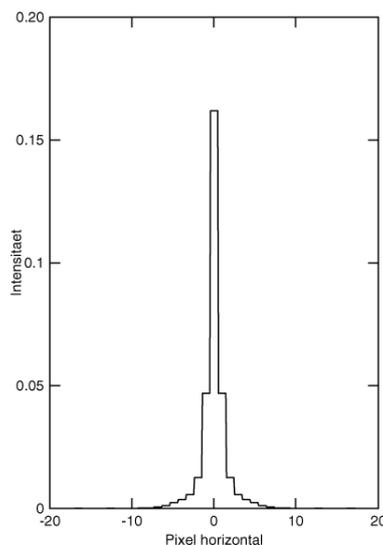


Abbildung 15: Punktbildfunktion der verwendeten Gatan Multi-Scan-Camera MSC Typ 794. Horizontale Verteilung der Gesamtintensität 1.

In der Abbildung 16 sind der zentrale Pixel Z und seine vier nächsten Nachbarn A abgebildet. Z und A besitzen laut Abbildung 15 knapp 46 % der eingestrahelten Intensität. Die vier folgenden nächsten Nachbarn B sind in der Abbildung 15 nicht erfasst, da sie sich nicht auf der Horizontalen befinden. Die Intensität dieser Pixel B müsste aber zwischen den Intensitäten der Pixel A und C liegen. Schätzungsweise besitzen diese vier Pixel B 9 % Intensität, so dass man, wenn man die Intensitäten der Pixel C mit 6 % ebenfalls addiert, nun auf 61 % der Gesamtintensität kommt. Eine weitergehende Analyse der Intensitätsverteilung eines punktförmigen Signals auf dem CCD-Chip ist nicht sinnvoll, da die zu addierenden Intensitäten zu gering werden (s. Abbildung 15). In Abbildung 15 ist aber noch zu erkennen, dass nach ± 8 Pixel vom betrachteten zentralen Pixel keine signifikante Intensität mehr vorhanden ist.

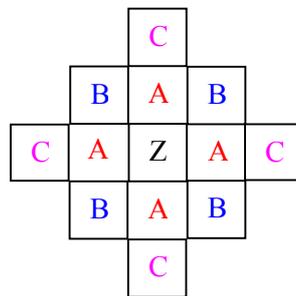


Abbildung 16: Pixel einer CCD-Kamera.

Zusammenfassend lässt sich sagen, dass die Intensitätsdelokalisierung eines punktförmigen Signals für das quantitative Auswerten von Elektronenbeugungsbildern klein genug ist.

2.2.2 Die energiedispersive Röntgenanalyse

Mit Hilfe der energiedispersiven Röntgenanalyse lassen sich kleine Probenbereiche auf ihre chemische Zusammensetzung sowohl qualitativ als auch quantitativ analysieren. Am Transmissionselektronenmikroskop CM300 besteht die Möglichkeit, den EDX-Detektor (Firma Noran Instruments) in unmittelbare Nähe der Probe zwischen die beiden Polschuhe zu fahren (s. Abbildung 17).

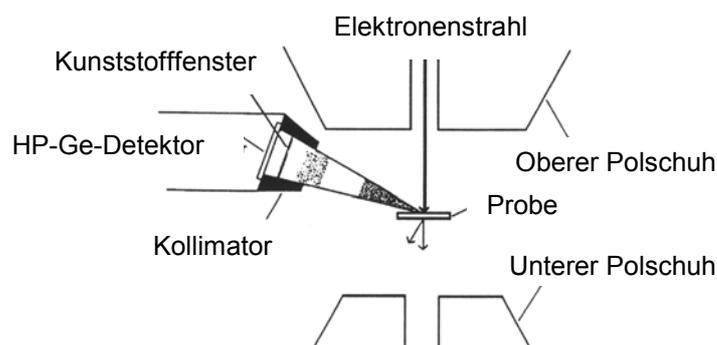


Abbildung 17: Anordnung der EDX-Einrichtung am Transmissionselektronenmikroskop.

2.2.2.1 Grundlagen der EDX

Hochenergetische Elektronen sind in der Lage, Elektronen aus den inneren Schalen eines Atoms herauszuschlagen. Die so entstandenen angeregten Kationen relaxieren beispielsweise, indem ein Elektron aus einer energetisch höher liegenden Schale in das entstandene „Loch“ fällt. Dabei zeigt das Kation ein für jedes Element charakteristisches Röntgenemissionsspektrum, das aus Liniengruppen verschiedener Frequenzen besteht, die als K-, L-, M-, N-, ... Serien bezeichnet werden. Fällt ein äußeres Elektron auf die K-Schale zurück (Hauptquantenzahl $n = 1$), so emittiert es elektromagnetische Strahlung charakteristischer Wellenlänge, fällt es auf die L-Schale (Hauptquantenzahl $n = 2$) zurück, dann wird ein Röntgenquant mit entsprechend kleinerer Energie freigesetzt, usw. (s. Abbildung 18).

Dies ist aber nicht der einzige auftretende Prozess, wenn hochenergetische Elektronen auf Atome treffen. Das durch den Elektronenbeschuss entstandene Kation kann sich auch durch Abstraktion eines weiteren Elektrons stabilisieren. Dieser sogenannte Auger-Effekt tritt überwiegend bei den leichteren Elementen auf. Auger-Elektronen besitzen ebenfalls eine für die jeweiligen Elemente charakteristische Energie.

Die in Abbildung 18 dargestellten „Energieverlustelektronen“ können ebenfalls zur Spektroskopie verwendet werden.

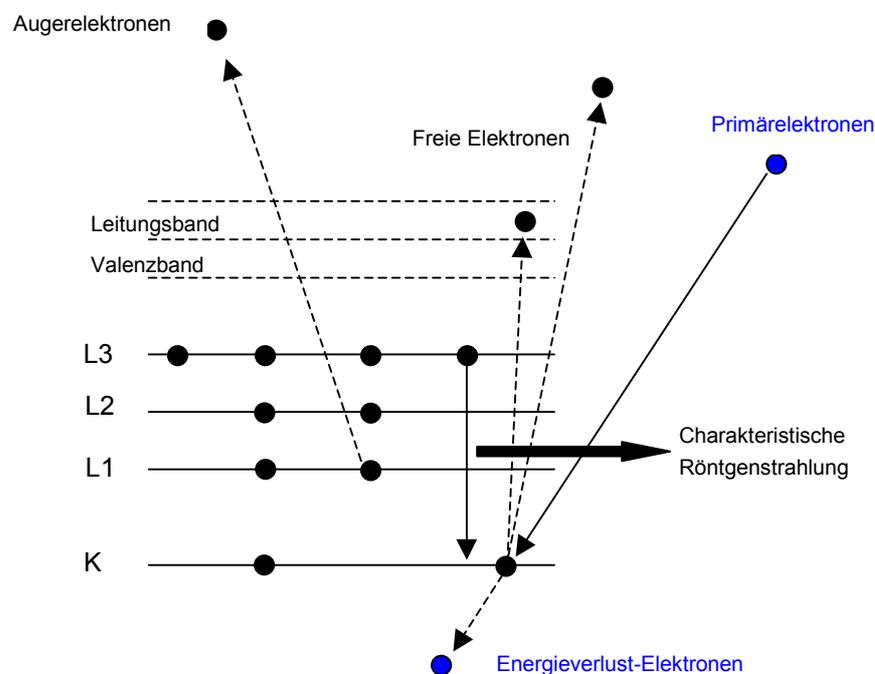


Abbildung 18: Inelastische Streuung von Elektronen und folgende Sekundärprozesse [68].

2.2.2.2 Aufbau und Funktionsweise der EDX

Der EDX-Detektor besteht aus einem lithiumdotierten Silizium- oder einem hochreinen Germaniumeinkristall, der mit Gold beidseitig bedampft wurde, wobei dieses jeweils als Elektrode dient. Treffen Röntgenstrahlen auf den Detektor, so bewirken sie ein Anheben von Valenzelektronen in das Leitungsband des Detektormaterials. Die so entstandenen Elektronen-Loch-Paare werden über eine Hochspannung, die an den beiden Goldelektroden angelegt ist, getrennt, bevor sie rekombinieren können. Der resultierende Stromfluss, dessen Stärke zu der Energie des eingefallenen Röntgenquants proportional ist, wird über einen Feldeffekttransistor als Vorverstärker in einen Stromstoß umgewandelt. Die Impulse werden weiter verstärkt und gelangen in den Vielkanalanalysator, wo sie entsprechend ihrer Energie sortiert werden. Die Auswertung und die Darstellung der Signale erfolgt computerunterstützt.

Da die mittlere Energie zur Erzeugung eines Elektronen-Loch-Paares bei einem lithiumdotierten Siliziumeinkristall 3.6 eV und bei einem Germaniumeinkristall nur 2.8 eV beträgt, müssen die Detektoren mit flüssigem Stickstoff gekühlt werden. Anderenfalls würde wegen der geringen Energielücke das thermische Rauschen das Signal-zu-Untergrundverhältnis verschlechtern und so die Messzeit verlängern. Zudem wird durch die Kühlung die Auflösung verbessert.

Der EDX-Detektor wird durch ein Fenster aus Beryllium oder Parylen, die sich beide durch eine schwache Absorption von Röntgenstrahlung auszeichnen, gegen das Vakuum des Transmissionselektronenmikroskops und gegen Kontamination geschützt (s. Abbildung 17).

2.3 Probenpräparation

Da für die Untersuchung im Transmissionselektronenmikroskop elektronentransparente d.h. sehr dünne Proben zur Verfügung stehen müssen, ist die Präparation eine in mehrere Schritte unterteilte Arbeit, die im folgenden beschrieben wird. Es gibt für die Herstellung von dünnen Präparaten mehrere unterschiedliche Verfahren, die je nach Probenmaterial und Fragestellung herangezogen werden.

Verfügt man über eine pulverförmige Probe, so wird diese in Aceton aufgeschlemmt und in einem Mörser verrieben. Auf einen amorphen, elektronentransparenten Kohlenstofffilm, welcher von einem Kupfernetz getragen wird, kann man einen Tropfen dieser Suspension aufbringen. Wenn das Aceton verdampft ist, kann diese Probe in ein Transmissionselektronenmikroskop eingebaut werden.

2.3.1 Vollpräparation

Der erste Schritt einer Vollpräparation ist das Ausbohren einer 3 mm großen, runden Scheibe aus der zu untersuchenden Probe (Abbildung 19a). Diese Scheibe wird von beiden Seiten planparallel bis zu einer Restdicke von ca. 100 μm geschliffen (Abbildung 19b). Anschließend wird mit einem Muldenschleifgerät (Firma Gatan Typ 656, s. Abbildung 20) eine Mulde in die Probe geschliffen (Abbildung 19c).

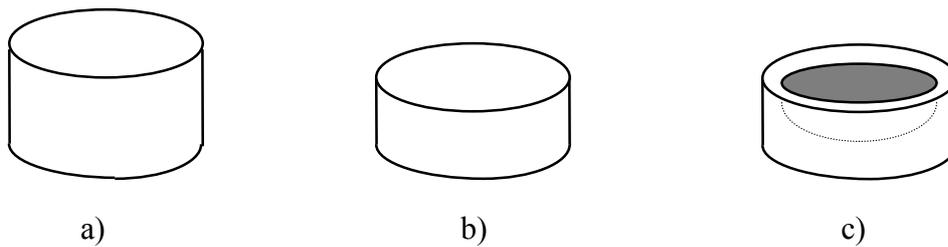


Abbildung 19: Schematische Darstellung der einzelnen Arbeitsschritte der Vollpräparation.

Durch das Muldenschleifen wird eine Restdicke von ca. 30 μm erreicht. Wenn die Probe insgesamt auf 30 μm heruntergeschliffen würde, würde diese für die weitere Verarbeitung zu instabil sein.

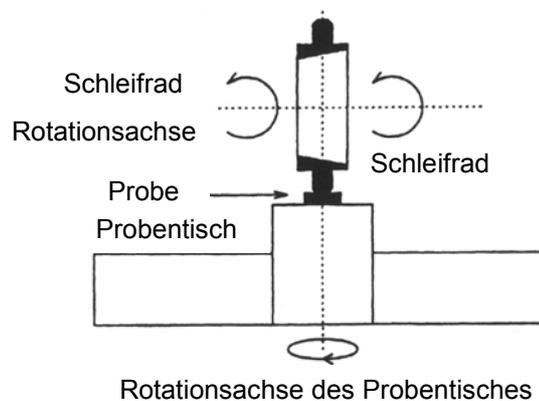


Abbildung 20: Schematische Darstellung des Muldenschleifgerätes.

Abschließend wird in die so bearbeitete Probe mit einem Argonionenstrahl ein Loch geätzt, so dass dann die Randbereiche des Probenloches für die TEM hinreichend dünn sind.

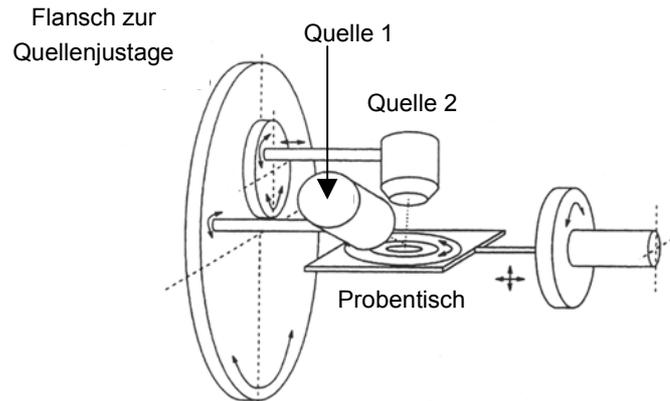


Abbildung 21: Schematischer Aufbau der Ionenätzanlage (RES-010 Firma Baltec).

Folgende Parameter können zur Optimierung des Ätzvorganges in erster Linie herangezogen werden:

- 1) die Beschleunigungsspannung der Argonionen,
- 2) die Größe des Ionenflusses (\equiv eingestellte Stromstärke) und
- 3) der Einfallswinkel des Ionenstrahls auf die Probenoberfläche.



3 Spezieller Teil

3.1 Erweiterung des Programmpaketes ELSTRU

Die Elektronenkristallographie ist ein wertvolles Hilfsmittel zur Strukturlösung, wenn mit den etablierten Röntgenmethoden keine hinreichenden Informationen erhalten werden können. So ist es in der Festkörperchemie immer ein Ziel Einkristalle zu erzeugen, die dann routinemäßig mit einem Vierkreis-Diffraktometer untersucht werden. Gelingt es nicht Einkristalle zu erhalten, so kann in der Röntgenanalytik die neue Verbindung nur in ihrer Gesamtheit mit Hilfe der Pulverdiffraktometrie untersucht werden. Die fehlenden Informationen über Symmetrieelemente, mit denen schließlich die Raumgruppe einer Substanz ermittelt werden kann, können z.B. mit Hilfe von CBED-Aufnahmen (Convergent Beam Electron Diffraction) erhalten werden [6].

Eine Strukturverfeinerung eines Strukturvorschlages mittels Elektronenbeugungsdaten war erst seit 1998 mit dem Programmpaket ELSTRU [37] vernünftig möglich. MSLS (Multi Slice Least Square) benutzt als erstes Verfeinerungsprogramm mit Elektronenbeugungsdaten die physikalisch richtige dynamische Berechnung der Reflexintensitäten. Es bedient sich dabei der Multi-Slice-Methode und als Optimierungsalgorithmus dem „Least Square“-Verfahren. Seitdem können Daten aus Punktbeugungsbildern zur Strukturverfeinerung verwendet werden. Die Aufzeichnung dieser Bilder verlangt aber spezielle Aufnahmebedingungen. So ist es eine notwendige Voraussetzung, dass die beleuchtete Probenstelle von homogener Kristalldicke ist, da sie ein mitentscheidender Parameter zur Reflexintensität ist. Deswegen ist auch die Probendicke ein verfeinerbarer Parameter innerhalb von MSLS. Um die Bestrahlung einer Probenstelle mit homogener Kristalldicke zu gewährleisten, ist die Verwendung eines sehr feinen Elektronenstrahl (\varnothing ca. 5 nm) damit unabdingbar. Der minimale Elektronenstrahldurchmesser liegt bei der Verwendung von parallelen Elektronenstrahlen, die zu Punktbeugungsbildern führen, bei ca. 10 nm. Dadurch ist die Erstellung verwertbarer Datensätze erschwert.

Bei den verwertbaren Probendicken ist der Primärstrahl von Punktbeugungsbildern in der Regel übersättigt. Überstrahlte Intensitäten können aufgrund des Ausleseverfahrens der CCD-Kamera in die Beugungsintensitäten hinein reichen (s. Abbildung 22 links). Die Intensitäten des Primärstrahls und die der beeinflussten Reflexe können damit nicht mit in die Strukturverfeinerung einbezogen werden. Eine Verringerung der Aufnahmezeit bewirkt zwar, dass die Delokalisierung der übersättigten Intensität kleiner wird, aber die Anzahl der verwertbaren Reflexe wird gleichzeitig erniedrigt. Damit muss das optimale Mittel unter diesen Bedingungen bei der Aufnahme von Punktbeugungsbildern gefunden werden.

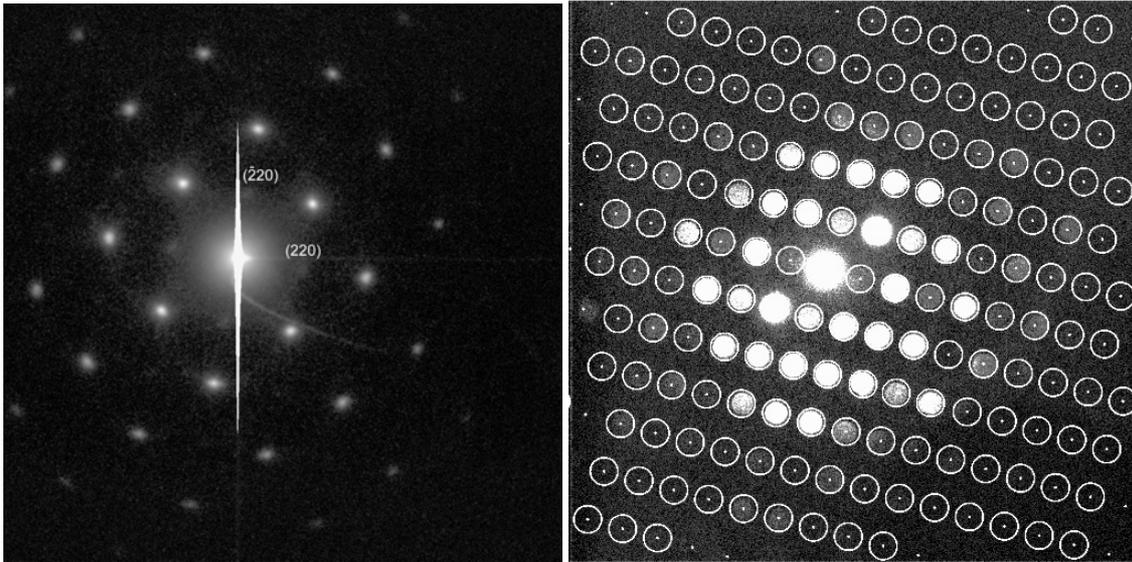


Abbildung 22 Links: Beugungsbild von $\gamma\text{-Al}_2\text{O}_3$ in $[001]$ mit einem parallelem Elektronenstrahl. Rechts: $\text{Sc}_3\text{Sc}_{0.32}\text{Al}_{0.68}\text{O}_6$ in $[2\bar{1}\bar{1}0]$ mit angepassten Ringen und dem ausgelesenen - in diesem Fall zentralen - Bereich (Auslesekoordinaten: $x = 0, y = 0$).

Da die Intensität in Beugungsbildern, die unter konvergenten Einstrahlbedingungen aufgezeichnet werden, auf eine größere Fläche verteilt ist, treten Übersättigungsprobleme des CCD-Chips bei gleichen Belichtungszeiten im Vergleich zu Punktbeugungsbildern nicht auf. Zudem ist der Elektronenstrahl auf der Probe bei diesen Beugungsbildern kleiner (\varnothing ca. 5 nm) als bei paralleler Beleuchtung, was die Bestrahlung einer homogenen Probendicke wahrscheinlicher macht. Da aber die Intensitäten der Reflexscheiben dieser Beugungsbilder sich aus mehreren Einstrahlrichtungen zusammensetzen (s. Abbildung 12), können sie nicht direkt im Programmpaket ELSTRU verarbeitet werden. Die Einstrahlrichtung oder analog die Verkippung des Beugungsbildes, im Programm MSLS „centre of Laue circle“ (= Lauezentrum) genannt, muss ein verfeinerbarer Parameter sein, da sie ebenso einen wesentlichen Einfluss auf die Reflexintensitäten besitzt. Demzufolge ist es unvermeidlich, Bragg-korrelierte Bereiche aus sämtlichen Beugungsscheiben eines Beugungsbildes herauszulesen und diese neuen Datensätze als gewissermaßen mit einem parallelen Elektronenstrahl aufgenommene Beugungsbilder unabhängig voneinander mit dem Programmpaket ELSTRU weiter zu bearbeiten. Demnach liegt der Unterschied der verschiedenen Datensätze eines Beugungsbildes in den verschiedenen Einstrahlrichtungen.

Es wurde im Rahmen dieser Arbeit ein Programm mit der Bezeichnung *konv2.pro* unter IDL[®] [38] entwickelt, welches Bragg-korrelierte Intensitätsbereiche aus den Reflexscheiben von Beugungsbildern, die unter konvergenten Einstrahlbedingungen aufgezeichnet wurden, ausliest. Mit diesen erhaltenen Daten erzeugt *konv2.pro* eine Eingabedatei für das Datenreduktionsprogramm GREED. Die weitere Vorgehensweise erfolgt analog zur Auswertung der Punktbeugungsbilder (s. Kap. 2.1.10).

Konv2.pro benötigt als Startwert die ungefähren Koordinaten von den Zentren der Primärstrahlscheibe und der zwei Basisreflexscheiben. Daraus berechnet es zunächst die übrigen danach ebenfalls ungefähren zentralen Positionen der anderen Reflexscheiben. Die optimalen zentralen Endpositionen werden visuell durch Anpassen geeignet großer Ringe um jeden Reflex ermittelt (s. Abbildung 22 rechts), welche innerhalb von *konv2.pro* berechnet werden. Jede Reflexscheibe sollte letztlich dabei konzentrisch von einem Ring umschlossen werden. Die Größe des ausgelesenen kreisförmigen Bereiches kann innerhalb von *konv2.pro* variiert werden.

Aus einem Beugungsbild, das unter konvergenten Einstrahlbedingungen aufgenommen worden ist, werden so in der Regel fünf Datensätze ausgelesen. Mit jedem dieser fünf Datensätze wird unabhängig voneinander eine sogenannte Vorverfeinerung gerechnet. D.h. nach der Datenreduktion mit dem Programm GREED werden mit MSLS und einem Strukturvorschlag in der Reihenfolge Skalenfaktor, Skalenfaktor und Lauezentrum, Skalenfaktor und Probedicke und schließlich alle eben genannten Parameter freigegeben und damit verfeinert. Die Ergebnisse der einzelnen Datensätze von einem Beugungsbild müssen einige Kriterien erfüllen. Die verfeinerten Werte der Skalenfaktoren sollten ebenso wie die Resultate der Kristalldicken zu einem Wert konvergieren, da die Aufnahmebedingungen der fünf Datensätze identisch waren und sie von derselben Probenstelle stammen. Schließlich sollten die verfeinerten Ergebnisse der Lauezentren die jeweilige Einstrahlrichtung der fünf Datensätze widerspiegeln. Sind alle diese Kriterien erfüllt, so wird jeweils ein Datensatz von unterschiedlichen Beugungsbildern zur eigentlichen Strukturverfeinerung verwendet. Diese neue Methode wurde zunächst an einem simulierten konvergenten Beugungsbild (Korund oder α - Al_2O_3 in [0001]) getestet, was im folgenden beschrieben ist.

3.1.1 Ergebnisse mit einem unter konvergenten

Einstrahlbedingungen simulierten Beugungsbild

Das unter konvergenten Einstrahlbedingungen errechnete Beugungsbild von Korund wurde mit dem Programmteil CB2 des Programmpakets EMS [39] berechnet. Dabei wurden folgende Simulationsparameter verwendet:

Beschleunigungsspannung	= 300 kV,
Probedicke	= 40 Å,
Zonenachse	= [0001],
Debye-Waller-Koeffizient (B)	= 0,

die weiteren kristallographischen Daten von Korund sind in Tabelle 1 aufgeführt [40]:

Raumgruppe	Gitterkonstanten [Å]	Atom	x/a	y/b	z/c	Besetzungsfaktor	Wykhoff Position
$R\bar{3}c$	$a = 4.776$	Al	0	0	0.3522	1	12 c
Nr. 167	$c = 12.991$	O	0.6914	0	0.25	1	18 e

Tabelle 1: Simulationsparameter des unter konvergenten Einstrahlbedingungen berechneten Beugungsbildes von Korund (α -Al₂O₃).

Fünf Datensätze sind in dem simulierten Beugungsbild (s. Abbildung 23) jeweils mit einer Farbe markiert. Ausgehend von einem Datensatz im Zentrum der Beugungsscheiben (x ; y : grün) wurden weitere vier Intensitätssätze bei ($x + 100$; y : türkis), ($x - 100$; y : rot), (x ; $y + 100$: blau) und (x ; $y - 100$: violett) ausgelesen. Die Verschiebungen sind jeweils in Pixeln angegeben. Der Durchmesser des ausgelesenen kreisförmigen Bereiches eines Datensatzes betrug 20 Pixel.

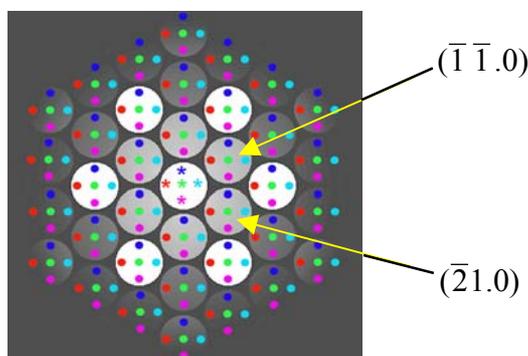


Abbildung 23: Simuliertes konvergentes Beugungsbild von Korund in [00.1] mit den fünf Datensätzen, die farblich mit Kreise markiert sind. Die farbigen Sterne zeigen die verfeinerten Ergebnisse der Lauzentren der jeweiligen Datensätze an.

Datensatz	Anzahl Reflexe	Lauzentrum h k		Probendicke [Å]	B [pm ²]	x/a Sauerstoff	Skalenfaktor	R-Faktor
1 (nahe am Zentrum)	35	0.04(3)	0.09(3)	42.5(1.5)	O: 0 Al: 0	0.6914(17)	1.038(7)	0,06 %
2 ($x + 100$ pixel)	29	-0.47(5)	0.10(3)	42.3(1.6)			1.052(8)	0,07 %
3 ($x - 100$ pixel)	28	0.54(5)	0.10(3)	42.2(1.5)			1.055(8)	0,06 %
4 ($y + 100$ pixel)	31	0.31(4)	-0.47(6)	42.1(1.5)			1.045(8)	0,07 %
5 ($y - 100$ pixel)	30	-0.25(4)	0.67(6)	42.5(1.5)			1.050(9)	0,08 %
Gesamt R-Wert (153 Reflexe) = 0,07 %								

Tabelle 2: Verfeinerungsergebnisse mit einem simulierten konvergenten Beugungsbild von Korund (α -Al₂O₃) in [00.1].

Mit den fünf Datensätzen wurde eine Strukturverfeinerung durchgeführt. Dabei wurde nur die x -Koordinate von Sauerstoff mit dem Startwert $2/3$ freigegeben, da die z -Koordinate von Aluminium in einer $[00.1]$ -Orientierung nicht verfeinert werden kann. Die Ergebnisse dieser Strukturverfeinerung sind in Tabelle 2 gezeigt, wobei die Resultate der einzelnen Datensätze in der Tabelle 2 entsprechend der Abbildung 23 mit der selben Farbe markiert sind.

Die Resultate der Skalenfaktoren und die der Kristalldicken der fünf Datensätze liegen im Rahmen der sehr kleinen Fehler auf den selben Werten. Die in Abbildung 23 als farbige Sterne dargestellten Ergebnisse der Lauezentren zeigen die gute Reproduktion der ausgelesenen Einstrahlrichtungen (s. a. Tabelle 2). Die x -Koordinate der Sauerstoffposition, die bei der Bildsimulation vorgegeben wurde, wird mit einem kleinen Fehler wiedergegeben. Schließlich zeigt der ausgezeichnete R -Wert von 0.07% ebenfalls eine erfolgreiche Verfeinerung an. Die minimale Abweichung von rund 2 \AA bei den Ergebnissen der Kristalldicken (40 \AA waren in der Simulation vorgegeben) ist auf die unterschiedlichen Berechnungsarten von CB2 und MSLS zurückzuführen. CB2 berechnet die Beugungsbilder nach der Blochwellenmethode, während MSLS die Intensitäten nach der Multi-Slice-Methode berechnet. Dieser geringe Unterschied wird auch im R -Wert deutlich, der eigentlich für ein simuliertes Beugungsbild bei Null liegen sollte.

3.1.2 Die Untergrundkorrektur

Das simulierte Beugungsbild wurde ohne Untergrundrauschen berechnet. Experimentell erstellte Beugungsbilder enthalten aber im näheren Umfeld des Primärstrahls eine Untergrundintensität, deren Wert von den Beugungsdaten subtrahiert werden sollte. Innerhalb von *konv2.pro* kann eine Untergrundkorrektur aktiviert werden, deren Prinzip nachstehend erläutert wird. Da die Untergrundintensität nicht direkt gemessen werden kann, muss sie durch Extrapolation bestimmt werden. Sie wird auf beiden Seiten jeder Beugungsscheibe ausserhalb dieser bei demselben Bragg-Winkel wie der auszulesene Bereich des Reflexes ermittelt. Hierfür wird zunächst um jede Beugungsscheibe ein Ring berechnet, der in Abbildung 24 rot gekennzeichnet ist. Ein weiterer in der Abbildung 24 blau gekennzeichnet Ring wird durch jeden ausgelesenen Bereich (grüne Kreise in Abbildung 24) ermittelt. Damit liegt der blaue Ring auf demselben Bragg-Winkel wie die ausgelesenen Bereiche. In den Schnittbereichen des roten Ringes mit dem blauen Ring (gelb gekennzeichnet in Abbildung 24) wird über die beiden Bereiche die mittlere Intensität bestimmt. Die so erhaltene Untergrundintensität wird pixelweise von der ausgelesenen Reflexintensität subtrahiert.

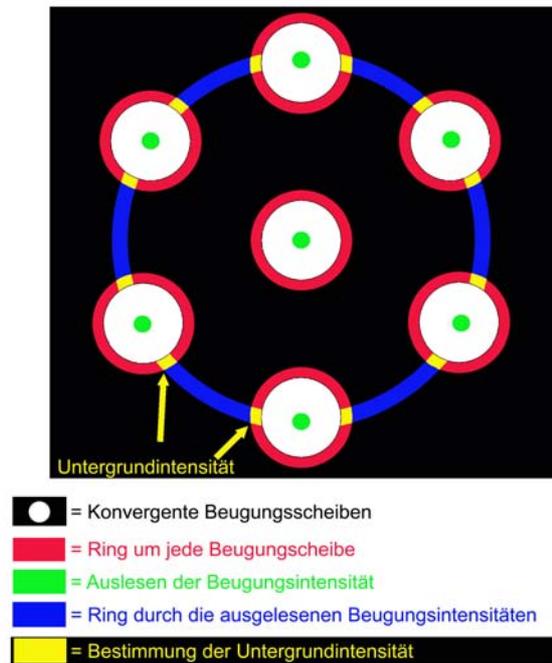


Abbildung 24: Bestimmung der Untergrundintensität.

Eine Ausnahme bildet die Bestimmung der Untergrundintensität bei der Primärstrahlscheibe. Dort wird einfach die mittlere Intensität des roten Ringes um ihn bestimmt (s. Abbildung 24). In der Praxis stellte sich heraus, dass eine Untergrundkorrektur nur bei den benachbarten Reflexe des Primärstrahls nötig ist, wenn diese einen geringen Abstand von der Primärstrahlscheibe besitzen (Beispiel s. Abbildung 22 rechts). Bei den Aufnahmen, die von verwertbaren Kristalldicken erstellt wurden, erreichen die diffus gestreuten elastischen Elektronen eine maximale Zählrate von 30 Counts, was bei einer Primärstrahlreflexintensität von bis ca. 4000 Counts vernachlässigbar ist. Bei eng benachbarten Reflexen, die eine Intensität von bis zu 300 Counts besitzen, muss die Untergrundintensität von der ausgelesenen Intensität subtrahiert werden. Primärstrahlferne Bereiche in Beugungsbildern besitzen eine verschwindend geringe Untergrundintensität, die demnach nicht subtrahiert werden muss. Auch die im Kapitel 2.2.1.2 erwähnte Punktbildfunktion ist klein genug, dass sie keine störende Intensitätsdelokalisierung ausserhalb der stärksten Reflexe verursacht.

3.1.3 Ergebnisse mit einem unter konvergenten Einstrahlbedingungen erstellten experimentellen Beugungsbild

Als praxisbezogener Test für die neue Methode zur Auswertung von Beugungsscheiben, wurde ein experimentelles unter konvergenten Einstrahlbedingungen erstelltes Beugungsbild von Korund ($\alpha\text{-Al}_2\text{O}_3$) in $[00.1]$ in fünf Datensätze unterteilt. Dafür wurde ein Korundvollpräparat in $[00.1]$ -Richtung hergestellt (s. Kapitel 2.3.1). Nach der

Probenpräparation wurde der ionengedünnte Kristall für 8 s bei 1200 °C in einen Ofen gegeben. Der vorher durch das Ionendünnen sehr unebene Bereich am Probenrand des Korundeinkristalls hat sich durch das kurzzeitige Tempern geglättet. Gleichzeitig haben sich Plateaus, die durch Stufen voneinander getrennt sind, gebildet. Durch diese Behandlung erhält man eine Probe für die TEM, die sich sehr einfach untersuchen lässt. Die kurze Einwirkzeit bei 1200 °C reicht aus, weil die ionengedünnte Probe am Rand zum Probenloch hin mit ca. 100 Å dünn genug für eine Reorganisation des Kristallgitters ist, und die Oberfläche einem lokalen thermodynamischen Minimum zustreben kann [41].

Von dieser Probe wurde ein Beugungsbild in [00.1] mit der Slow-Scan-Kamera aufgenommen (s. Abbildung 25). Die Aufnahmezeit lag bei 2 s, was ein Kompromiss zwischen höherer Reflexintensität einerseits und einer geringen Strahlschädigung der Probe andererseits ist. Nach einer Belichtungszeit von 5 s war die Strahlschädigung der Probe so weit vorangeschritten, dass diese Aufnahmen nicht mehr verwertet werden konnten.

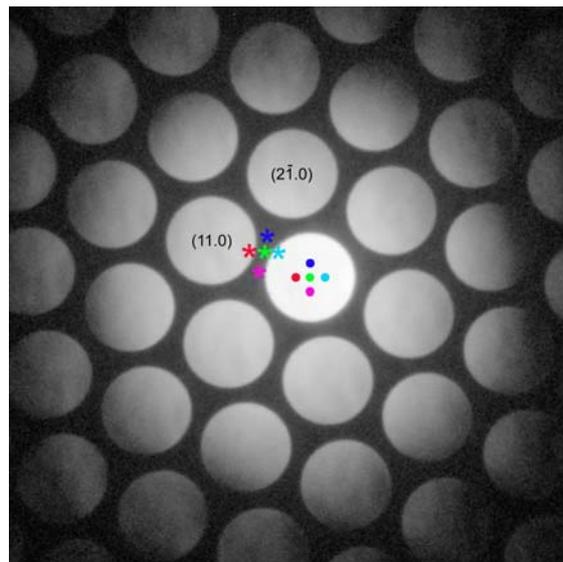


Abbildung 25: Experimentelles konvergentes Beugungsbild von Korund ($\alpha\text{-Al}_2\text{O}_3$) in [00.1]. Die farbigen Kreise stellen exemplarisch für alle Beugungsreflexe die ausgelesenen Datensätze im Primärstrahl dar. Die farbigen Sterne sind die verfeinerten Werte des Laugezentrum.

Der Durchmesser des ausgelesenen Bereiches betrug 20 Pixel.

Mit den fünf Datensätzen wurde analog zum simulierten Bild die x -Koordinate des Sauerstoffs verfeinert. Gleichzeitig wurde eine Verfeinerung der datensatzspezifischen Parameter (Skalenfaktor, Kristalldicke und Laugezentrum) durchgeführt. Die Ergebnisse sind in Tabelle 3 dargestellt. Als Startwert für die Verfeinerung wurde die x -Koordinate, analog der Vorgehensweise bei dem simulierten Beugungsbild (s. Kap. 3.1.1), auf $2/3$ gesetzt.

Wie aus den verfeinerten Werten in Tabelle 3 zu ersehen ist, sind die Skalenfaktoren und die Werte der Kristalldicken von allen fünf Datensätzen unabhängig voneinander zum selben Wert konvergiert. Die ausgelesenen Bereiche der einzelnen Datensätze spiegeln sich im Rahmen der berechneten Fehler in den Ergebnissen der Lauezentren wider und die x -Koordinate des Sauerstoffs ist zu dem Wert verfeinert worden, der in der Literatur von der Röntgenbeugung ebenfalls gefunden wird ($x = 0.6914$) [40]. Der R-Wert von 3,1 % über alle 138 Reflexe zeigt schließlich auch den Erfolg der gesamten Verfeinerung an.

Datensatz	Anzahl Reflexe	Lauezentrum		Proben- dicke [Å]	B [pm ²]	x/a Sauerstoff	Skalen- faktor	R- faktor
		h	k					
1 (nahe am Zentrum)	28	0.34(10)	0.74(13)	106(6)	O: 4 Al: 4	0.692(8)	0.87(4)	2.9 %
2 ($x + 30$ pixel)	27	0.19(8)	0.70(11)	110(6)			0.88(4)	2.6 %
3 ($x - 30$ pixel)	27	0.60(14)	0.80(16)	106(6)			0.87(3)	3.6 %
4 ($y + 30$ pixel)	28	0.81(13)	0.27(9)	108(6)			0.87(4)	3.1 %
5 ($y - 30$ pixel)	28	0.09(9)	0.95(17)	106(6)			0.87(4)	3.1 %
Gesamt R-Wert (138 Reflexe) = 3.1%								

Tabelle 3: Verfeinerungsergebnisse mit Intensitätsdaten aus einem konvergenten Beugungsbild von Korund (α -Al₂O₃) in [00.1].

3.2 Ergebnisse mit der neuen Methode

In den nächsten drei Kapiteln werden Strukturverfeinerungen von einer bereits bekannten (Ta_2P) und von zwei neuen Verbindungen ($\text{Sc}_3\text{Sc}_{0.32}\text{Al}_{0.68}\text{O}_6$ und ScGaO_3) beschrieben. Dabei werden jeweils zunächst die einzelnen Beugungsbilder mit den Ergebnissen der Vorverfeinerungen gezeigt. Anschließend wird für jede Verbindung das Resultat der Strukturverfeinerung mit abschließender Diskussion der erhaltenen Kristallstruktur dargelegt.

3.2.1 Strukturverfeinerung von Ta_2P

Mit dieser neu entwickelten Methode wurde die bereits seit 1966 bekannte Kristallstruktur von Ta_2P wieder verfeinert. Ta_2P kristallisiert in der orthorhombischen Raumgruppe $Pnmm$ (Nr. 58) mit den Gitterkonstanten $a = 11.5494(5) \text{ \AA}$, $b = 14.4204(6) \text{ \AA}$ und $c = 3.3997(2) \text{ \AA}$. Es befinden sich drei Formeleinheiten in der Elementarzelle. Sowohl die Tantalatome als auch die Phosphoratome befinden sich auf der 4g-Position $(x,y,0)$ in der Elementarzelle, die alle vollständig mit einer Atomsorte besetzt sind (s. Tabelle 4) [42]. *Weirich* et al. [43] hatten die Kristallstruktur mit Hilfe von Elektronenbeugungsdaten nochmals strukturell bestimmt; sie benutzten für die Strukturlösung und die sich anschließende Verfeinerung allerdings Programme für die Röntgenbeugung, die auf der kinematischen Theorie basieren. Der daraus resultierende hohe R-Wert von über 20 % ist eine Folge davon und konnte zur Güte der abschließenden Strukturverfeinerung nichts beitragen, da die kinematisch berechneten Intensitäten der Beugungsreflexe teilweise erheblich von denen der experimentellen Reflexintensitäten abwichen. Trotz dieser Voraussetzungen konnten die Tantalpositionen gefunden und verfeinert werden. Sie stimmten mit denen aus der Röntgenstrukturanalyse überein. Die fehlenden Phosphoratome konnten jedoch nicht aufgefunden werden, da sie von den Tantalatomen als sehr viel stärkere Elektronenstreuer überdeckt wurden.

Mit Hilfe von MSLS sollte die Ta_2P -Struktur mit Elektronenbeugungsdaten physikalisch korrekt verfeinert werden, was eine Verfeinerung der Phosphorpositionen einschließt. Gleichzeitig soll damit eine Strukturlösung nur mit Hilfe von Methoden der Elektronenkristallographie demonstriert werden.

Atom	x/a	y/a	z/c	Debye-Waller Faktor	Besetzungsfaktor	Wykhoff Pos.
Ta1	0.15290	0.02263	0	2	1	4g
Ta2	0.07860	0.25496	0	2	1	4g
Ta3	0.58638	0.07970	0	2	1	4g
Ta4	0.47001	0.39100	0	2	1	4g
Ta5	0.80325	0.20475	0	2	1	4g
Ta6	0.87746	0.42201	0	2	1	4g
P1	0.42180	0.20589	0	2	1	4g
P2	0.24340	0.41599	0	2	1	4g
P3	0.68396	0.34751	0	2	1	4g

Tabelle 4: Startwerte der Strukturverfeinerung von Ta₂P nach [43].

3.2.1.1 Die zur Strukturverfeinerung benutzen Zonenachsen

Für die Strukturverfeinerung wurden insgesamt zehn Datensätze aus unterschiedlichen Zonenachsen verwendet: Drei Datensätze in der Zonenachse [010], zwei Datensätze in der Zonenachse [110], drei Datensätze in der Zonenachse [112] und jeweils ein Datensatz in der Zonenachse [230] und [310]. Alle Beugungsdaten wurden mit der Multi-Scan-Kamera aufgezeichnet. Dabei wurde eine C2-Kondensorblende mit einem Durchmesser von 30 µm und eine spot size von 11 verwendet. Der Durchmesser des kreisförmigen auszulesenen Bereiches betrug 20 Pixel.

3.2.1.1.1 Zonenachse [010] von Ta₂P

In dieser Zonenachse wurden drei Datensätze mit unterschiedlicher Dicke zur Strukturverfeinerung herangezogen. Die drei Beugungsbilder in Abbildung 27 von Ta₂P in [010] wurden von dem gekennzeichneten Bereich des Kristalls in der Abbildung 26 aufgezeichnet.

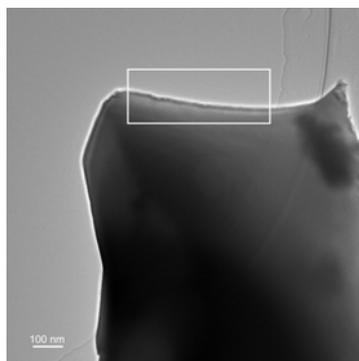
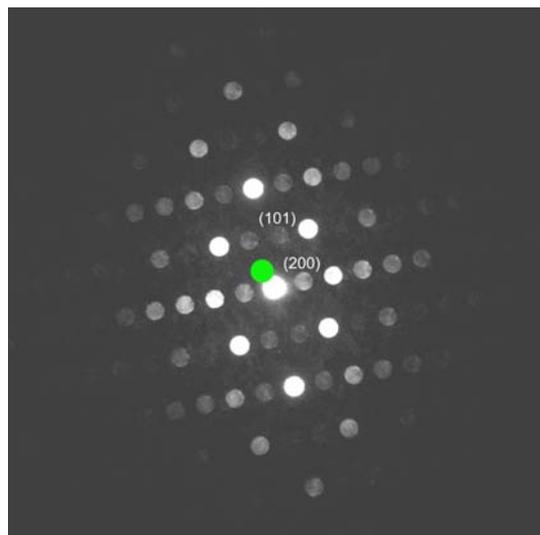
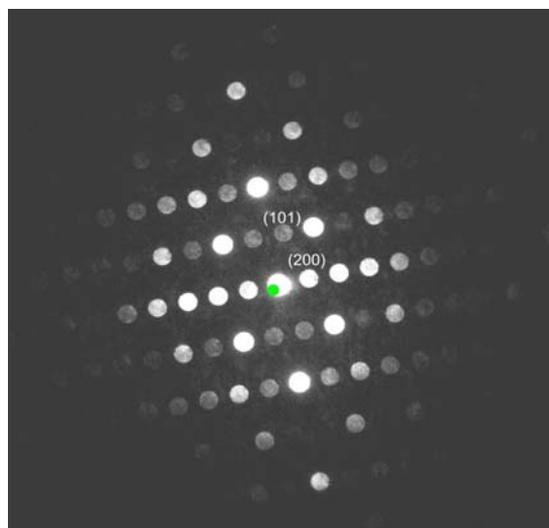


Abbildung 26: Hellfeldabbildung des untersuchten Ta₂P-Kristalls in [010]. Von diesem Kristall wurden auch die Aufnahmen der Zonenachsen [110], [230] und [310] erstellt. Alle Beugungsbilder wurden an der markierten Stelle aufgezeichnet.

a)



b)



c)

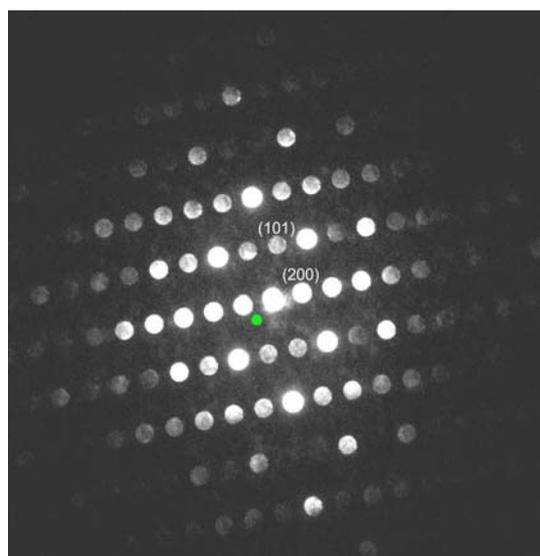


Abbildung 27: Drei Beugungsbilder von Ta_2P in $[010]$. Kameralänge: 780 mm. Grüner Kreis: Verfeinertes Lauezentrum des zentral ausgelesenen Datensatzes (Auslesekoordinaten x und $y = 0$). Größe des grünen Kreises = Fehler in der Bestimmung des Lauezentrums (s. Tabellen 5 bis 7). Aufnahmezeit: 2 s.

Daten- satz Nr.	Auslesekoor.		Anzahl Reflexe	Lauzentrum			Dicke [Å]	Skalen- faktor	R- Wert
	x	y		h	k	l			
1	0	0	171	-0.4(13)	0	0.24(4)	44(2)	0.912(6)	0.5 %
2	-10	0	192	-0.7(14)	0	0.28(5)	41(2)	0.914(6)	0.6 %
3	10	0	191	-0.3(11)	0	0.38(5)	47(2)	0.901(7)	0.6 %
4	0	-10	181	-0.3(11)	0	0.10(3)	45(2)	0.924(5)	0.4 %
5	0	10	200	-0.5(10)	0	0.36(4)	48(2)	0.917(6)	0.5 %
6	5	5	209	-0.3(10)	0	0.36(5)	47(2)	0.908(6)	0.6 %
7	-5	-5	210	-1.7(11)	0	0.15(4)	43(2)	0.920(5)	0.5 %

Daten- satz Nr.	Auslesekoor.		Anzahl Reflexe	Lauzentrum			Dicke [Å]	Skalen- faktor	R- Wert
	x	y		h	k	l			
1	0	0	199	-0.3(5)	0	0.057(10)	74(1)	0.921(7)	0.6 %
2	-10	0	195	0.2(6)	0	0.074(13)	69(1)	0.914(7)	0.6 %
3	10	0	200	-0.5(5)	0	0.098(11)	71(1)	0.907(6)	0.5 %
4	0	-10	210	-0.2(5)	0	-0.063(11)	71(1)	0.914(6)	0.5 %
5	0	10	155	-0.7(5)	0	0.142(13)	74(1)	0.923(8)	0.6 %
6	10	10	182	-1.3(5)	0	0.138(11)	75(1)	0.927(7)	0.5 %
7	-10	-10	179	-0.6(6)	0	-0.044(13)	69(1)	0.916(7)	0.6 %

Daten- satz Nr.	Auslesekoor.		Anzahl Reflexe	Lauzentrum			Dicke [Å]	Skalen- faktor	R- Wert
	x	y		h	k	l			
1	0	0	226	-1.1(2)	0	-0.142(7)	108(1)	0.917(8)	1.0 %
2	-10	0	173	-1.0(3)	0	-0.142(8)	109(1)	0.939(10)	1.3 %
3	10	0	186	-1.4(3)	0	-0.120(7)	107(1)	0.915(9)	1.1 %
4	0	-10	194	-1.5(3)	0	-0.206(8)	109(1)	0.922(10)	1.1 %
5	0	10	218	-0.8(2)	0	-0.069(7)	108(1)	0.916(8)	1.1 %
6	10	10	185	-1.2(2)	0	-0.031(7)	108(1)	0.925(8)	1.0 %
7	-10	-10	197	-0.8(2)	0	-0.212(7)	109(1)	0.938(8)	1.0 %

Tabellen 5, 6 und 7: Berechnete Werte nach der Vorverfeinerung mit Intensitätsdaten in der [010]-Zonenachsen von Ta₂P. Tabelle 5 gehört zur Abbildung 27a, Tabelle 6 zur Abbildung 27b und Tabelle 7 zur Abbildung 27c.

Die Verwendbarkeit der zentral ausgelesenen Datensätze für die Strukturverfeinerung zeigen die verfeinerten Werte der Tabellen 5 bis 7 an. Aus den Daten können folgende Erkenntnisse gewonnen werden:

Für Aufnahmen von dünneren Probenbereichen ist es charakteristisch, dass die Reproduzierbarkeit der Einstrahlrichtungen in den Ergebnissen der Lauzentren nicht gegeben ist. Die dazugehörigen Fehler sind größer als der Abstand in Pixel der einzelnen Datensätze in dem konvergenten Beugungsbild (s. Tabelle 5). Die

berechneten Fehler der Werte der Lauezentren sind zwar in der Tabelle 6 immer noch größer als der Abstand in Pixel zwischen den einzelnen Datensätzen aber die Tendenz, die unterschiedlichen Einstrahlrichtungen aus den verfeinerten Datensätzen herauszulesen, ist ablesbar. Erkennen lässt sich dieses z. B. an den h -Werten der Lauezentren der Datensätze 1, 2 und 3. Von $h = 0.2(6)$ im zweiten Datensatz in der Tabelle 6 über $h = -0.3(5)$ im zentral ausgelesenen Datensatz 1 geht der Wert schließlich zu $h = -0.5(5)$ im dritten Datensatz. An dieser kontinuierlichen Abnahme des h -Wertes, dessen Wert sich gemäß der Indizierung des Beugungsbildes auch in diese Richtung so ändern muss (s. Abbildung 27b), lässt sich die konsistente Reproduktion der Einstrahlrichtungen in den verfeinerten Werten der Lauezentren erkennen. Betrachtet man die Datensätze Nr. 1, Nr. 6 und Nr. 7 in Tabelle 7 so lässt sich erkennen, dass sowohl die h -Werte als auch die l -Werte in Übereinstimmung mit ihren relativen Lagen zueinander ihre ausgelesenen Position widerspiegeln. Das zugrundeliegende Beugungsbild der Tabelle 7 wurde an einem Bereich mit der in dieser Zonenachse größten Kristalldicke von $108(1) \text{ \AA}$ erstellt.

In den Tabellen 5 bis 7 sind die verfeinerten Werte der Probendicke und die der Skalenfaktoren zu den selben Werten konvergiert. Damit sind diese Kriterien für die Verwendung der zentral ausgelesenen Datensätze zur Strukturverfeinerung gleichermaßen erfüllt.

3.2.1.1.2 Zonenachsen [110], [310] und [230] von Ta_2P

In der Zonenachse [110] wurden zwei Datensätze mit unterschiedlichen Kristalldicken für die Strukturverfeinerung von Ta_2P und in den übrigen beiden Zonenachsen [310] und [230] jeweils ein Datensatz herangezogen.

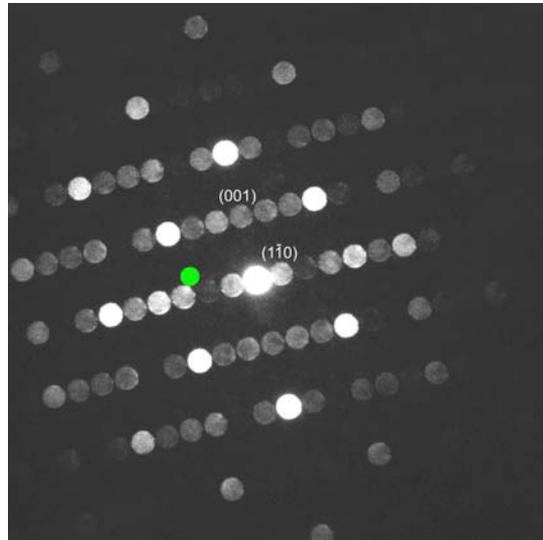
Die intensitätsstarken Reflexe am Rande des Beugungsbildes der Abbildung 28c sind Reflexe der ersten Laue Zone und werden nicht mit in die Verfeinerung eingeschlossen. Trotz der in den Tabellen 9 und 10 gezeigten relativ großen R-Werte von rund 6 % bzw. 7 %, ist jeweils der zentrale Datensatz zur Strukturverfeinerung herangezogen worden, da sowohl die verfeinerten Werte der Kristalldicke als auch die der Skalenfaktoren aller Datensätze einer Tabelle zu den selben Werten konvergiert sind. Zudem werden die unterschiedlichen Einstrahlrichtungen der verschiedenen Datensätze in Tabelle 9 in den verfeinerten Lauezentren reproduziert. Durch die größere Anzahl an Entscheidungskriterien für die Verwendbarkeit eines Datensatzes zur Strukturverfeinerung als bei der ursprünglichen Methode von *Jansen et al.* [37], ist es leichter möglich sich für die Benutzung eines Datensatzes zu entscheiden.

Die Reproduktion der verschiedenen Einstrahlrichtungen ist bei den verfeinerten Werten der Lauezentren in den Tabellen 10 und 11 nicht möglich. Für eine genauere Bestimmung des Lauezentrums ist eine größere Anzahl intensitätsstarker Reflexe erforderlich. In dieser Orientierung und bei dieser Probendicke ist das aber nicht

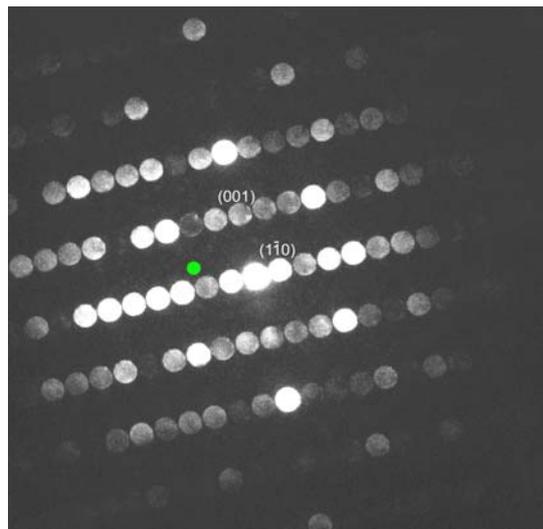
realisierbar. Die Skalenfaktoren und die Probendicken wurden zu den selben Werten verfeinert.

Der verwendete Kristall ist in Abbildung 26 dargestellt.

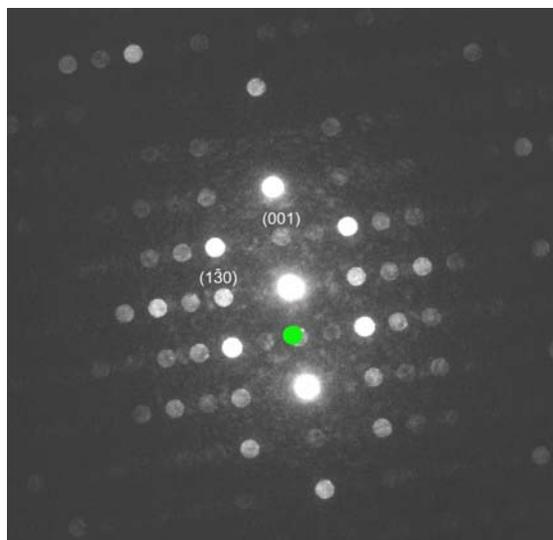
a)



b)



c)



d)

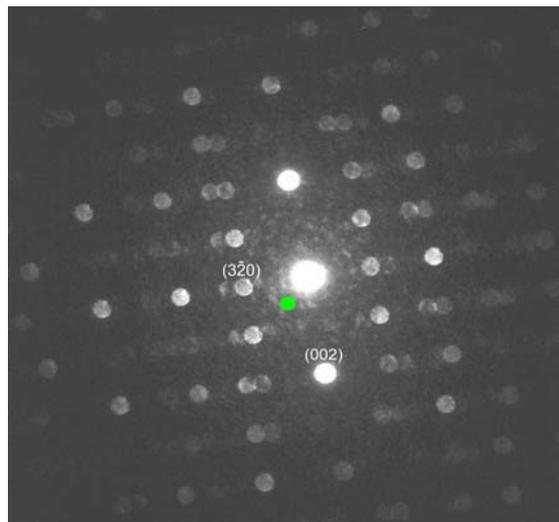


Abbildung 28: Beugungsbilder von Ta₂P in [110] (a und b), in [310] (c) und in [230] (d). Kameralänge: 980 mm (a und b) bzw. 780 mm (c und d). Grüner Kreis: Verfeinertes Lauezentrum des zentral ausgelesenen Datensatzes (Auslesekoordinaten x und $y = 0$). Größe des grünen Kreises = Fehler in der Bestimmung des Lauezentrums (s. Tabellen 8 bis 11). Aufnahmezeit: 2 s.

Datensatz Nr.	Auslesekoor.		Anzahl Reflexe	Lauezentrum			Dicke [Å]	Skalen- faktor	R- Wert
	x	y		h	k	l			
1	0	0	104	-2.5(4)	2.5(4)	0.20(3)	67(2)	0.974(18)	1.6 %
2	-10	0	118	-1.9(3)	1.9(3)	0.18(3)	69(2)	0.982(15)	1.4 %
3	10	0	106	-3.2(4)	3.2(4)	0.21(4)	67(2)	0.968(19)	1.8 %
4	0	-10	110	-2.2(3)	2.2(3)	0.17(3)	67(2)	0.967(18)	1.8 %
5	0	10	116	-2.3(3)	2.38(3)	0.29(3)	67(2)	0.983(17)	1.5 %
6	10	10	105	-2.2(3)	2.2(3)	0.22(3)	69(2)	0.983(18)	1.7 %
7	-10	-10	116	-1.7(3)	1.7(3)	0.14(3)	68(2)	0.979(17)	1.6 %

Datensatz Nr.	Auslesekoor.		Anzahl Reflexe	Lauezentrum			Dicke [Å]	Skalen- faktor	R- Wert
	x	y		h	k	l			
1	0	0	115	-2.2(3)	2.2(3)	0.36(3)	98(2)	0.92(2)	5.8 %
2	-10	0	108	-1.7(3)	1.7(3)	0.35(3)	98(2)	0.91(3)	6.1 %
3	10	0	106	-3.0(3)	3.0(3)	0.35(3)	101(3)	0.93(3)	5.5 %
4	0	-10	109	-2.7(3)	2.7(3)	0.31(3)	98(3)	0.91(3)	6.4 %
5	0	10	113	-2.0(3)	2.0(3)	0.42(3)	99(2)	0.92(3)	6.0 %
6	10	10	116	-2.6(3)	2.6(3)	0.43(3)	99(2)	0.93(3)	6.0 %
7	-10	-10	106	-1.8(3)	1.8(3)	0.30(4)	95(2)	0.91(3)	6.8 %

Datensatz Nr.	Auslesekoord.		Anzahl Reflexe	Lauzentrum			Dicke [Å]	Skalen faktor	R- Wert
	x	y		h	k	l			
1	0	0	107	0.0(6)	-0.1(19)	-1.0(4)	57(10)	0.86(2)	6.5 %
2	-10	0	89	0.0(6)	0.0(19)	-1.1(4)	58(11)	0.89(3)	6.1 %
3	10	0	109	0.1(6)	-0.2(19)	-1.0(4)	57(11)	0.85(2)	7.2 %
4	0	-10	99	0.1(6)	-0.2(18)	-1.0(5)	62(11)	0.88(3)	6.6 %
5	0	10	111	0.0(7)	0.(2)	-1.0(5)	55(11)	0.85(2)	7.1 %
6	10	10	90	-0.2(8)	1.(2)	-1.0(5)	54(13)	0.84(3)	7.4 %
7	-10	-10	98	0.1(6)	0.1(19)	-1.0(5)	56(12)	0.86(3)	7.0 %

Datensatz Nr.	Auslesekoord.		Anz. Refl.	Lauzentrum			Dicke [Å]	Skalen- faktor	R- Wert
	x	y		h	k	l			
1	0	0	109	1.(2)	-0.4(15)	0.5(2)	42(9)	0.825(12)	1.8 %
2	-10	0	104	0.1(18)	0.0(12)	0.34(17)	45(8)	0.810(14)	1.7 %
3	10	0	107	0.0(17)	0.0(11)	0.37(17)	45(9)	0.821(14)	1.9 %
4	0	-10	99	0.8(19)	-0.5(13)	0.23(11)	44(9)	0.839(13)	1.5 %
5	0	10	110	-0.2(18)	0.1(12)	0.35(16)	46(7)	0.811(14)	1.8 %
6	5	5	109	-0.3(18)	0.2(12)	0.28(12)	46(7)	0.832(15)	1.7 %
7	-5	-5	115	0.2(16)	-0.1(11)	0.21(9)	45(8)	0.827(13)	1.6 %

Tabellen 8, 9, 10 und 11: Berechnete Werte nach der Vorverfeinerung mit Intensitätsdaten in der [110] (Tabellen 8 und 9), in der [310] (Tabelle 10) und in der [230]-Zonenachse (Tabelle 11) von Ta₂P. Tabelle 8 gehört zur Abbildung 28a, Tabelle 9 gehört zur Abbildung 28b, Tabelle 10 gehört zur Abbildung 28c und Tabelle 11 gehört zur Abbildung 28d.

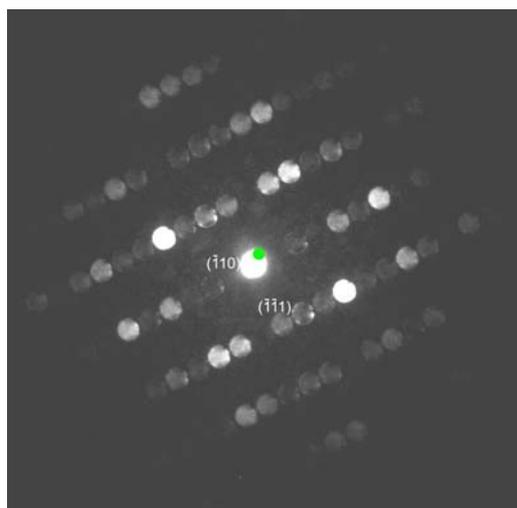
3.2.1.1.3 Zonenachse [112] von Ta₂P

In dieser Zonenachse wurden drei Datensätze zur Strukturverfeinerung herangezogen. Die Beugungsbilder in der Zonenachse [112] sind von dem in der Abbildung 30 mit weißen Linien und dem Pfeil markierten Bereiche aufgezeichnet worden.

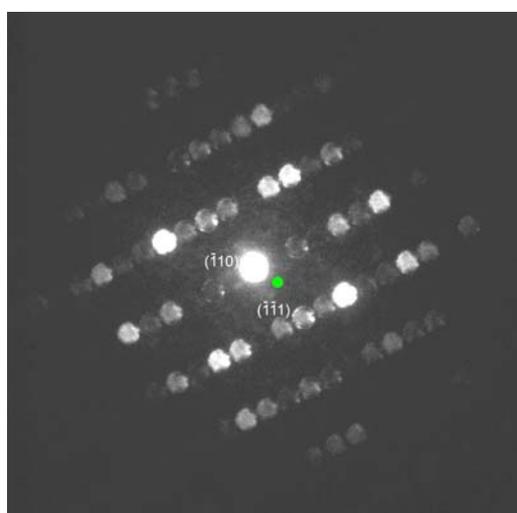
Der Datensatz Nr. 4 in Tabelle 12 stellt eine Ausnahme dar. Die verfeinerte Kristalldicke ist mit 62(4) kleiner als die übrigen Werte der verfeinerten Probendicken in dieser Tabelle. Auch die Fehler bei den verfeinerten Daten des Lauzentrums sind im Vergleich zu den übrigen Ergebnissen der anderen Datensätze in Tabelle 12 etwas größer geworden. Offensichtlich wurden die Intensitäten im unteren Bereich der Beugungsscheiben infolge einer Aufladung der C2-Kondensorblende verfälscht.

Die verfeinerten Daten der Lauzentren in den Tabellen 12 und 14 reproduzieren die ausgelesenen Einstrahlrichtungen. Diese Werte in Tabelle 13 lassen nur eine richtige Tendenz erkennen, da die berechneten Fehler größer sind als der Abstand der einzelnen Datensätze innerhalb des Beugungsbildes.

a)



b)



c)

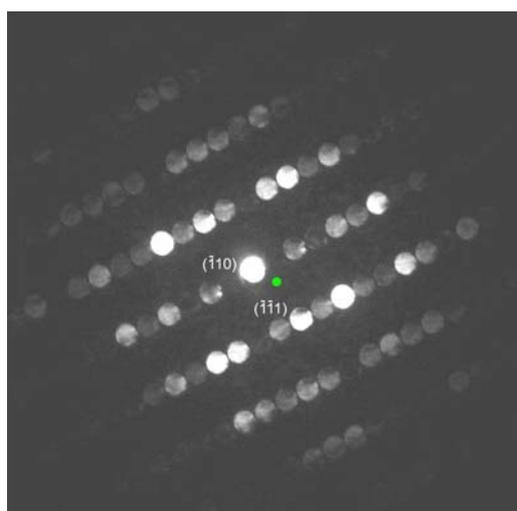


Abbildung 29: Drei Beugungsbilder von Ta₂P in [112]. Kameralänge: 980 mm. Grüner Kreis: Verfeinertes Lauezentrum des zentral ausgelesenen Datensatzes (Auslesekoordinaten x und y = 0) an. Größe des grünen Kreises = Fehler in der Bestimmung des Lauezentrums (s. Tabellen 12 bis 14). Aufnahmezeit: 2 s.

Datensatz Nr.	Auslesekoord.		Anz. Refl.	Lauzentrum			Dicke [Å]	Skalen- faktor	R- Wert
	x	y		h	k	l			
1	0	0	104	0.58(19)	0.11(19)	-0.34(7)	69(4)	0.846(12)	1.6 %
2	-10	0	96	0.82(17)	-0.27(17)	-0.27(6)	69(4)	0.871(11)	1.1 %
3	10	0	104	0.28(19)	0.57(19)	-0.43(7)	73(4)	0.844(12)	1.8 %
4	0	-10	106	0.7(2)	-0.1(2)	-0.27(9)	62(4)	0.853(11)	1.6 %
5	0	10	102	0.76(17)	0.02(17)	-0.39(7)	70(3)	0.859(12)	1.4 %
6	10	10	98	0.59(17)	0.25(17)	-0.42(6)	75(4)	0.859(11)	1.5 %
7	-10	-10	99	0.84(19)	-0.46(19)	-0.19(6)	67(4)	0.872(12)	1.3 %

Datensatz Nr.	Auslesekoord.		Anz. Refl.	Lauzentrum			Dicke [Å]	Skalen- faktor	R- Wert
	x	y		h	k	l			
1	0	0	98	0.44(12)	-0.87(12)	0.21(4)	94(3)	0.864(12)	1.4 %
2	-10	0	101	0.46(12)	-0.93(12)	0.24(4)	94(3)	0.859(11)	1.6 %
3	10	0	102	0.40(10)	-0.73(10)	0.16(4)	96(3)	0.863(10)	1.2 %
4	0	-10	96	0.43(15)	-0.97(15)	0.27(5)	92(3)	0.856(11)	1.5 %
5	0	10	102	0.49(11)	-0.74(11)	0.12(4)	95(3)	0.865(10)	1.3 %
6	10	10	103	0.42(9)	-0.56(9)	0.07(3)	95(3)	0.862(9)	1.1 %
7	-10	-10	108	0.37(14)	-0.93(14)	0.28(5)	94(4)	0.826(12)	2.1 %

Datensatz Nr.	Auslesekoord.		Anz. Refl.	Lauzentrum			Dicke [Å]	Skalen- faktor	R- Wert
	x	y		h	k	l			
1	0	0	106	0.19(7)	-0.61(7)	0.21(3)	122(2)	0.876(11)	1.5 %
2	-10	0	96	0.31(9)	-0.75(9)	0.22(3)	126(3)	0.865(12)	1.7 %
3	10	0	101	0.11(6)	-0.53(6)	0.21(2)	131(2)	0.882(11)	1.5 %
4	0	-10	101	0.07(8)	-0.70(8)	0.31(3)	121(3)	0.888(11)	1.3 %
5	0	10	95	0.34(7)	-0.72(7)	0.19(3)	127(2)	0.869(11)	1.5 %
6	10	10	94	0.23(7)	-0.60(7)	0.19(3)	136(3)	0.869(12)	1.7 %
7	-10	-10	109	0.43(7)	-0.76(7)	0.16(3)	124(3)	0.857(12)	1.7 %

Tabellen 12, 13 und 14: Berechnete Werte nach der Vorverfeinerung mit Intensitätsdaten in der [112]-Zonenachse von Ta₂P. Tabelle 12 gehört zur Abbildung 29a, Tabelle 13 zur Abbildung 29b und Tabelle 14 zur Abbildung 29c.

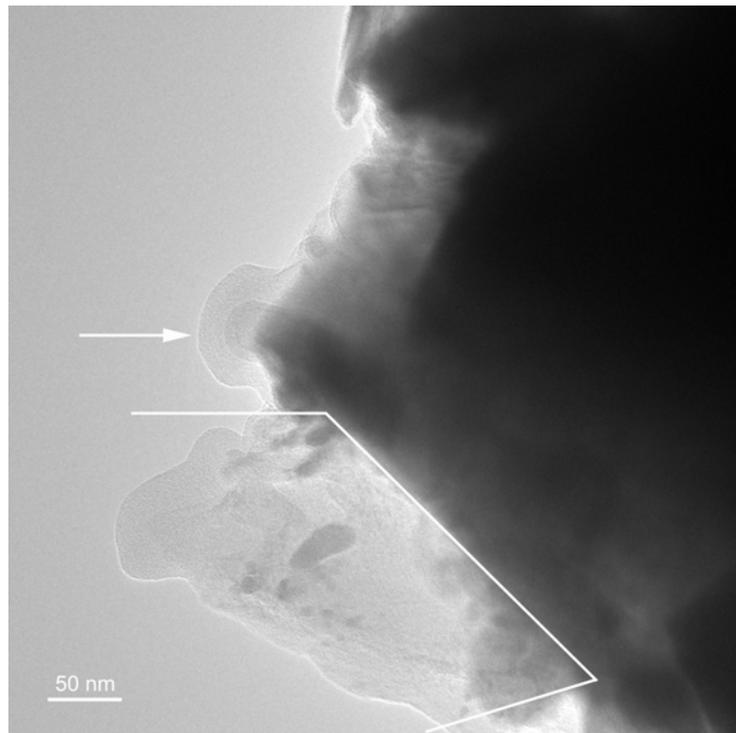


Abbildung 30: Hellfeldabbildung des untersuchten Ta₂P-Kristalls in [112].

3.2.2 Strukturverfeinerung von Ta₂P

Alle zentral ausgelesenen Datensätze der eben beschriebenen Beugungsbilder wurden zur Strukturverfeinerung von Ta₂P verwendet. Da nur unter konvergenten Einstrahlbedingungen erstellte Beugungsbilder zur Strukturverfeinerung verwendet wurden, ist aufgrund fehlender Intensitätsdaten z.B. in der [001]-Richtung die Vollständigkeit der möglichen Datensätze nicht gegeben. Ta₂P wurde im Rahmen dieser Arbeit als Testsubstanz für das Auswerten von unter konvergenten Einstrahlbedingungen aufgenommenen Beugungsbilder eingesetzt. Beugungsreflexe vom Typ (0k0), die mit konvergenten Elektronenstrahl erstellt werden, können mit dieser Methode nicht ausgewertet werden, da sich diese Beugungsscheibchen infolge der langen *b*-Achse überlappen. Diese Reflexe treten in der [100]- und der [001]-Zonenachse auf. Die erhaltenen Ergebnisse (s. Tabelle 15 und Tabelle 16) lassen dennoch einige interessante und in der Literatur noch nicht diskutierten Aspekte der hier verfeinerten Ta₂P-Struktur erkennen.

Atom	<i>x/a</i>	<i>y/a</i>	<i>z/c</i>	Debye-Waller Faktor	Besetzungsfaktor	Wyckhoff Pos.
Ta1	0.1525(17)	0.021(2)	0	1.97(19)	1	4g
Ta2	0.0781(15)	0.257(2)	0	1.9(2)	1	4g
Ta3	0.5871(20)	0.082(2)	0	1.98(19)	1	4g
Ta4	0.4702(17)	0.3899(18)	0	1.85(20)	1	4g
Ta5	0.8035(14)	0.205(2)	0	1.9(3)	1	4g
Ta6	0.8778(15)	0.4226(18)	0	1.9(3)	1	4g
P1	0.423(4)	0.206(6)	0	1.6(7)	1	4g
P2	0.244(5)	0.415(7)	0	1.9(7)	1	4g
P3	0.685(3)	0.348(6)	0	2.3(9)	1	4g

Tabelle 15: Ergebnisse der Strukturverfeinerung von Ta₂P.

Zonenachse	Lauezentrum			Dicke [Å]	Skalenfaktor	R-Faktor [%]
	<i>h</i>	<i>k</i>	<i>l</i>			
[010]	6.(2)	0	0.27(13)	44(8)	0.917(18)	0.3
[010]	-0.3(12)	0	0.06(3)	75(4)	0.92(2)	0.5
[010]	-1.1(6)	0	-0.141(18)	110(3)	0.91(2)	0.8
[110]	-2.4(5)	2.4(5)	0.19(4)	69(4)	0.98(3)	1.3
[110]	-2.1(3)	2.1(3)	0.34(3)	101(2)	0.94(2)	3.8
[112]	0.6(2)	0.1(2)	-0.35(9)	70(6)	0.791(15)	1.5
[112]	0.44(17)	-0.88(17)	0.22(6)	94(5)	0.873(18)	1.2
[112]	0.19(10)	-0.62(10)	0.21(4)	122(4)	0.889(19)	1.2
[310]	0.0(2)	-0.1(7)	-1.04(16)	59(4)	0.857(15)	6.2
[230]	11.0(19)	-7.3(13)	0.46(12)	42(6)	0.791(15)	1.5
Absorption = 0.000023(6)						
Gesamt R-Wert 1356 Reflexe = 2.2 %						

Tabelle 16: Berechnete datensatzspezifische Werte nach der Strukturverfeinerung von den benutzten Zonenachsen von Ta₂P.

Aufgrund der Anwesenheit von Tantal, dass das Auftreten von Absorptionsphänomenen wahrscheinlich macht, wurde bei der Verfeinerung ein Absorptionsparameter mit verfeinert.

3.2.2.1 Diskussion der Strukturverfeinerung an Ta₂P

Ta₂P ist keine ionische Verbindung im allgemeingültigen Sinn. Zum Teil sind kubisch innenzentrierte Tantalcluster in das Anionennetzwerk eingebaut, welche Rückschlüsse auf eine Tantal-Tantal-Wechselwirkung zulassen.

Atom Nr. 1	Atom Nr. 2	Abstand [Å]
Ta1	P2	2.577(68)
	P3	2.566(67)
	Ta1	3.572(28)
	Ta2	3.507(40)
	Ta3	3.352(33)
	Ta4	2.908(29)
	Ta4	2.993(26)
	Ta6	3.210(23)
Ta2	P1	2.524(40)
	P2	2.975(89)
	P3	2.587(56)
	Ta2	3.396
	Ta3	2.876(33)
	Ta4	2.986(30)
	Ta5	3.155(21)
	Ta6	3.322(33)
Ta3	P1	2.604(73)
	P2	2.483(45)
	Ta2	2.876(33)
	Ta3	3.103(38)
	Ta5	3.063(33)
	Ta6	2.885(31)
	Ta6	2.954(24)
Ta4	P1	2.705(89)
	P2	2.636(62)
	P3	2.552(44)
	Ta4	3.396
	Ta5	2.908(25)
Ta5	P1	2.536(53)
	P2	2.519(74)
	P3	2.473(79)
	Ta5	3.396
	Ta6	3.250(38)
Ta6	Ta6	3.596(30)
P1	P1	3.396
	P2	3.322(56)
P2	P2	3.396
P3	P3	3.396

Tabelle 17: Gefundene Atomabstände im Ta₂P.

In einigen Fällen ist in Tabelle 17 kein Fehler bei den Atomabständen angegeben. Der Grund dafür liegt darin, dass der Unterschied dieser Atompaare nur in der z-Koordinate besteht, die auf der 4g-Position in der Raumgruppe *Pnmm* (Nr. 58) den festgelegten Wert von Null annimmt und damit nicht verfeinerbar ist. Einige Tantal-Tantal-Bindungsabstände sind kleiner als 3 Å, weswegen eine Wechselwirkung zwischen diesen Atomen gegeben ist (Atomabstand von Tantal im Metall = 2.92 Å [42]). Daher ist es gerechtfertigt, diese Struktur als eine aus Tantalclustern aufgebaute Struktur anzusehen.

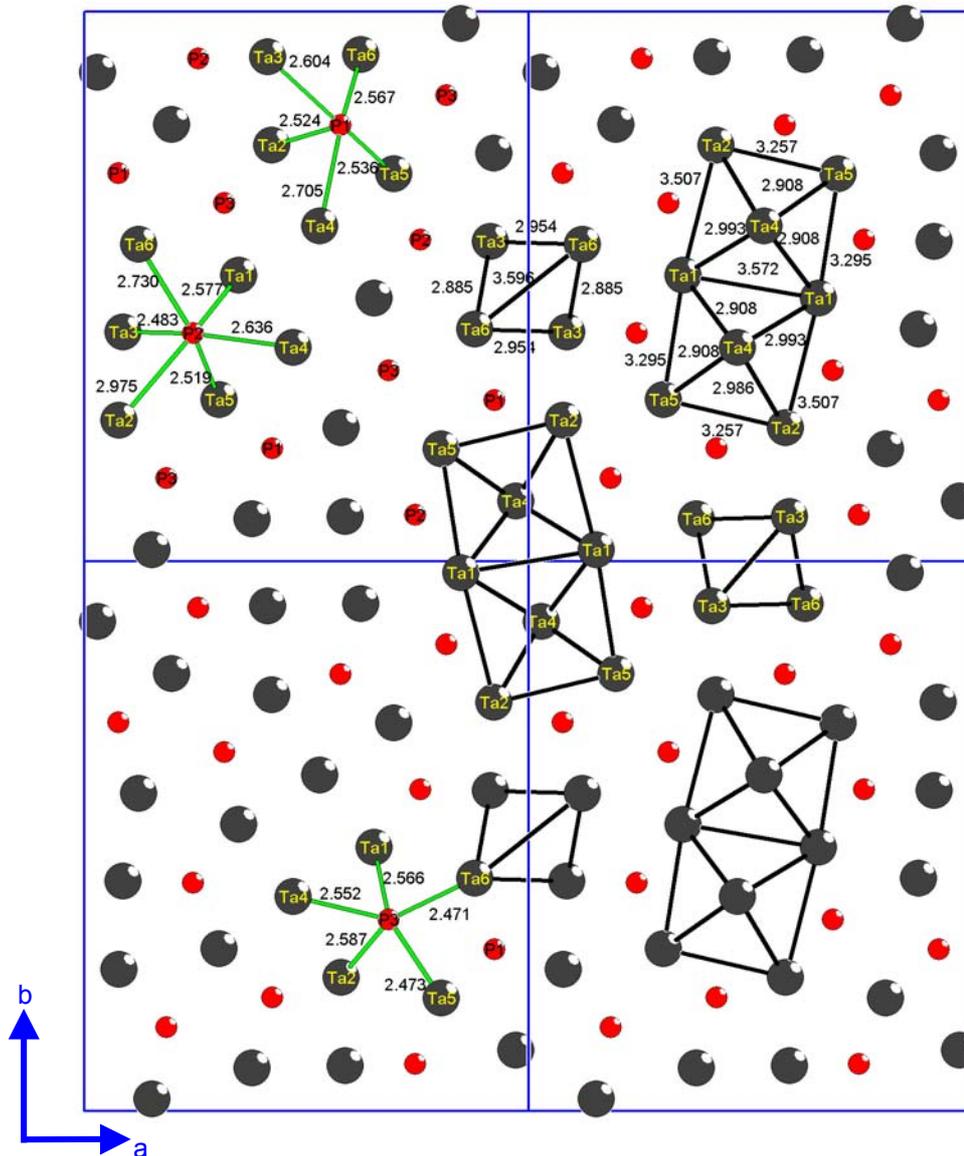


Abbildung 31: Tantal-Cluster nach [44] und die Phosphor-Koordination im Ta₂P in [001] mit ausgewählten Bindungsabständen. Dargestellt sind insgesamt vier Elementarzellen. Die blaue Linie kennzeichnet die Elementarzellenbegrenzung.

Die in Abbildung 31 in Projektion eingezeichneten isolierten - und auch kantenverknüpfte Ta₆-Oktaeder wurden von *Simon* postuliert [44]. Die hier gefundenen

Bindungsabstände weisen allerdings nicht auf die von *Simon* postulierten Oktaeder hin sondern auf innenzentrierte Ta_9 -Würfel. Innerhalb der „Oktaeder“ liegen Bindungsabstände von z.B. $3.507(40)$ Å ($Ta1-Ta2$) vor, die auch von *Simon* erwähnt werden [44]. Für eine Wechselwirkung sind Tantal-Tantalatom-Abstände von kleiner 3 Å nötig. Diese Abstände konnten hier innerhalb von innenzentrierten Ta_9 -Würfeln gefunden werden (rote Bindungen in Abbildung 32). Auch die kubisch-innenzentrierte Elementarstruktur von Tantal [45] rechtfertigt die Annahme eines innenzentrierten Ta_9 -Würfels innerhalb einer elektronenreichen Tantalverbindung. Die in der Abbildung 32 mit roten Bindungen gekennzeichneten innenzentrierten Ta_9 -Würfeln sind entlang der c -Achse flächenverknüpft. Dabei bilden sie einen Doppelstrang, der ebenfalls durch eine Flächenverknüpfung entstanden ist (s. Abbildung 32 und Abbildung 34).

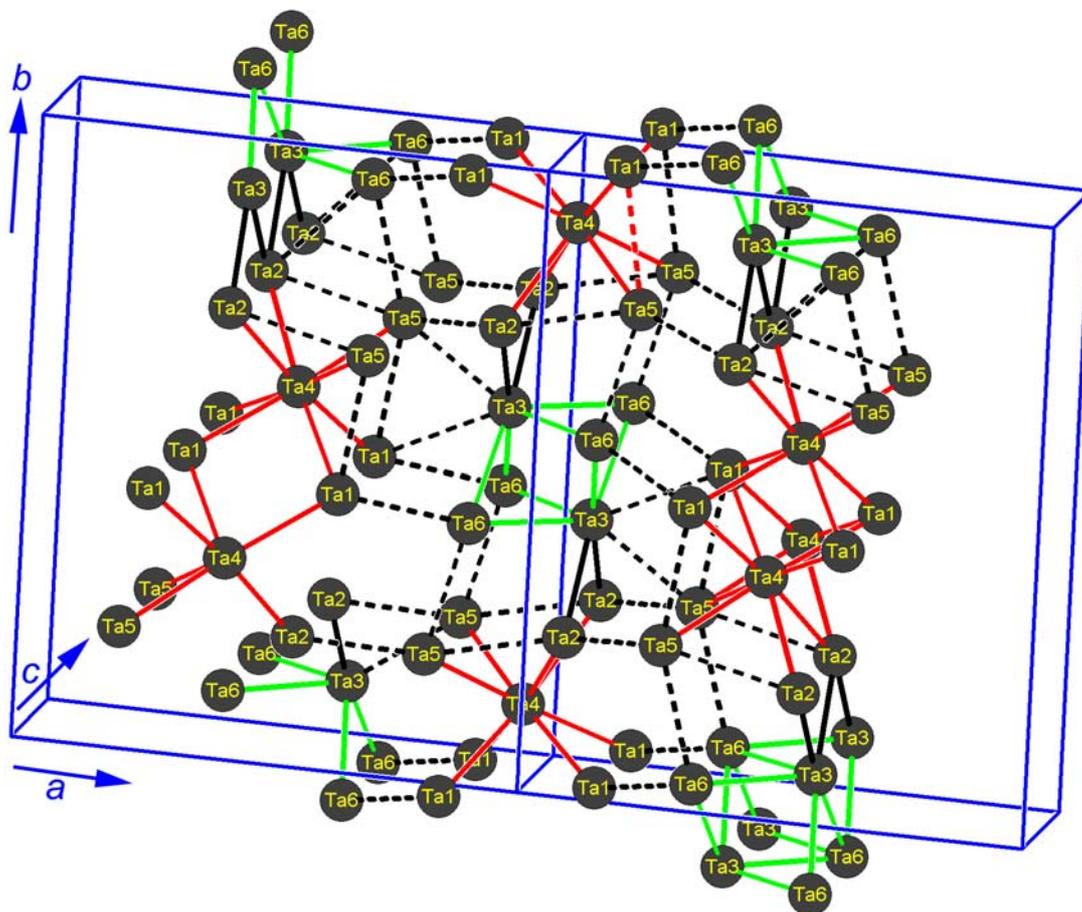


Abbildung 32: Zwei Elementarzellen des Ta_2P . Die blaue Linie kennzeichnet die Elementarzellenbegrenzung. Dargestellt ist das Ta-Untergitter im Ta_2P . Alle durchgezogenen Linien markieren Ta-Ta-Abstände kleiner 3 Å, punktierte Linien kennzeichnen Ta-Ta-Abstände zwischen 3 - und 3.5 Å. Nicht eingezeichnet sind die durch die Symmetrie erzeugten $Ta(X)-Ta(X)$ -Abstände, die immer einen Wert von 3.396 Å aufweisen.

Die in Abbildung 32 grün gekennzeichneten Ta_6 -Cluster besitzen innerhalb der von *Simon* postulierten oktaedrischen Tantalcluster ebenfalls mit $3.596(39)$ Å und 3.396 Å

(Ta₆-Ta₆-Abstände) zu große Bindungsabstände für eine Tantal-Tantal-Wechselwirkung (s. Abbildung 32). Um das Ta₃-Atom befinden sich neben vier Ta₆-Atomen (grüne Verbindungslinie in Abbildung 32) auch noch zwei weitere Ta₂-Atome mit einem Abstand von 2.876(33) Å (schwarze Verbindungslinie in Abbildung 32). Dieser Bindungsabstand lässt eine Tantal-Tantal-Wechselwirkung zu. Die sechs nahen Tantalatome sind in den Ecken eines innenzentrierten Würfels platziert. Eine der beiden noch freien Koordinationsstellen wird durch das Ta₅ (Abstand zum Ta₃ = 3.063(33) Å) besetzt (unterbrochene schwarze Verbindungslinie in Abbildung 32). Die noch fehlende Stelle wird von Ta₁ im Abstand von 3.352(33) Å belegt (unterbrochene schwarze Verbindungslinie in Abbildung 32). Durch das Ta₁-Atom ist der resultierende innenzentrierte Ta₉-Würfel stark verzerrt (s. Abbildung 32). Bei diesem stark verzerrten innenzentrierten Würfel müssen demnach die Phosphoratome eine Rolle spielen, da zwei Bindungsabstände oberhalb 3 Å liegen. Dazu wird zunächst die Umgebung der angesprochenen Tantal-Cluster mit den Phosphoratomen beschrieben.

Bei beiden in Abbildung 32 grün gekennzeichneten Tantal-Clustern ist eine Phosphorabschirmung vorhanden. Die Ta₃-Atome sind jeweils von einer trigonalen Fläche von ihrer Umgebung abgeschirmt. Hingegen sind die Ta₆-Atome von einer trigonalen Pyramide aus Phosphoratomen umgeben. Dabei befindet sich das Ta₆-Atom nicht genau im Zentrum dieser Pyramide sondern es liegt leicht unterhalb der trigonalen Basisfläche (s. Abbildung 33).

Auch die in Abbildung 32 rot gekennzeichneten Würfel sind von allen Seiten von Phosphoratomen umschlossen. Die Ta₁-Atome sind dabei von einer rechteckigen Fläche und die Ta₂- bzw. die Ta₅-Atome, die den Abschluss des Doppelstranges bilden, von einer rechtwinkligen Pyramide aus Phosphoratomen umgeben. Dabei befinden sich die Ta₂- bzw. die Ta₅-Ionen leicht oberhalb der rechtwinkligen Basisfläche der Pyramide (s. Abbildung 34).

Die trigonale Fläche aus Phosphoratomen (s. Abbildung 33) liegt genau zwischen dem Ta₃-Atom und dem Ta₂-Atom, die mit dem kurzen Abstand von 2.876(33) Å voneinander entfernt sind. Die rechtwinklige Fläche aus Phosphoratomen befindet sich zwischen den mit 3.210(23) Å voneinander entfernt liegenden Ta₁- und dem Ta₆-Atom. Durch die Anwesenheit von vier nächsten Phosphoratomen kann der größere Ta₁-Ta₆-Abstand im Vergleich zu den von drei Phosphoratomen koordinierten Ta₃- bzw. Ta₂-Atomen in der Struktur realisiert werden. Die große Verzerrung des z.T. in der Abbildung 32 grün gekennzeichneten innenzentrierten Würfels lässt sich damit auf die Abschirmung von den Phosphoratomen zurückführen.

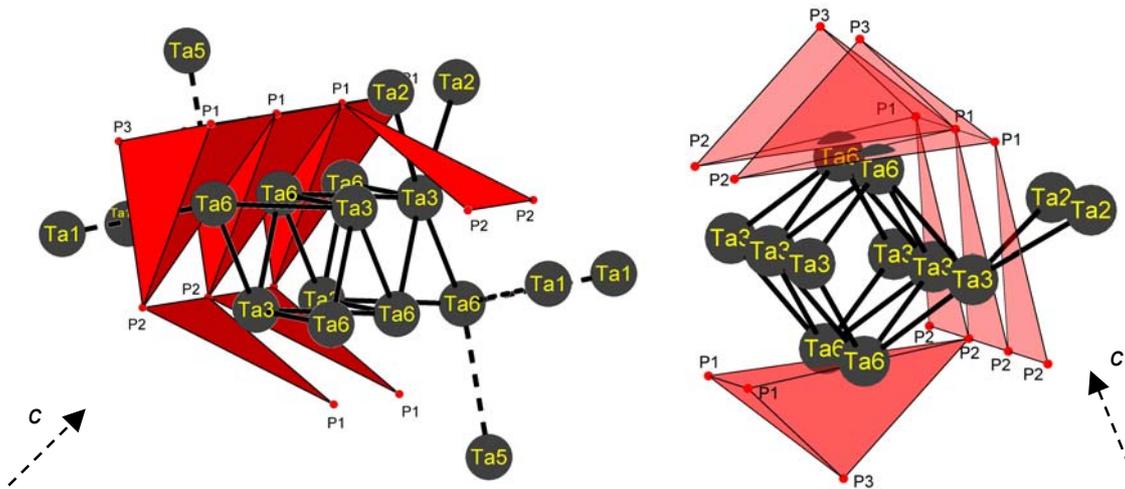


Abbildung 33: Phosphorpolyeder um links dem Ta₆-Oktaeder mit der Basisfläche aus Ta₆-Ionen und rechts mit der Basisfläche aus Ta₃-Ionen. Die *c*-Achse ist in beiden Fällen schräg nach hinten gezeichnet.

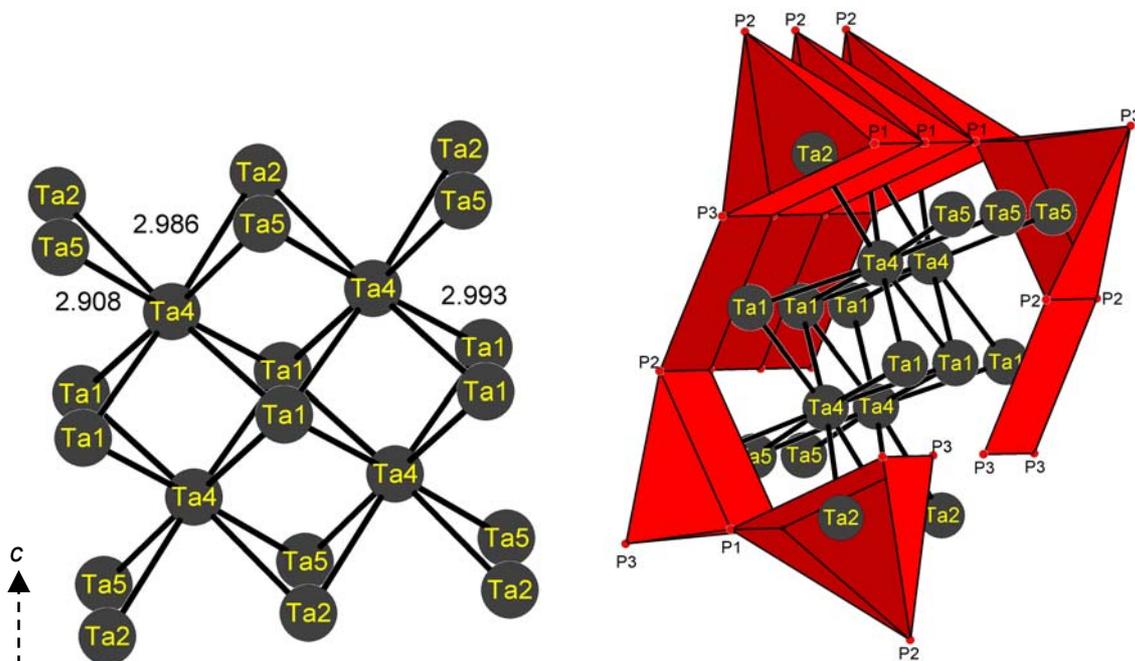


Abbildung 34: Links sind die flächenverknüpften innenzentrierten Ta₉-Würfel und rechts ist der innenzentrierte Ta₉-Würfel mit den Phosphorpolyedern dargestellt.

Die Struktur lässt sich auch mit unterschiedlichen P-Ta_x-Polyedern ($x = 7$ bis 9) beschreiben. Dieses ist schon länger bekannt [42] und soll der Vollständigkeit wegen hier noch aufgeführt werden.

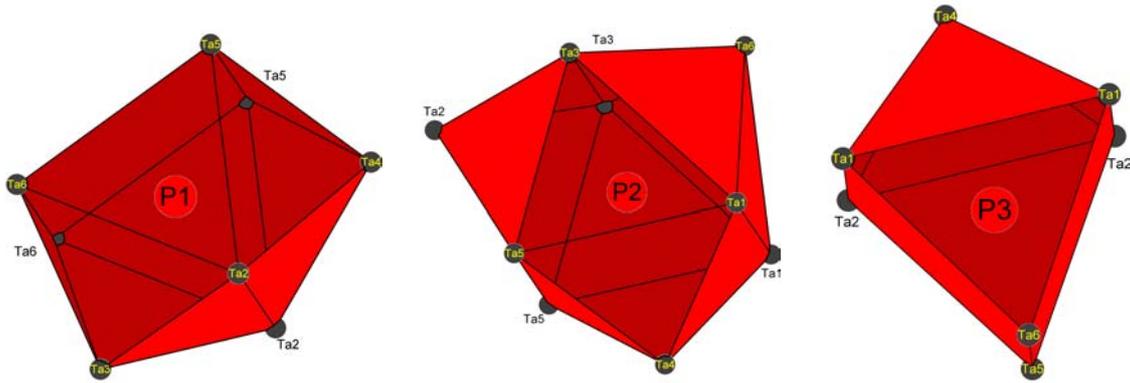


Abbildung 35: Die drei Phosphorpolyeder im Ta₂P.

Die drei kristallographisch verschiedenen Phosphoratome weisen eine unterschiedliche Koordinationssphäre mit Tantalatomen auf. P1 ist von einem zweifach überkappten trigonalen Prisma, P2 ist von einem dreifach überkappten trigonalen Prisma und P3 ist von einem einfach überkappten trigonalen Prisma aus Tantalatomen umgeben (s. Abbildung 35). Die P1-Polyeder sind, wie die P2 Polyeder, untereinander an der Basisfläche des trigonalen Prismas entlang der *c*-Achse flächenverknüpft (s. Abbildung 36). Demgegenüber sind die P3-Polyeder über die Seitenkante des trigonalen Prismas miteinander verknüpft, deren Ende von Ta1- und Ta2-Atomen besetzt sind. Die resultierende Polyederkette ist wie die P1- und die P2-Polyederkette entlang [001] ausgerichtet (s. Abbildung 37 links). Die P1-Polyeder besitzen mit den P2-Polyedern eine gemeinsame Fläche, die von einem überkappenden Tantalion (Ta4) aufgespannt wird (s. Abbildung 37 rechts). Die verbleibenden Verknüpfungen P1-P3-Polyeder (s. Abbildung 38 links) und P2-P3-Polyeder (s. Abbildung 38 rechts) weisen jeweils eine gemeinsame Kante auf, wobei die P2-P3-Polyeder zusätzlich noch über eine Ecke, die das überkappende Tantalion (Ta4) ist, miteinander verbunden sind (s. Abbildung 35). In Übereinstimmung mit der peripheren Lage des Ta4-Atoms in Abbildung 35 liegt es in Abbildung 32 im Zentrum des rot gekennzeichneten innenzentrierten Würfels. Das im Vergleich zum Ta4-Atom nicht ganz so am Rande liegende Ta3-Atom (im P1-Ta₈-Polyeder ist es wie Ta4 das überkappende Atom, während es im P2-Ta₉-Polyeder eine Seitenkante besetzt) zeigt aber dennoch die zentrale Lage des Ta3-Atoms in einem Tantalcluster an.

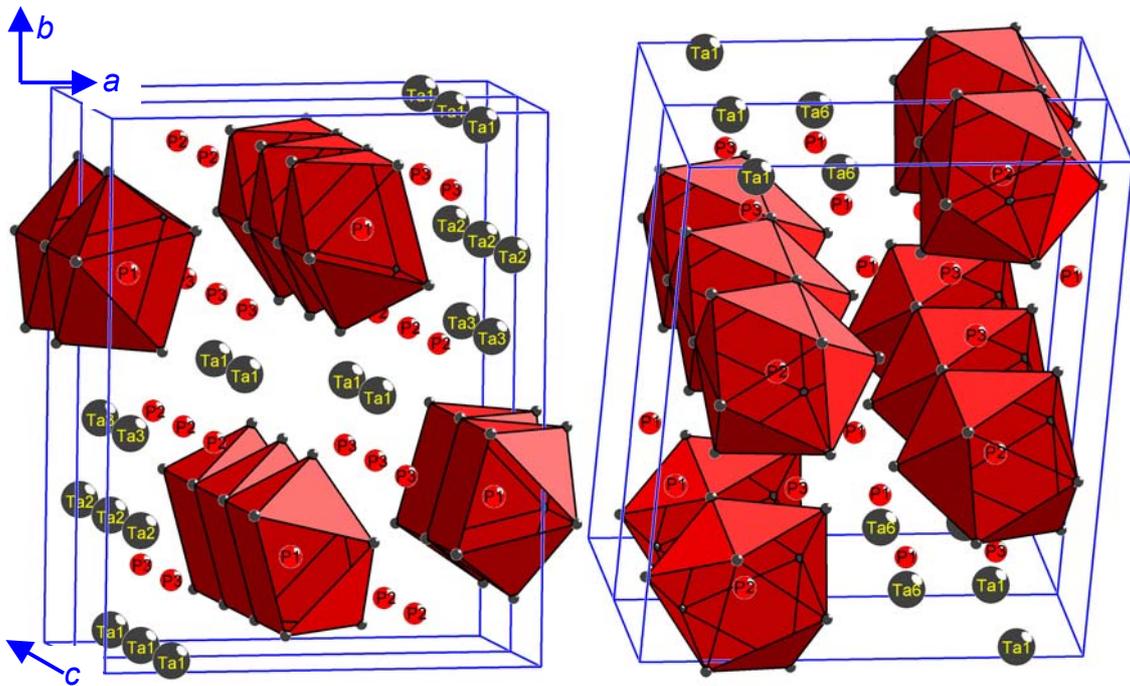


Abbildung 36: Phosphorpolyeder in Ta₂P. Links sind die P1- und rechts die P2-Polyeder in zwei Elementarzellen dargestellt. Die blaue Linie kennzeichnet die Elementarzellenbegrenzung.

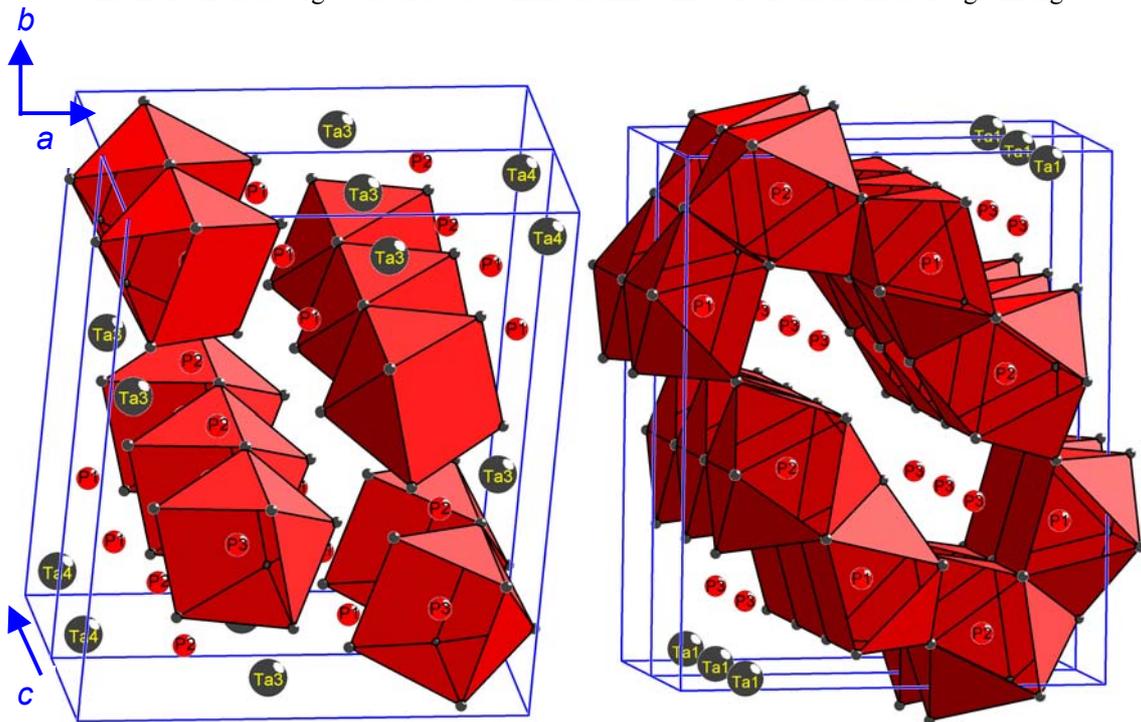


Abbildung 37: Phosphorpolyeder in Ta₂P. Links sind die P3- und rechts die P1- und die P2-Polyeder zusammen in zwei Elementarzellen gezeigt. Die blaue Linie kennzeichnet die Elementarzellenbegrenzung.

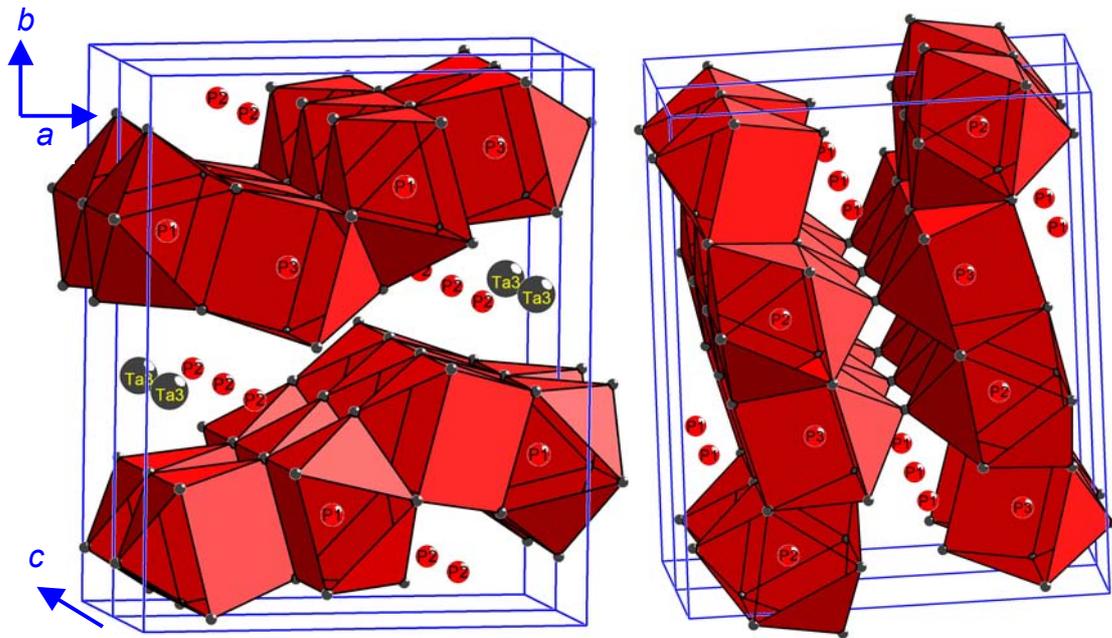


Abbildung 38: Phosphorpolyeder in Ta_2P . Links sind die P1- und die P3-Polyeder zusammen und rechts die P2- und die P3-Polyeder zusammen in zwei Elementarzellen gezeigt. Die blaue Linie kennzeichnet die Elementarzellenbegrenzung.

3.3 Strukturverfeinerung von $\text{Sc}_3\text{Sc}_{0.32}\text{Al}_{0.68}\text{O}_6$

Die Verbindung $\text{Sc}_3\text{Sc}_{0.32}\text{Al}_{0.68}\text{O}_6$ wurde kürzlich im Rahmen einer Dissertation synthetisiert und röntgenographisch charakterisiert. Sie ist isostrukturell zu Mg_3TeO_6 , was diese Untersuchungen gezeigt haben [47]. Sc_3CrO_6 als Mitglied dieser Strukturfamilie ist zwar als erstes in der Literatur erwähnt worden [48], gelöst werden konnte die Struktur aber erst mit Mg_3TeO_6 -Einkristallen [46]. Heutzutage existieren zahlreiche Verbindungen der Zusammensetzung M_3XO_6 $\text{M} = \text{Mn, Mg, Ca, Ho}$ und Sc , $\text{X} = \text{W, Te, U, Sc}$ und Cr , die in dieser ungewöhnlichen Struktur kristallisieren [46, 49-51]. Mg_3TeO_6 kristallisiert in der Raumgruppe $R\bar{3}$ (Nr. 148). Sauerstoff liegt dabei ebenfalls wie Mg auf der allgemeinen 18f-Lage, wohingegen Tellur sich auf die zwei speziellen Lagen 3a und 3b verteilt [46]. Im $\text{Sc}_3\text{Sc}_{0.32}\text{Al}_{0.68}\text{O}_6$ ist auf der mit Aluminium besetzten Position (3a bzw. 3b) auch Scandium vorhanden, so dass man auf die Stöchiometrie der Verbindung kommt. Dies ist möglich, da trotz des relativ großen Ionenradienunterschieds zwischen Al^{3+} (54 pm) und Sc^{3+} (75 pm) bei dem Sc_3CrO_6 das Cr^{3+} einen Ionenradius von 62 pm besitzt, so dass das kleinere Al^{3+} -Ion durch das größere Sc^{3+} im Mittel ausgeglichen wird (Alle angegebenen Ionenradien nach *Shannon* in oktaedrischer Koordination [53]). Die Gitterparameter von $\text{Sc}_3\text{Sc}_{0.32}\text{Al}_{0.68}\text{O}_6$ sind $a = 8.7722 \text{ \AA}$ und $c = 10.2030 \text{ \AA}$. Damit ist das Auswerten aller unter konvergenten Bedingungen erstellten Beugungsaufnahmen mit der hier entwickelten Methode möglich. Es befinden sich sechs Formeleinheiten in der Elementarzelle.

Da MSLS nur in der Dreiernotation Verfeinerungen berechnen kann, werden die Ergebnisse von MSLS gemäß Kap. 2.1.11 in die Vierernotation umgerechnet.

3.3.1 Die zur Strukturverfeinerung verwendeten Zonenachsen von $\text{Sc}_3\text{Sc}_{0.32}\text{Al}_{0.68}\text{O}_6$

Eine vernünftige Strukturverfeinerung mit MSLS ist nur möglich, wenn Beugungsaufnahmen unterschiedlicher Zonenachsen und mit mindestens zwei unterschiedlichen Kristalldicken vorliegen. Im folgenden werden die einzelnen Ergebnisse der Vorverfeinerung der einzelnen Zonenachsen dargelegt. Anschließend werden die Ergebnisse der damit gerechneten Strukturverfeinerung beschrieben und diskutiert. Es werden bei den einzelnen Beugungsaufnahmen nur noch die Anzahl der Reflexe der zentral ausgelesenen Datensätze angegeben, da nur diese für die Strukturverfeinerung verwendet wurden.

Alle Beugungsbilder wurden mit einer $30 \mu\text{m}$ großen C2-Kondensorblende und einer „spot size“ von 11 aufgenommen. Durch die Wahl der kleinsten „spot size“ am

Transmissionselektronenmikroskop wird der Durchmesser des einfallenden Elektronenstrahls auf der Probe auf ca. 5 nm verkleinert. Die kreisförmigen ausgelesenen Bereiche innerhalb der Beugungsscheiben besaßen einen Durchmesser von 10 Pixel. Alle Daten wurden mit der Muti-Scan-Kamera aufgezeichnet.

Die Vorverfeinerung wurde mit den Atompositionen des Strukturtyps Mg_3TeO_6 durchgeführt (Daten s. Tabelle 18), da von dieser Verbindung verlässliche Einkristallstrukturdaten vorliegen [46]. Die Besetzungsfaktoren wurden gemäß der Einwaage für die Synthese und den EDX-Analysen auf die Werte in der Tabelle 18 gesetzt [47]. Das dabei erhaltene Verhältnis zwischen Scandium und Aluminium wurden auf beiden Positionen 3a und 3b gleich eingesetzt.

Atom	x/a	y/a	z/c	Debye-Waller Faktor	Besetzungsfaktor	Wykhoff Pos.
Sc1	0.03993	0.26517	0.21157	1.0	1	18f
Sc2	0.0	0.0	0.5	1.0	0.32	3b
Sc3	0.0	0.0	0.0	1.0	0.32	3a
Al1	0.0	0.0	0.5	1.0	0.68	3b
Al2	0.0	0.0	0.0	1.0	0.68	3a
O1	0.02960	0.20080	0.40110	2.0	1	18f
O2	0.18530	0.15990	0.11550	2.0	1	18f

Tabelle 18: Startwerte der Strukturverfeinerung nach [46].

3.3.1.1 Zonenachse [0001] von $\text{Sc}_3\text{Sc}_{0.32}\text{Al}_{0.68}\text{O}_6$

In dieser Zonenachse wurden drei unter konvergenten Einstrahlbedingungen erstellte Beugungsbilder, die von Bereichen mit unterschiedlichen Kristalldicken herrühren, für die Strukturverfeinerung herangezogen.

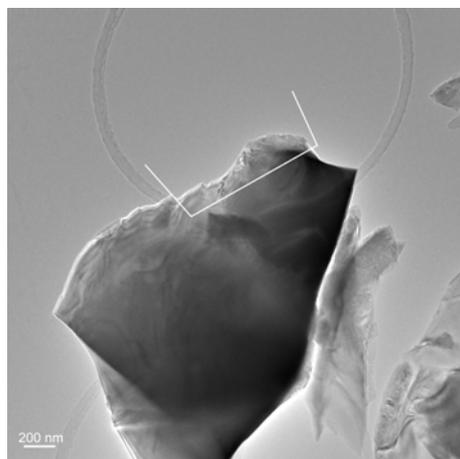
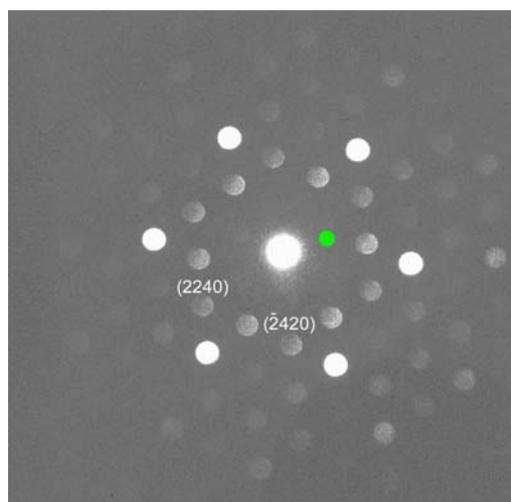
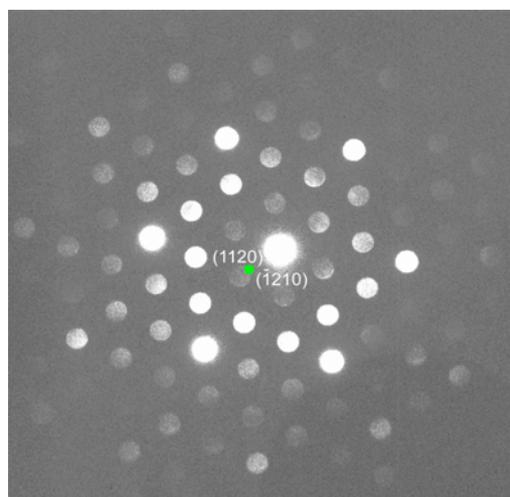


Abbildung 39: Hellfeldabbildung des untersuchten $\text{Sc}_3\text{Sc}_{0.32}\text{Al}_{0.68}\text{O}_6$ -Kristalls in [0001].

a)



b)



c)

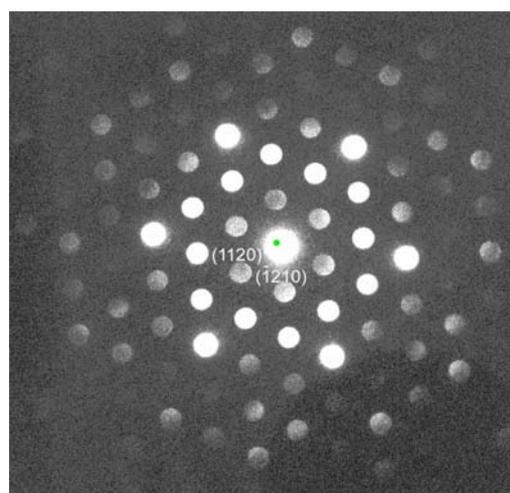


Abbildung 40: Drei Beugungsbilder von $\text{Sc}_3\text{Sc}_{0.32}\text{Al}_{0.68}\text{O}_6$ in $[0001]$. Kameralänge: 980 mm. Grüner Kreis: Verfeinertes Luezentrum des zentral ausgelesenen Datensatzes (Auslesekoordinaten x und $y = 0$). Größe des grünen Kreises = Fehler in der Bestimmung des Luezentrums (s. Tabellen 19 bis 21). Aufnahmezeit = 1 s.

Datensatz Nr.	Auslesekoor.		Lauzentrum				Dicke [Å]	Skalen- faktor	R- Wert
	x	y	h	k	i	l			
1	0	0	-1.2(3)	0.48(14)	-0.7(2)	0	74(10)	1.031(10)	0.4%
2	-10	0	-0.9(3)	0.33(12)	-0.6(2)	0	80(15)	1.027(11)	0.6%
3	10	0	-0.7(3)	0.24(10)	-0.46(20)	0	92(14)	1.037(11)	0.3%
4	0	-10	-1.3(3)	0.60(15)	-0.7(2)	0	72(9)	1.019(17)	0.4%
5	0	10	-0.8(3)	0.24(9)	-0.56(20)	0	89(21)	1.033(17)	0.4%
6	5	5	-0.9(3)	0.32(11)	-0.6(2)	0	84(15)	1.031(14)	0.3%
7	-5	-5	-1.5(4)	0.8(2)	-0.7(3)	0	69(10)	1.027(12)	0.5%

Datensatz Nr.	Auslesekoor.		Lauzentrum				Dicke [Å]	Skalen- faktor	R- Wert
	x	y	h	k	i	l			
1	0	0	0.85(10)	0.83(8)	1.68(9)	0	137(5)	0.994(10)	0.4%
2	-10	0	0.74(9)	0.82(9)	1.56(9)	0	137(5)	0.995(9)	0.4%
3	10	0	0.90(11)	0.81(9)	1.71(10)	0	142(5)	0.995(9)	0.3%
4	0	-10	0.80(7)	1.00(8)	1.80(8)	0	134(3)	1.010(8)	0.3%
5	0	10	0.95(12)	0.66(8)	1.61(10)	0	138(5)	0.989(9)	0.4%
6	5	5	1.01(9)	0.83(7)	0.92(8)	0	135(3)	0.990(9)	0.3%
7	-5	-5	0.74(8)	0.86(8)	1.6(8)	0	138(4)	1.001(8)	0.4%

Datens. Nr.	Auslesekoor.		Lauzentrum				Dicke [Å]	Skalen- faktor	R- Wert
	x	y	h	k	i	l			
1	0	0	0.10(2)	-0.16(3)	-0.06(3)	0	158(9)	0.946(10)	0.9%
2	-10	0	-0.042(19)	-0.14(3)	-0.18(2)	0	153(10)	0.950(10)	0.9%
3	10	0	0.25(4)	-0.17(3)	0.08(4)	0	162(9)	0.936(14)	1.1%
4	0	-10	0.042(18)	-0.03(2)	0.012(19)	0	153(9)	0.949(14)	0.9%
5	0	10	0.19(3)	-0.30(4)	0.11(4)	0	159(9)	0.942(18)	1.0%
6	5	5	0.21(3)	0.22(3)	0.43(3)	0	163(8)	0.946(17)	1.0%
7	-5	-5	-0.008(16)	-0.071(20)	-0.079(18)	0	156(8)	0.949(14)	0.8%

Tabellen 19, 20 und 21: Berechnete Werte nach der Vorverfeinerung mit Intensitätsdaten in der [0001]-Zonenachse von $\text{Sc}_3\text{Sc}_{0.32}\text{Al}_{0.68}\text{O}_6$. Tabelle 19 gehört zur Abbildung 40a, Tabelle 20 zur Abbildung 40b und Tabelle 21 zur Abbildung 40c.

Die Beugungsaufnahmen wurden von dem Kristall in Abbildung 39 erstellt. Der weiße Rahmen kennzeichnet das Gebiet des Kristalls, das zur Erstellung der Aufnahmen verwendet wurde.

Die Ergebnisse der Vorverfeinerung mit Intensitätsdaten in der [0001]-Zonenachse sind in den Tabellen 19 bis 21 gezeigt.

Der zentral ausgelesene Datensatz des Beugungsbildes in Abbildung 40a enthält 65, der zur Abbildung 40b 76 und der zur Abbildung 40c ebenfalls 76 verwertbare Reflexe.

Die ansteigende Kristalldicke lässt sich in den Beugungsaufnahmen der Abbildung 40 a-c u.a. an der ansteigenden Intensität der Reflexe vom Typ $(11\bar{2}0)$ erkennen.

Alle verfeinerten Ergebnisse der Skalenfaktoren und die der Kristalldicken in den Tabellen 19 bis 21 sind zu den selben Werten konvergiert. Die mit ansteigender Probedicke kleiner werdenden Fehler bei den Ergebnissen der Luezentren basieren auf dem Anstieg der Intensität den bei dünnen Kristalldicken intensitätsschwachen Reflexen. Dadurch stehen mehr und verlässlichere Daten zur Bestimmung des Luezentrens zur Verfügung. Die ausgelesenen Einstrahlrichtungen werden besonders gut bei den verfeinerten Werten des Luezentrens der Aufnahme in Abbildung 40c (= Tabelle 21) reproduziert.

3.3.1.2 Zonenachse $[2\bar{1}\bar{1}0]$ von $\text{Sc}_3\text{Sc}_{0.32}\text{Al}_{0.68}\text{O}_6$

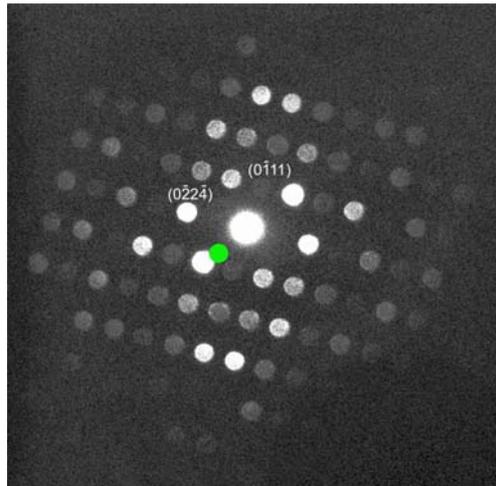
Bei dieser Zonenachse handelt es sich um die Richtung parallel zur a -Achse. Sie stellt damit die zweite Hauptzonenachse in diesem System dar. Wie auch schon in der $[0001]$ -Richtung sind auch hier drei Aufnahmen mit jeweils unterschiedlichen Probedicken zur Strukturverfeinerung herangezogen worden.

Die Beugungsaufnahmen wurden von dem Kristall in Abbildung 42 erstellt. Die mit den weißen Pfeilen markierten Bereiche kennzeichnen die Gebiete des Kristalls, die zur Erstellung der Beugungsaufnahmen verwendet wurden.

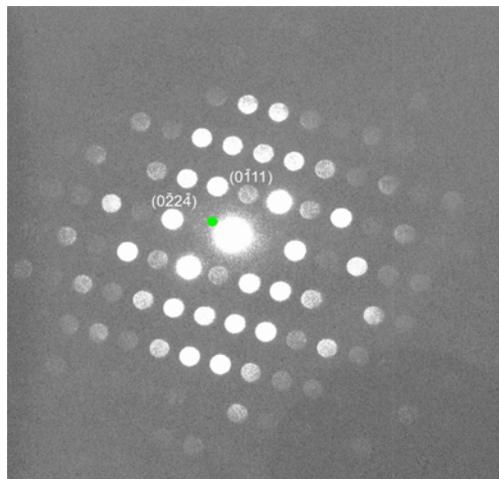
Der zentral ausgelesene Datensatz des Beugungsbildes in Abbildung 41a enthält 98, der zur Abbildung 41b 84 und der zur Abbildung 41c 96 verwertbare Reflexe.

Die Ergebnisse der Vorverfeinerung der $[2\bar{1}\bar{1}0]$ -Zonenachse sind in den Tabellen 22-24 gezeigt. Wie bei den Ergebnissen der Vorverfeinerung in der Zonenachse $[0001]$ (s. Tabellen 19 bis 21), zeigen auch die Daten nach der Vorverfeinerung in der Zonenachse $[2\bar{1}\bar{1}0]$ dieselben Zusammenhänge auf. Die Werte der verfeinerten Skalenfaktoren und die der Kristalldicke sind zu den selben Zahlen konvergiert. Die Bestimmung des Luezentrens ist ebenfalls mit größer werdender Kristalldicke mit einem kleineren Fehler behaftet.

a)



b)



c)

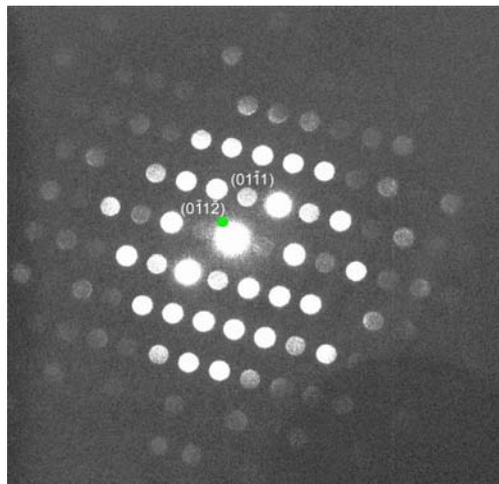


Abbildung 41: Drei Beugungsbilder von $\text{Sc}_3\text{Sc}_{0.32}\text{Al}_{0.68}\text{O}_6$ in $[2\bar{1}\bar{1}0]$. Kameralänge: 780 mm. Grüner Kreis: Verfeinertes Lauezentrum des zentral ausgelesenen Datensatzes (Auslesekoordinaten x und $y = 0$). Größe des grünen Kreises = Fehler in der Bestimmung des Lauezentrums (s. Tabellen 22 bis 24). Aufnahmezeit: 1 s.

Datensatz Nr.	Auslesekoor.		Lauzentrum				Dicke [Å]	Skalen- faktor	R- Wert
	<i>x</i>	<i>y</i>	<i>h</i>	<i>k</i>	<i>i</i>	<i>l</i>			
1	0	0	0	0.2(5)	-0.2(5)	-2.0(7)	45(6)	1.008(14)	0.6%
2	-10	0	0	0.0(8)	0.0(8)	-3.6(15)	38(7)	1.036(13)	0.5%
3	10	0	0	0.3(5)	-0.3(5)	-1.4(5)	49(3)	1.039(13)	0.5%
4	0	-10	0	-0.3(6)	0.3(6)	-2.3(9)	43(6)	1.024(15)	0.5%
5	0	10	0	0.1(5)	-0.1(5)	-2.2(6)	45(5)	1.030(15)	0.5%
6	5	5	0	0.9(5)	-0.9(5)	-2.0(12)	48(6)	1.003(13)	0.6%
7	-5	-5	0	0.1(8)	-0.1(8)	-3.1(14)	35(7)	1.029(20)	0.6%

Datensatz Nr.	Auslesekoor.		Lauzentrum				Dicke [Å]	Skalen- faktor	R- Wert
	<i>x</i>	<i>y</i>	<i>h</i>	<i>k</i>	<i>i</i>	<i>l</i>			
1	0	0	0	-0.9(3)	0.9(3)	-1.2(2)	79(5)	0.960(9)	0.5%
2	-10	0	0	-1.1(3)	1.1(3)	-1.2(3)	76(5)	0.963(8)	0.5%
3	10	0	0	-1.0(2)	1.0(2)	-0.9(2)	78(5)	0.975(8)	0.4%
4	0	-10	0	-1.3(3)	1.3(3)	-1.2(2)	76(4)	0.964(9)	0.4%
5	0	10	0	-1.1(3)	1.1(3)	-0.7(3)	73(5)	0.959(12)	0.5%
6	5	5	0	-1.1(3)	1.1(3)	-1.0(3)	74(5)	0.961(12)	0.5%
7	-5	-5	0	-1.2(2)	1.2(2)	-1.0(2)	73(4)	0.978(9)	0.4%

Datensatz Nr.	Auslesekoor.		Lauzentrum				Dicke [Å]	Skalen- faktor	R- Wert
	<i>x</i>	<i>y</i>	<i>h</i>	<i>k</i>	<i>i</i>	<i>l</i>			
1	0	0	0	-0.85(13)	0.85(13)	-0.72(9)	128(5)	0.937(8)	0.5%
2	-10	0	0	-0.89(14)	0.89(14)	-0.63(11)	123(5)	0.939(8)	0.5%
3	10	0	0	-0.65(12)	0.65(12)	-0.78(9)	128(5)	0.947(8)	0.5%
4	0	-10	0	-0.92(13)	0.92(13)	-0.94(10)	130(4)	0.950(8)	0.5%
5	0	10	0	-0.72(11)	0.72(11)	-0.59(8)	128(4)	0.953(8)	0.4%
6	5	5	0	-0.72(13)	0.72(13)	-0.70(9)	125(5)	0.945(7)	0.5%
7	-5	-5	0	-0.85(13)	0.85(13)	-0.80(9)	128(5)	0.942(8)	0.5%

Tabellen 22, 23 und 24: Berechnete Werte nach der Vorverfeinerung mit Intensitätsdaten in der $[2\bar{1}\bar{1}0]$ -Zonenachse von $\text{Sc}_3\text{Sc}_{0.32}\text{Al}_{0.68}\text{O}_6$. Tabelle 22 gehört zur Abbildung 41a, Tabelle 23 zur Abbildung 41b und Tabelle 24 zur Abbildung 41c.

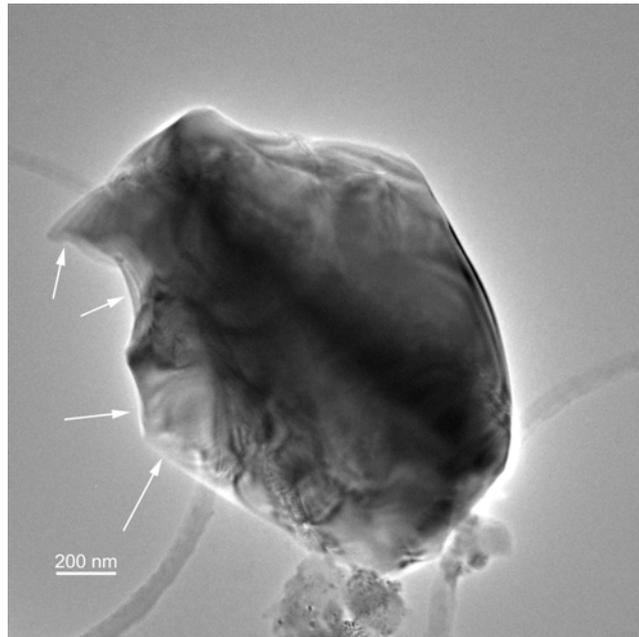


Abbildung 42: Hellfeldabbildung des untersuchten $\text{Sc}_3\text{Sc}_{0.32}\text{Al}_{0.68}\text{O}_6$ -Kristalls in $[2\bar{1}\bar{1}0]$.

3.3.1.3 Zonenachse $[10\bar{1}1]$ von $\text{Sc}_3\text{Sc}_{0.32}\text{Al}_{0.68}\text{O}_6$

In dieser Zonenachse wurden drei Beugungsbilder von Bereichen mit unterschiedlichen Kristalldicken aufgenommen. Die drei Beugungsbilder wurden in dem mit einer weißen Begrenzung umrahmten Bereich aufgezeichnet. Aus der zur Kristallmitte geringer werdenden Transparenz lässt sich eine kontinuierliche Zunahme der Kristalldicke ablesen. Diese Art von Kristallen eignen sich besonders gut für diese Methode, da Datensätze mit unterschiedlicher Probendicke leicht zugänglich sind (s. Abbildung 43).

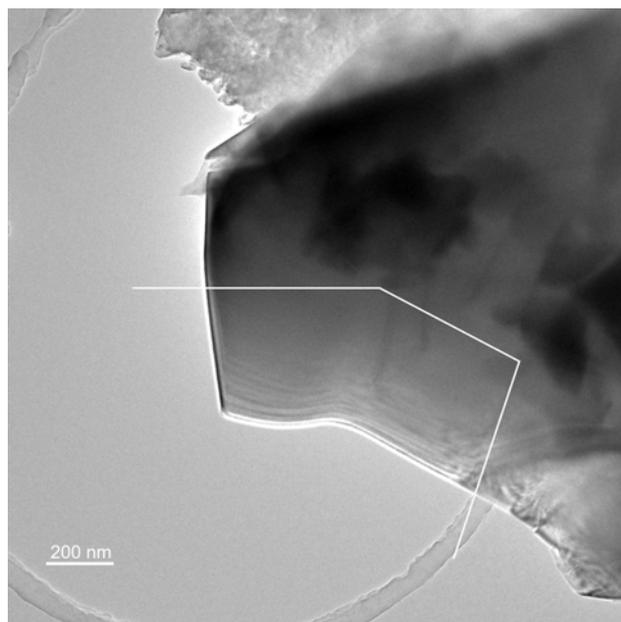
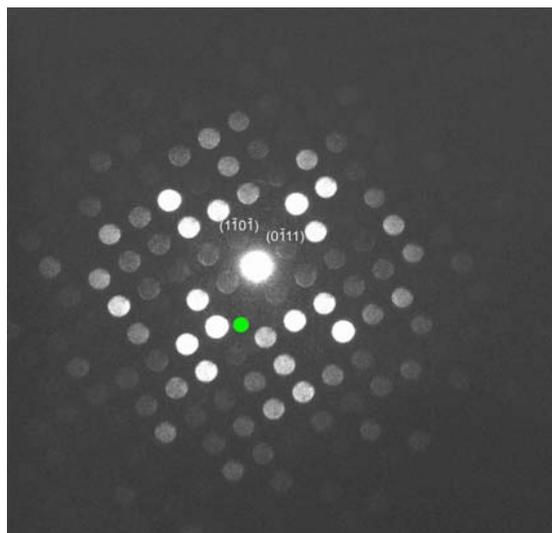
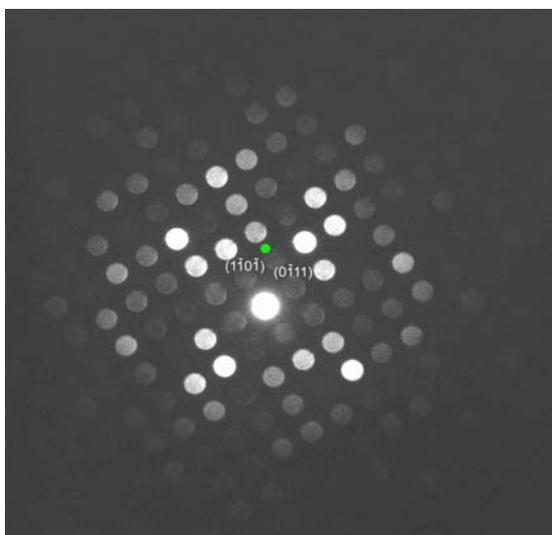


Abbildung 43: Hellfeldabbildung des untersuchten $\text{Sc}_3\text{Sc}_{0.32}\text{Al}_{0.68}\text{O}_6$ -Kristalls in $[10\bar{1}1]$.

a)



b)



c)

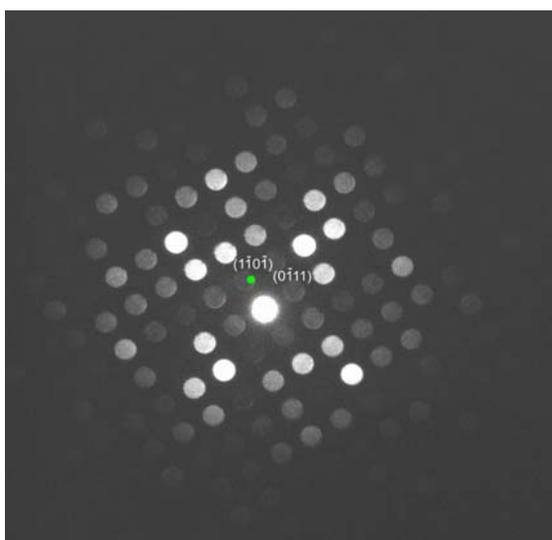


Abbildung 44: Drei Beugungsbilder von $\text{Sc}_3\text{Sc}_{0.32}\text{Al}_{0.68}\text{O}_6$ in $[10\bar{1}1]$. Kameralänge: 780 mm. Grüner Kreis: Verfeinertes Luezentrum des zentral ausgelesenen Datensatzes (Auslesekoordinaten x und $y = 0$). Größe des grünen Kreises = Fehler in der Bestimmung des Luezentrums (s. Tabellen 25-27). Aufnahmezeit: 1 s.

Datenstz. Nr.	Auslesekoord.		Lauzentrum				Dicke [Å]	Skalen- faktor	R- Wert
	<i>x</i>	<i>y</i>	<i>h</i>	<i>k</i>	<i>i</i>	<i>l</i>			
1	0	0	-1.29(20)	2.6(4)	-1.3(3)	-0.05(18)	61(4)	0.997(12)	1.7%
2	-10	0	-0.96(12)	2.0(3)	-1.0(2)	-0.05(12)	69(3)	0.978(10)	1.3%
3	10	0	-1.10(13)	2.6(3)	-1.5(2)	-0.44(14)	65(3)	0.988(10)	1.3%
4	0	-10	-1.10(15)	2.7(3)	-1.6(2)	-0.49(17)	65(3)	1.003(11)	1.6%
5	0	10	-0.98(18)	2.0(4)	-1.0(3)	-0.02(18)	61(4)	1.005(13)	1.8%
6	5	5	-0.96(14)	2.0(3)	-1.0(2)	-0.10(15)	67(4)	0.999(12)	1.7%
7	-5	-5	-1.06(15)	2.4(3)	-1.3(2)	-0.24(14)	65(3)	0.993(10)	1.4%

Datenstz. Nr.	Auslesekoord.		Lauzentrum				Dicke [Å]	Skalen- faktor	R- Wert
	<i>x</i>	<i>y</i>	<i>h</i>	<i>k</i>	<i>i</i>	<i>l</i>			
1	0	0	1.55(16)	-2.3(3)	0.8(2)	-0.80(14)	84(3)	0.981(12)	2.0%
2	-10	0	1.63(15)	-2.4(3)	0.8(2)	-0.82(13)	86(3)	0.989(12)	1.9%
3	10	0	1.52(15)	-2.1(3)	0.6(2)	-0.93(13)	88(3)	0.986(12)	1.9%
4	0	-10	1.41(15)	-2.0(3)	0.6(2)	-0.83(14)	83(4)	0.983(12)	1.9%
5	0	10	1.62(15)	-2.4(3)	0.8(2)	-0.83(13)	87(3)	0.990(12)	1.9%
6	5	5	1.51(14)	-2.2(3)	0.7(2)	-0.85(12)	88(3)	0.985(12)	2.0%
7	-5	-5	1.43(13)	-2.1(3)	0.7(2)	-0.74(12)	86(3)	0.979(12)	1.9%

Datens. Nr.	Auslesekoord.		Lauzentrum				Dicke [Å]	Skalen- faktor	R- Wert
	<i>x</i>	<i>y</i>	<i>h</i>	<i>k</i>	<i>i</i>	<i>l</i>			
1	0	0	0.94(6)	-1.17(14)	0.23(10)	-0.72(7)	104(3)	0.975(9)	1.2%
2	-10	0	0.91(6)	-1.2(13)	0.29(10)	-0.61(6)	104(2)	0.974(9)	1.2%
3	10	0	0.96(6)	-1.15(13)	0.19(10)	-0.77(7)	107(2)	0.979(10)	1.3%
4	0	-10	0.85(6)	-1.01(13)	0.16(10)	-0.70(7)	106(2)	0.980(10)	1.4%
5	0	10	1.04(7)	-1.35(16)	0.31(10)	-0.73(7)	103(3)	0.978(10)	1.2%
6	5	5	0.94(5)	-1.17(12)	0.23(9)	-0.70(6)	107(2)	0.982(9)	1.1%
7	-5	-5	0.89(6)	-1.09(14)	0.20(10)	-0.70(7)	105(3)	0.973(9)	1.2%

Tabellen 25, 26 und 27: Berechnete Werte nach der Vorverfeinerung mit Intensitätsdaten in der $[10\bar{1}1]$ -Zonenachse von $\text{Sc}_3\text{Sc}_{0.32}\text{Al}_{0.68}\text{O}_6$. Tabelle 25 gehört zur Abbildung 44a, Tabelle 26 zur Abbildung 44b und Tabelle 27 zur Abbildung 44c.

Der zentral ausgelesene Datensatz des Beugungsbildes in Abbildung 44a enthält 136, der zur Abbildung 44b 149 und der zur Abbildung 44c 146 verwertbare Reflexe.

Alle Daten der Skalenfaktoren und die der verfeinerten Kristalldicke der Tabellen 25 bis 27 konvergieren im Rahmen der Fehlergrenzen zu den selben Werten.

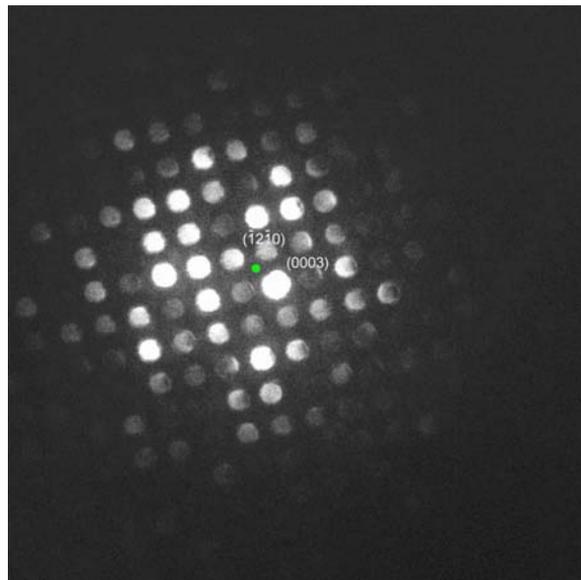
Die verschiedenen Einstrahlrichtungen der einzelnen Datensätze in Tabelle 27 werden tendenziell in den verfeinerten Daten der Lauzentren wiedergegeben. Bei den Ergebnissen der Lauzentren in den Tabellen 25 und 26 ist der dazugehörige Fehler

größer als der Abstand der einzelnen Datensätze innerhalb des Beugungsbildes, so dass hier keine eindeutigen Aussagen erhalten werden.

3.3.1.4 Zonenachse $[10\bar{1}0]$ von $\text{Sc}_3\text{Sc}_{0.32}\text{Al}_{0.68}\text{O}_6$

Wie in Kapitel 2.1.11 zu sehen ist, handelt es sich bei dieser Zonenachse um die für die trigonale Symmetrie notwendige zweite Richtung in der Basalebene. Von dieser Richtung wurden zwei Beugungsbilder zur Strukturverfeinerung herangezogen.

a)



b)

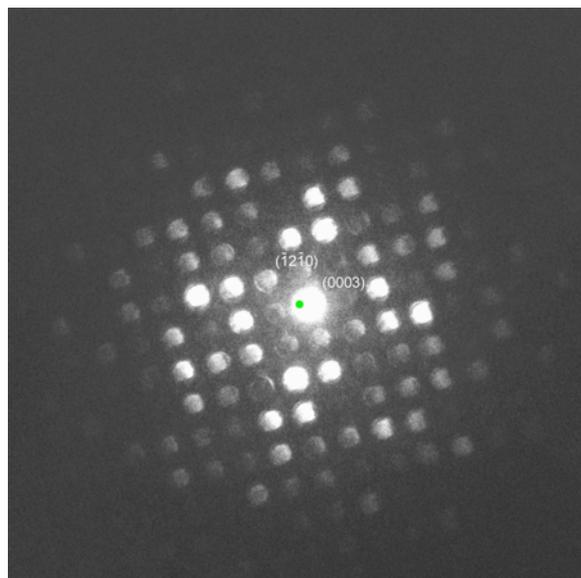
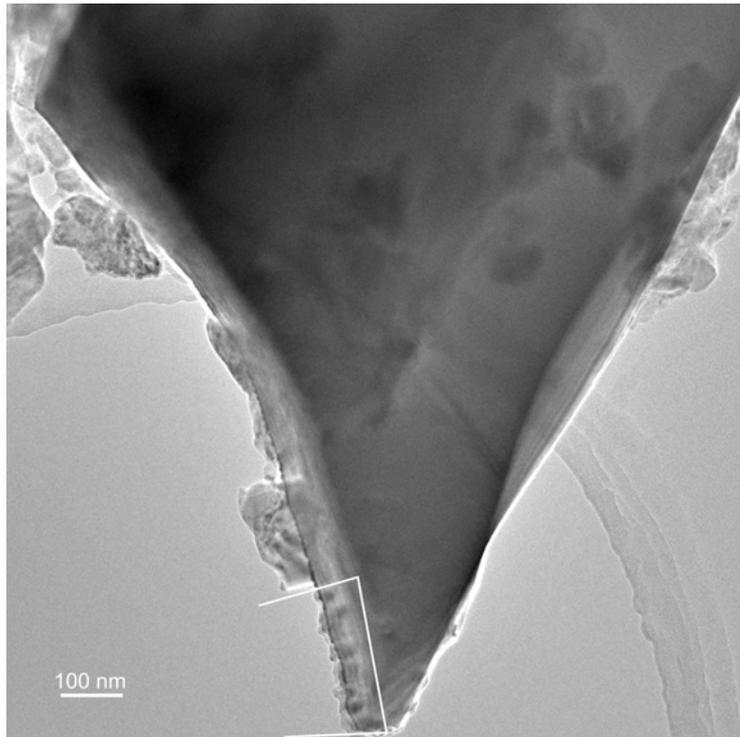


Abbildung 45: Zwei Beugungsbilder von $\text{Sc}_3\text{Sc}_{0.32}\text{Al}_{0.68}\text{O}_6$ in $[10\bar{1}0]$. Kameralänge: 780 mm. Grüner Kreis: Verfeinertes Laugezentrum des zentral ausgelesenen Datensatzes (Auslesekoordinaten x und $y = 0$).

Größe des grünen Kreises = Fehler in der Bestimmung des Laugezentrums (s. Tabellen 28 und 29).

Aufnahmezeit: 1 s.


 Abbildung 46: Hellfeldabbildung des untersuchten $\text{Sc}_3\text{Sc}_{0.32}\text{Al}_{0.68}\text{O}_6$ -Kristalls in $[10\bar{1}0]$.

Datns. Nr.	Auslesekoord.		Lauzentrum				Dicke [Å]	Skalen- faktor	R- Wert
	<i>x</i>	<i>y</i>	<i>h</i>	<i>k</i>	<i>i</i>	<i>l</i>			
1	0	0	-0.65(16)	1.3(3)	-0.7(2)	-1.22(9)	123(4)	0.985(8)	0.3%
2	-10	0	-0.67(11)	1.3(2)	-0.63(16)	-1.02(8)	126(4)	0.980(7)	0.2%
3	10	0	-0.69(11)	1.4(2)	-0.71(16)	-0.93(9)	116(3)	0.988(6)	0.2%
4	0	-10	-0.75(16)	1.5(3)	-0.7(2)	-1.13(13)	119(4)	0.988(9)	0.3%
5	0	10	-0.61(11)	1.2(2)	-0.59(16)	-1.12(8)	123(3)	0.977(7)	0.2%
6	5	5	-0.63(13)	1.3(3)	-0.67(2)	-1.21(10)	120(4)	0.985(8)	0.3%
7	-5	-5	-0.58(15)	1.2(3)	-0.6(2)	-1.17(12)	129(4)	0.986(11)	0.4%

Datns. Nr.	Auslesekoord.		Lauzentrum				Dicke [Å]	Skalen- faktor	R- Wert
	<i>x</i>	<i>y</i>	<i>h</i>	<i>k</i>	<i>i</i>	<i>l</i>			
1	0	0	-0.08(15)	0.2(3)	-0.1(2)	-1.10(11)	136(6)	0.937(10)	0.6%
2	-10	0	0.0(2)	0.0(4)	0.0(3)	-0.68(17)	126(7)	0.919(12)	0.7%
3	10	0	-0.16(11)	0.3(2)	-0.14(16)	-1.05(7)	146(5)	0.938(9)	0.4%
4	0	-10	-0.2(14)	0.4(3)	-0.2(2)	-0.92(11)	137(5)	0.930(10)	0.5%
5	0	10	-0.11(15)	0.2(3)	-0.1(2)	-0.99(12)	135(6)	0.938(10)	0.6%
6	5	5	-0.09(13)	0.2(3)	-0.11(17)	-0.97(9)	145(6)	0.943(10)	0.5%
7	-5	-5	-0.13(17)	0.3(3)	-0.2(2)	-0.77(14)	127(6)	0.924(11)	0.5%

Tabellen 28 und 29: Berechnete Werte nach der Vorverfeinerung mit Intensitätsdaten in der $[10\bar{1}0]$ -Zonenebene von $\text{Sc}_3\text{Sc}_{0.32}\text{Al}_{0.68}\text{O}_6$. Tabelle 28 gehört zur Abbildung 45a und Tabelle 29 zur Abbildung

45b.

Der zentral ausgelesene Datensatz des Beugungsbildes in Abbildung 45a enthält 52 und der zur Abbildung 45b 56 verwertbare Reflexe.

Die Beugungsaufnahmen wurden von dem mit dem weißen Rahmen markierten Bereich an der Kristallspitze erstellt (s. Abbildung 46). Dieser Kristall ist im Gegensatz zu den übrigen hier gezeigten Kristallen von $\text{Sc}_3\text{Sc}_{0.32}\text{Al}_{0.68}\text{O}_6$ im Elektronenstrahl nicht so stabil. Das äußert sich zum einen in den nicht so klaren Beugungsbildern, die trotz der kurzen Aufnahmezeit von einer Sekunde für das menschliche Auge nicht besser aufgezeichnet werden konnten (s. Abbildung 45b). Zum anderen sind in Abbildung 46 an der Kristallspitze dunkle Kontraste zu erkennen, die erst nach der Erstellung der Beugungsbilder erschienen. Auch andere hier nicht gezeigte Kristalle haben in dieser Orientierung in gleicher Weise reagiert. Demnach hat es den Anschein, dass es eine richtungsabhängige Stabilität von $\text{Sc}_3\text{Sc}_{0.32}\text{Al}_{0.68}\text{O}_6$ im Elektronenstrahl gibt.

Die Variationen der verfeinerten Kristalldicken in Tabelle 29 sind allerdings an der Grenze der erlaubten Abweichung. Der Datensatz Nr. 2 liegt, wenn der maximale Fehler berücksichtigt wird, bei einer maximalen Dicke von 133 Å. Das andere Extrem von 141 Å wurde bei der Verfeinerung mit Datensatz Nr. 3 der Tabelle 29 erhalten. Das entspricht in dieser Richtung einer ganzen Elementarzelle, was als äußerste Toleranzgrenze dieser Methode noch akzeptiert werden kann.

Die Resultate der Kristalldicken und die der Skalenfaktoren, die in Tabelle 28 gezeigt sind, konvergieren zu den selben Werten. Da die in Tabelle 28 und 29 dargestellten Fehler der Lauezentren größer sind als der Orientierungsunterschied der ausgelesenen Datensätze, können keine Aussagen über die Konsistenz dieser Daten gemacht werden.

3.3.1.5 Zonenachsen $[01\bar{1}2]$ und $[20\bar{2}\bar{1}]$ von $\text{Sc}_3\text{Sc}_{0.32}\text{Al}_{0.68}\text{O}_6$

Die Nebenzonenachsen $[01\bar{1}2]$ und $[20\bar{2}\bar{1}]$ wurden gewählt, weil für die Strukturverfeinerung neben den Hauptzonenachsen $[0001]$, $[2\bar{1}\bar{1}0]$ und $[10\bar{1}0]$ und der Nebenzonenachse $[10\bar{1}1]$ in diesen Orientierungen im Gegensatz zu den bisher behandelten Einstrahlrichtungen die Kationen nicht von der Sauerstoffatomen überdeckt werden (s. Abbildung 47a und b). In allen anderen Zonenachsen werden sie immer von den Sauerstoffatomen mindestens teilweise überdeckt.

Die Zonenachsen $[01\bar{1}2]$ und $[10\bar{1}\bar{2}]$ sind kristallographisch äquivalent und unterscheiden sich hier nur in ihrer Indizierung. Die Kristalle von dem diese drei Beugungsaufnahmen erstellt wurden sind in Abbildung 49 ($[01\bar{1}2]$ bzw. $[10\bar{1}\bar{2}]$) und in Abbildung 50 ($[20\bar{2}\bar{1}]$) gezeigt. Die Beugungsaufnahmen sind jeweils in dem Bereich innerhalb des weißen Rahmens aufgezeichnet worden.

Der zentral ausgelesene Datensatz des Beugungsbildes in Abbildung 48a enthält 53, der zur Abbildung 48b 103 und der zur Abbildung 48c 63 verwertbare Reflexe.

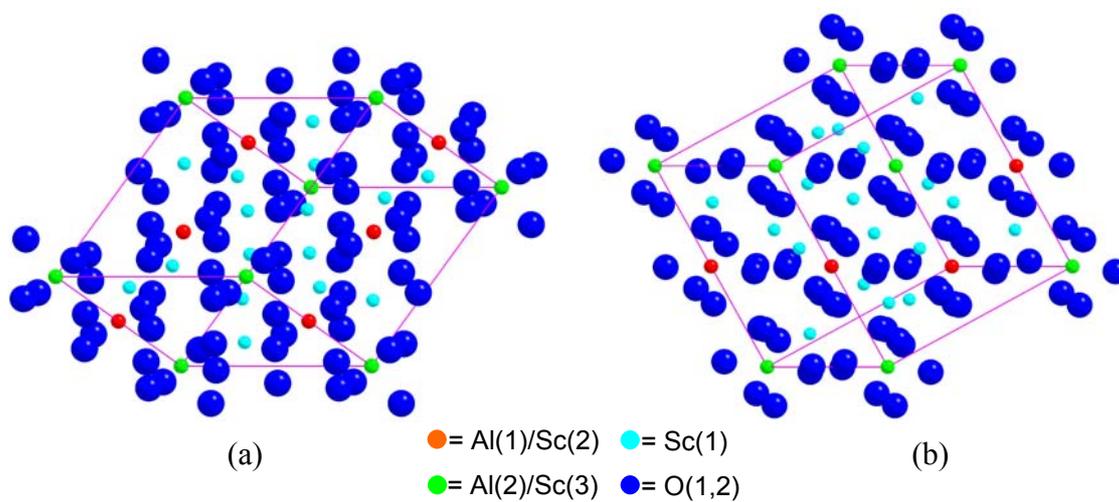
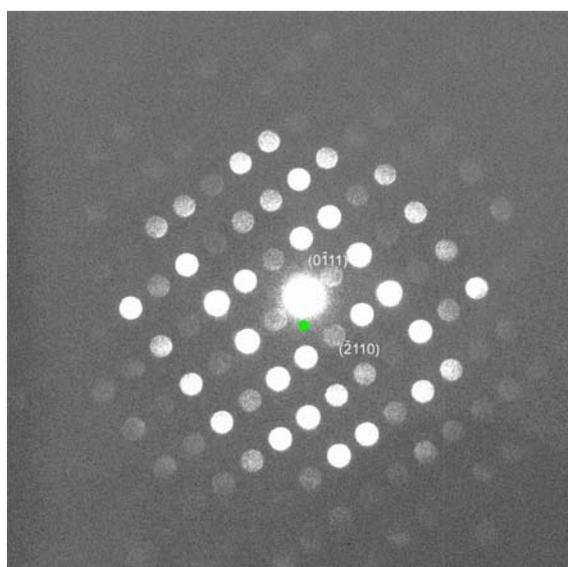
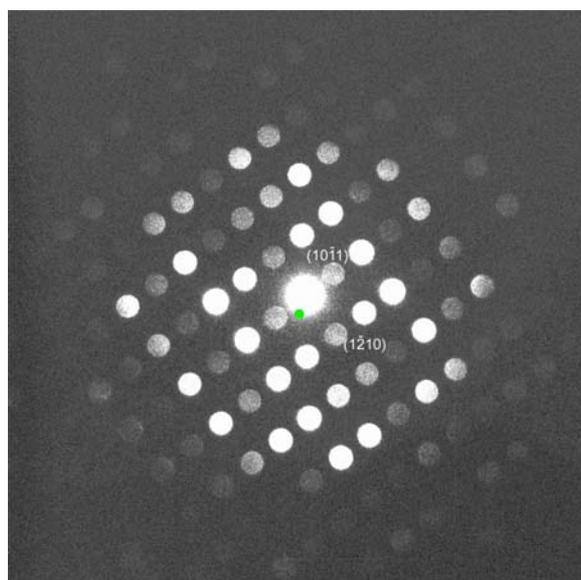


Abbildung 47: Elementarzelle von $\text{Sc}_3\text{Sc}_{0.32}\text{Al}_{0.68}\text{O}_6$ in $[01\bar{1}2]$ (a) und in $[202\bar{1}]$ (b).

a)



b)



c)

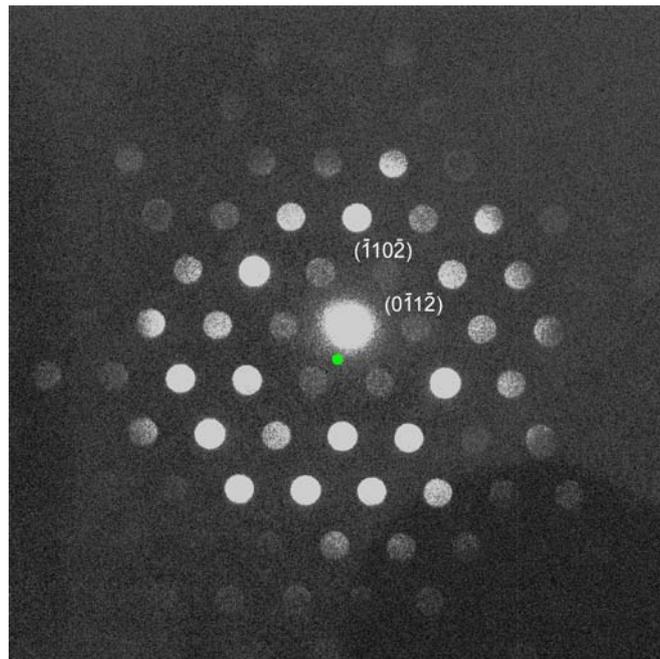


Abbildung 48: Beugungsbilder von $\text{Sc}_3\text{Sc}_{0.32}\text{Al}_{0.68}\text{O}_6$ in $[01\bar{1}2]$ bzw. $[10\bar{1}\bar{2}]$ (a und b) und in $[20\bar{2}\bar{1}]$ (c). Kameralänge: 780 mm (a und b) bzw. 980 mm (c). Grüner Kreis: Verfeinertes Lauezentrum des zentral ausgelesenen Datensatzes (Auslesekoordinaten x und $y = 0$). Größe des grünen Kreises = Fehler in der Bestimmung des Lauezentrums (s. Tabellen 30 bis 32). Aufnahmezeit: 1 s.

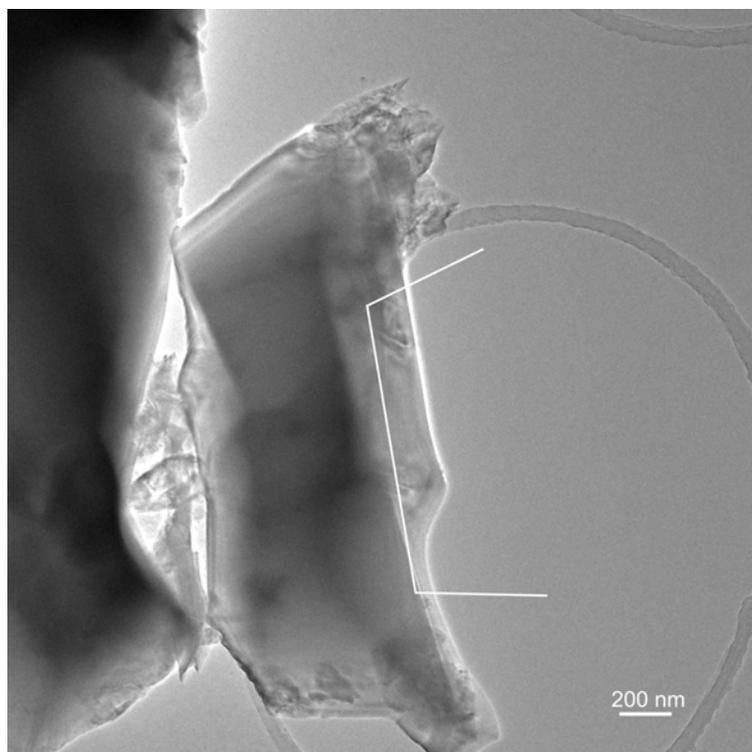


Abbildung 49: Hellfeldabbildung des untersuchten $\text{Sc}_3\text{Sc}_{0.32}\text{Al}_{0.68}\text{O}_6$ -Kristalls in $[10\bar{1}\bar{2}]$ bzw. $[01\bar{1}2]$.

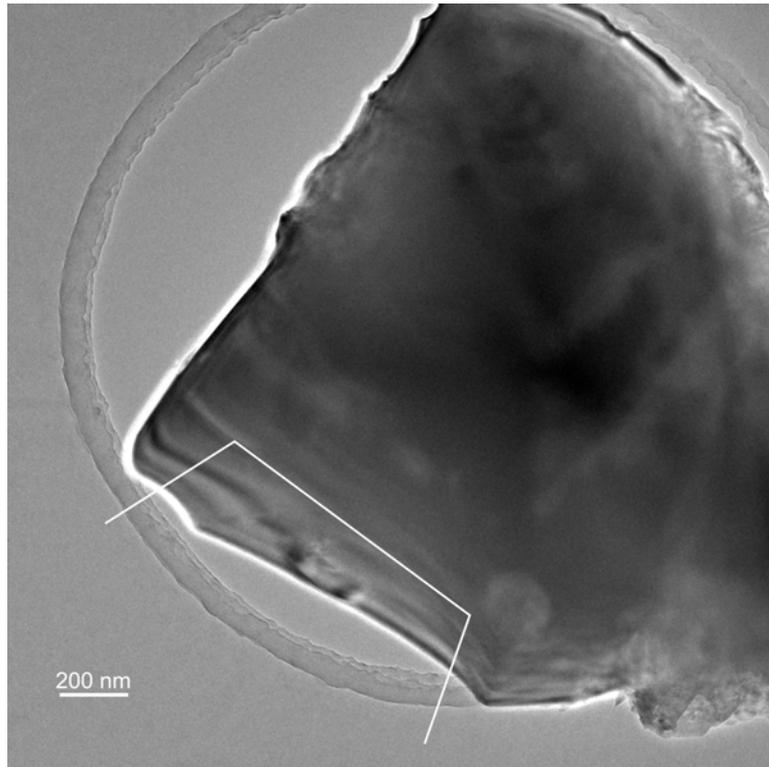


Abbildung 50: Hellfeldabbildung des untersuchten $\text{Sc}_3\text{Sc}_{0.32}\text{Al}_{0.68}\text{O}_6$ -Kristalls in $[202\bar{1}]$.

Datensatz Nr.	Auslesekoord.		Lauzentrum				Dicke [Å]	Skalen- faktor	R- Wert
	<i>x</i>	<i>y</i>	<i>h</i>	<i>k</i>	<i>i</i>	<i>l</i>			
1	0	0	-0.9(2)	0.9(2)	0.0(2)	-0.4(2)	121(20)	1.13(9)	13.6 %
2	-10	0	-0.8(2)	0.8(2)	0.0(2)	-0.41(18)	113(8)	1.01(4)	13.5 %
3	10	0	-0.6(2)	0.8(2)	-0.2(2)	-0.52(19)	111(9)	1.02(5)	12.9 %
4	0	-10	-0.7(2)	0.8(2)	-0.1(2)	-0.40(19)	110(8)	1.02(5)	13.2 %
5	0	10	-0.7(2)	0.8(2)	-0.1(2)	-0.49(13)	113(10)	1.02(5)	14.5 %
6	5	5	-0.7(2)	0.9(2)	-0.2(2)	-0.54(18)	115(9)	1.01(5)	14.6 %
7	-5	-5	-0.8(2)	0.8(2)	0.0(2)	-0.37(18)	112(9)	1.02(5)	13.0 %

Datens. Nr.	Auslesekd.		Lauzentrum				Dicke [Å]	Skalen- faktor	R- Wert
	<i>x</i>	<i>y</i>	<i>h</i>	<i>k</i>	<i>i</i>	<i>l</i>			
1	0	0	-0.20(18)	-0.55(18)	0.75(18)	-0.47(15)	128(9)	1.01(5)	13.8%
2	-10	0	-0.15(19)	-0.58(19)	0.73(19)	-0.44(16)	127(10)	1.02(5)	13.9%
3	10	0	-0.35(18)	-0.48(18)	0.83(18)	-0.59(15)	126(9)	1.01(4)	13.7%
4	0	-10	-0.29(19)	-0.54(20)	0.83(20)	-0.56(20)	128(10)	1.01(5)	14.6%
5	0	10	-0.22(17)	-0.46(18)	0.68(18)	-0.45(15)	131(9)	1.02(5)	14.1%
6	5	5	-0.33(18)	-0.47(18)	0.80(18)	-0.56(15)	129(9)	1.01(5)	14.2%
7	-5	-5	-0.13(18)	-0.65(19)	0.78(2)	-0.46(15)	127(10)	1.02(5)	14.2%

Datenstz. Nr.	Auslesekd.		Lauzentrum				Dicke [Å]	Skalen- faktor	R- Wert
	<i>x</i>	<i>y</i>	<i>h</i>	<i>k</i>	<i>i</i>	<i>l</i>			
1	0	0	0.64(8)	-0.80(17)	0.16(13)	0.98(16)	225(16)	0.91(6)	19.1%
2	-10	0	0.54(8)	-0.64(17)	0.10(13)	0.90(16)	224(17)	0.90(6)	20.0%
3	10	0	0.72(8)	-0.94(17)	0.22(13)	1.01(14)	230(16)	0.92(6)	18.2%
4	0	-10	0.67(8)	-0.85(17)	0.18(13)	0.98(15)	231(17)	0.93(6)	17.8%
5	0	10	0.63(7)	-0.82(17)	0.19(12)	0.90(15)	225(15)	0.90(6)	19.1%
6	5	5	0.67(8)	-0.88(18)	0.21(13)	0.93(16)	224(17)	0.87(6)	18.6%
7	-5	-5	0.60(7)	-0.74(16)	0.14(12)	0.92(14)	225(15)	0.90(6)	19.1%

Tabellen 30, 31 und 32: Berechnete Werte nach der Vorverfeinerung mit Intensitätsdaten in einer $[01\bar{1}2]$ -, $[10\bar{1}2]$ - und der $[20\bar{2}1]$ -Zonenachse von $\text{Sc}_3\text{Sc}_{0.32}\text{Al}_{0.68}\text{O}_6$. Tabelle 30 gehört zur Abbildung 48a, Tabelle 31 zur Abbildung 48b und Tabelle 32 zur Abbildung 48c.

Die in den Tabellen 30 bis 32 angegebenen R-Werte von 13-20 % sind sehr viel größer als die in den übrigen Zonenachsen von $\text{Sc}_3\text{Sc}_{0.32}\text{Al}_{0.68}\text{O}_6$. Damit ist eine größere Differenz zwischen den mit MSLS berechneten und den experimentellen Beugungsintensitäten in diesen Zonenachsen vorhanden. Demnach sollte bei der Strukturverfeinerung der variablen strukturellen Parameter eine Verschiebung eintreten. Gerade an den Ergebnissen in diesen beiden Zonenachsen erkennt man trotz der sehr viel schlechteren R-Werte im Vergleich zu denen in den übrigen hier gezeigten Zonenachsen, dass solche Datensätze zur Strukturverfeinerung verwendet werden können, da die übrigen Kriterien zur Verwendung erfüllt sind.

3.3.2 Strukturverfeinerung von $\text{Sc}_3\text{Sc}_{0.32}\text{Al}_{0.68}\text{O}_6$ mit den Datensätzen aus der Vorverfeinerung

MSLS bietet die Möglichkeit bis zu 14 Datensätze für eine Strukturverfeinerung zu verwenden. Alle in den Kapiteln 3.3.1.1 bis 3.3.1.5 beschriebenen zentral ausgelesenen Datensätze wurden für die Strukturverfeinerung verwendet. Insgesamt standen damit 1253 Reflexe für die Strukturverfeinerung zur Verfügung. Die zonenachsenspezifischen Ergebnisse der Strukturverfeinerung (Kristalldicke, Lauzentrum und Skalenfaktor) sind in Tabelle 33 dargestellt. Die Atomparameter der verfeinerten Struktur sind in Tabelle 34 gezeigt.

Dtns. Nr.	Zonen- achse	Center of Laue circle				Dicke [Å]	Skalen- faktor	R- Wert
		<i>h</i>	<i>k</i>	<i>i</i>	<i>l</i>			
1	[0001]	-1.21(16)	0.47(8)	0.74(12)	0	74(5)	1.049(12)	0.1%
2	[0001]	0.69(8)	0.69(7)	0.00(8)	0	147(6)	1.020(10)	0.05%
3	[0001]	0.076(12)	-0.125(13)	0.049(13)	0	174(4)	0.989(14)	0.2%
4	[2 $\bar{1}$ $\bar{1}$ 0]	0	0.0(2)	0.0(2)	-0.9(2)	64(4)	1.065(14)	0.6%
5	[2 $\bar{1}$ $\bar{1}$ 0]	0	-0.65(16)	0.65(16)	-0.83(16)	88(4)	1.030(12)	0.3%
6	[2 $\bar{1}$ $\bar{1}$ 0]	0	-0.64(10)	0.64(10)	-0.52(8)	132(4)	0.999(11)	0.3%
7	[10 $\bar{1}$ 1]	-1.11(19)	2.2(5)	-1.1(3)	-0.02(20)	61(4)	1.053(16)	1.4%
8	[10 $\bar{1}$ 1]	1.38(14)	-2.1(3)	0.7(2)	-0.71(13)	81(3)	1.033(16)	1.2%
9	[10 $\bar{1}$ 1]	0.88(8)	-1.11(18)	0.23(13)	-0.64(9)	97(3)	1.018(15)	0.7%
10	[10 $\bar{1}$ 0]	-0.61(11)	1.2(2)	-0.59(16)	-1.08(8)	123(3)	0.998(8)	0.2%
11	[10 $\bar{1}$ 0]	-0.06(9)	0.11(17)	-0.05(13)	-0.96(7)	135(3)	0.949(7)	0.4%
12	[01 $\bar{1}$ 2]	-0.81(9)	0.82(7)	0.01(8)	-0.41(5)	106(4)	1.18(2)	1.7%
13	[101 $\bar{2}$]	-0.18(5)	-0.56(6)	0.74(6)	-0.46(4)	120(3)	1.022(17)	1.0%
14	[20 $\bar{2}$ $\bar{1}$]	0.787(16)	-0.94(3)	0.15(2)	1.28(3)	201(3)	1.003(11)	5.7%
Gesamt R-Wert 1253 Reflexe = 0.7 %								

Tabelle 33: Berechnete datensatzspezifische Werte nach der Strukturverfeinerung von den verwendeten Zonenachsen von $\text{Sc}_3\text{Sc}_{0.32}\text{Al}_{0.68}\text{O}_6$.

Atom	<i>x/a</i>	<i>y/a</i>	<i>z/c</i>	Debye-Waller Faktor	Besetzungs- faktor	Wykhoff Pos.
Sc1	0.0410(3)	0.2666(3)	0.2020(3)	0.54(3)	1	18f
Sc2	0.0	0.0	0.5	1.0	0.409(7)	3b
Sc3	0.0	0.0	0.0	1.0	0.226(8)	3a
Al1	0.0	0.0	0.5	1.0	0.591(12)	3b
Al2	0.0	0.0	0.0	1.0	0.774(12)	3a
O1	0.0248(10)	0.2026(11)	0.4023(9)	2.71(15)	1	18f
O2	0.1824(10)	0.1510(10)	0.1170(8)	2.44(14)	1	18f

Tabelle 34: Verfeinerte Kristallstruktur von $\text{Sc}_3\text{Sc}_{0.32}\text{Al}_{0.68}\text{O}_6$.

Die berechneten Fehler der datensatzspezifischen Ergebnisse (s. Tabelle 33) sind im Vergleich zu den Fehlern der Vorverfeinerungen (s. Tabellen 19 bis 32) kleiner geworden, was ein Hinweis auf eine gelungene Strukturverfeinerung ist. Insbesondere die R-Werte der Datensätze 12 bis 14 in Tabelle 33 (Zonenachsen [01 $\bar{1}$ 2], [101 $\bar{2}$] und [20 $\bar{2}$ $\bar{1}$]) sind nach der Strukturverfeinerung von 13.6 % auf 1.7 % ([01 $\bar{1}$ 2]), von 13.8 % auf 1.0 % ([101 $\bar{2}$]) und von 19.1 % auf 5.7 % ([20 $\bar{2}$ $\bar{1}$]) zurückgegangen. Ein ausgezeichnete R-Wert von 0.7 % über alle Reflexe zeigt den Erfolg der

Strukturverfeinerung an. Die resultierenden Bindungsabstände und –winkel (s. Tabelle 35) liegen im üblichen Bereich für Oxide von Sc und Al (s. a. Kapitel 3.5).

Atom 1	Atom 2	Abstand [Å]
Sc1	O1	2.105(9)
	O1	2.175(6)
	O1	2.202(7)
	O2	2.091(6)
	O2	2.136(6)
	O2	2.139(8)
Al1 bzw. Sc2	O1	1.953(7)
Al2 bzw. Sc3	O2	1.903(6)

Tabelle 35: Gefundene Bindungsabstände im $\text{Sc}_3\text{Sc}_{0.32}\text{Al}_{0.68}\text{O}_6$.

Die Tatsache, dass der gefundene Bindungsabstand zwischen den O2- und den Al2/Sc3-Atome kleiner ist als der zwischen den O1- und den Al1/Sc2-Atome geht einher mit dem Besetzungsgrad auf diesen Positionen. Auf der Al2/Sc3-Position ist mit 77 % ein Al-Überschuss vorhanden. Mit gut 59 % ist auf der Al1/Sc2-Position ein geringerer Al-Überschuss vorhanden, weshalb die Kationen auf dieser Position im Mittel mehr Platz beanspruchen.

Die komplexe Struktur von $\text{Sc}_3\text{Sc}_{0.32}\text{Al}_{0.68}\text{O}_6$ lässt sich einfacher verstehen, wenn zuerst die nächste Umgebung der einzelnen Kationen betrachtet wird.

3.3.2.1 Sc(1)-Koordination

Jedes Sc(1)-Atom ist oktaedrisch von sechs Sauerstoffionen umgeben. Die entsprechenden Bindungslängen sind ohne die dazugehörigen Fehler, die in Tabelle 35 gezeigt sind, in Abbildung 30 dargestellt. Dieses Oktaeder ist stark verzerrt, was auch an den Bindungswinkeln, die teilweise in Abbildung 30 gezeigt sind, deutlich wird. Der besseren Übersicht wegen ist auf die Darstellung der restlichen neun Bindungswinkel des Oktaeders innerhalb der Graphik verzichtet worden. Sie sind am Ende dieser Seite beschrieben. Zur besseren Unterscheidung der Sauerstoffatome sind die entsprechenden Sc-O(X)-Bindungsabstände (in [Å]) den jeweiligen Sauerstoffatomen nachgestellt.

$\text{O}(1)[2.175]\text{-Sc-O}(2)[2.139] = 97.83(23)^\circ$, $\text{O}(1)[2.175]\text{-Sc-O}(1)[2.202] = 73.11(24)^\circ$,
 $\text{O}(1)[2.175]\text{-Sc-O}(2)[2.105] = 142.47(48)^\circ$, $\text{O}(2)[2.136]\text{-Sc-O}(2)[2.139] = 152.59(33)^\circ$,
 $\text{O}(1)[2.136]\text{-Sc-O}(2)[2.091] = 80.03(28)^\circ$, $\text{O}(2)[2.136]\text{-Sc-O}(1)[2.105] = 91.51(33)^\circ$,
 $\text{O}(2)[2.091]\text{-Sc-O}(1)[2.105] = 103.05(34)^\circ$, $\text{O}(2)[2.091]\text{-Sc-O}(1)[2.202] = 155.80(34)^\circ$,
 und $\text{O}(2)[2.139]\text{-Sc-O}(1)[2.202] = 81.23(35)^\circ$.

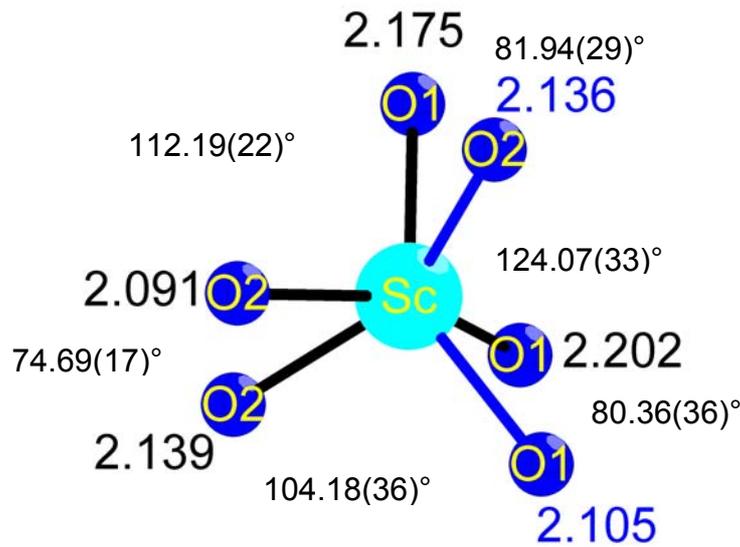


Abbildung 51: $\text{Sc}(1)\text{O}_6$ -Oktaeder mit Bindungslängen und sechs Bindungswinkel von $\text{Sc}_3\text{Sc}_{0.32}\text{Al}_{0.68}\text{O}_6$. Abstandsangaben in [Å].

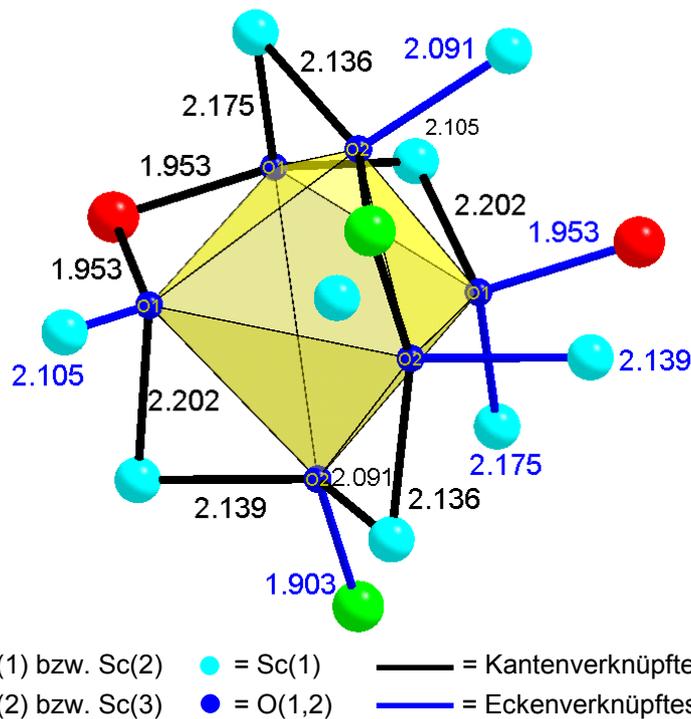


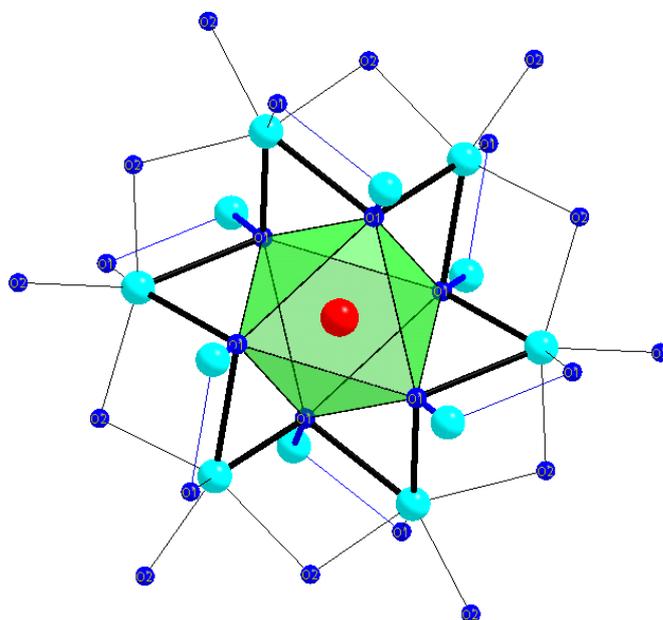
Abbildung 52: Nächste Umgebung des $\text{Sc}(1)\text{O}_6$ -Oktaeders. Alle Abstandsangaben in [Å].

In Abbildung 52 ist die nächste Sphäre des $\text{Sc}(1)\text{O}_6$ -Oktaeders gezeigt. In dieser Sphäre befinden sich drei eckenverknüpfte $\text{Sc}(1)\text{O}_6$ -Oktaeder und jeweils ein eckenverknüpfter $\text{Al}(1,2)/\text{Sc}(2,3)\text{O}_6$ -Oktaeder. Weiterhin sind vier kantenverknüpfte $\text{Sc}(1)\text{O}_6$ -Oktaeder und jeweils ein kantenverknüpfter $\text{Al}(1,2)/\text{Sc}(2,3)\text{O}_6$ -Oktaeder in dieser Sphäre vorhanden. Die Anordnung der unterschiedlichen Oktaeder ist in Abbildung 52

dargestellt. Hier ist bereits die Komplexität der gesamten Struktur ablesbar, da Oktaeder der gleichen Art verschiedene Verknüpfungsmuster besitzen.

3.3.2.2 Al(1)/Sc(2)-Koordination

Wie bei der Sc(1)-Position sind auch die Al(1)- bzw. die Sc(2)-Kationen oktaedrisch von sechs Sauerstoffatomen umgeben. Aufgrund der Punktlagensymmetrie der 3b-Position handelt es sich im Gegensatz zum verzerrten Sc(1)O₆-Oktaeder um einen regelmäßigen Oktaeder mit dem Bindungsabstand von 1.953(7) Å. In Abbildung 53 ist die nächste Umgebung eines Al(1)/Sc(2)O₆-Oktaeders dargestellt.



Kat- und Anionen wie in Abbildung Nr. 52

— = Kantenverknüpftes Oktaeder

— = Eckenverknüpftes Oktaeder

— = Verknüpfung an einem kantenverknüpften Oktaeder

— = Verknüpfung an einem eckenverknüpften Oktaeder

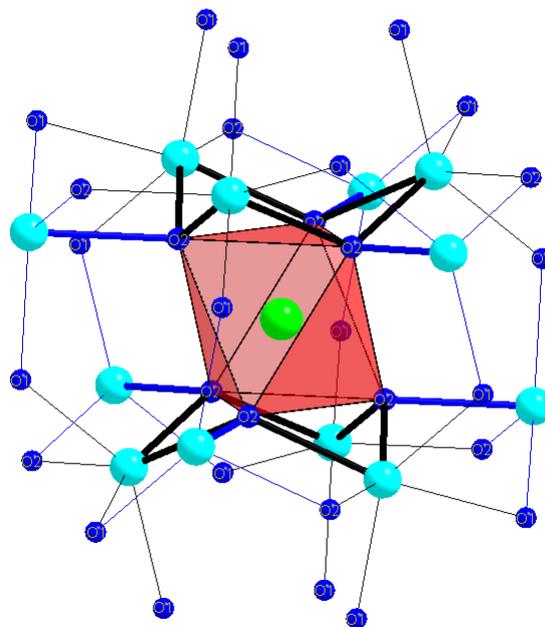
Abbildung 53: Sphäre um das Al(1)/Sc(2)O₆-Oktaeder im Sc₃Sc_{0.32}Al_{0.68}O₆.

Um dieses Al(1)/Sc(2)O₆-Oktaeder befinden sich nur Sc(1)O₆-Oktaeder. Davon sind sechs Oktaeder über Kanten und sechs über Ecken verknüpft. Jedes der sechs Sauerstoffatome des Al(1)/Sc(2)O₆-Oktaeders verbindet einen eckenverknüpften Sc(1)O₆-Oktaeder. Dabei weisen drei dieser Sc(1)O₆-Oktaeder nach vorne d.h. aus der Papierebene heraus und drei in die Papierebene hinein. Man kann dieses als eine zweidimensionale Ausdehnung der eckenverknüpften Sc(1)O₆-Oktaeder bezeichnen.

Diese zum zentralen $\text{Al}(1)/\text{Sc}(2)\text{O}_6$ -Oktaeder eckenverknüpften $\text{Sc}(1)\text{O}_6$ -Oktaeder besitzen mit denen zum zentralen $\text{Al}(1)/\text{Sc}(2)\text{O}_6$ -Oktaeder kantenverknüpften $\text{Sc}(1)\text{O}_6$ -Oktaeder eine gemeinsame Kante. Die zum $\text{Al}(1)/\text{Sc}(2)\text{O}_6$ -Oktaeder kantenverknüpften $\text{Sc}(1)\text{O}_6$ -Oktaeder sind ihrerseits mit ihren Nachbarn jeweils über ein weiteres nicht zum $\text{Al}(1)/\text{Sc}(2)\text{O}_6$ -Oktaeder zugehöriges Sauerstoffanion miteinander kantenverknüpft. Somit gehören alle sechs Sauerstoffanion des $\text{Al}(1)/\text{Sc}(2)\text{O}_6$ -Oktaeders sowohl zu einer Ecken- als auch zu einer Kantenverknüpfung (s. Abbildung 53).

3.3.2.3 $\text{Al}(2)/\text{Sc}(3)$ -Koordination

Im Prinzip gelten dieselben Aussagen wie bei der 3b-Position von $\text{Al}(1)/\text{Sc}(2)$, wenngleich die Sphäre um den regelmäßigen $\text{Al}(2)/\text{Sc}(3)\text{O}_6$ -Oktaeder im Rahmen der gleichen Punktlagensymmetrie wie bei der 3b-Position geringfügig variiert.



Legende wie in Abbildung 52 und Abbildung 53

Abbildung 54: Sphäre um das $\text{Al}(2)/\text{Sc}(3)\text{O}_6$ -Oktaeder im $\text{Sc}_3\text{Sc}_{0.32}\text{Al}_{0.68}\text{O}_6$.

Wie auch bei dem $\text{Al}(1)/\text{Sc}(2)\text{O}_6$ -Oktaeder sind hier ebenfalls sechs ecken- und sechs kantenverknüpfte $\text{Sc}(1)\text{O}_6$ -Oktaeder mit dem zentralen $\text{Al}(2)/\text{Sc}(3)\text{O}_6$ -Oktaeder verbunden. Während auf der 3b-Position die kantenverknüpften $\text{Sc}(1)\text{O}_6$ -Oktaeder in alle Raumrichtungen weisen, so reichen die $\text{Al}(2)/\text{Sc}(3)\text{O}_6$ -Oktaeder in Abbildung 54 nach oben und nach unten. D.h. sie bilden einen zweidimensionalen Strang innerhalb der Struktur. Die zum zentralen $\text{Al}(2)/\text{Sc}(3)\text{O}_6$ -Oktaeder eckenverknüpften $\text{Sc}(1)\text{O}_6$ -

Oktaeder weisen auf dieser Lage im Gegensatz zu der 3b-Lage in alle drei Raumrichtungen. Daraus lässt sich schließen, dass sich die ecken- und kantenverknüpften Sc(1)O₆-Oktaeder konvers zu denen der 3b-Position verhalten (s. Abbildung 54).

3.3.2.4 Elementarzelle von $\text{Sc}_3\text{Sc}_{0.32}\text{Al}_{0.68}\text{O}_6$

Nachdem die einzelnen Sphären der unterschiedlichen Kationen dargestellt worden sind, wird nun abschließend die komplexe Elementarzelle von $\text{Sc}_3\text{Sc}_{0.32}\text{Al}_{0.68}\text{O}_6$ gezeigt, so dass ein Gesamtüberblick gegeben wird.

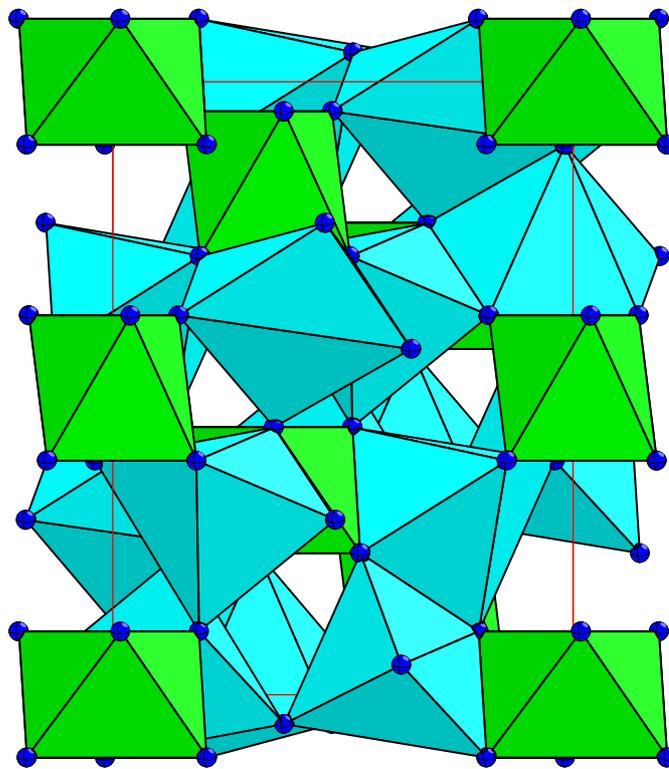


Abbildung 55: Die Elementarzelle von $\text{Sc}_3\text{Sc}_{0.32}\text{Al}_{0.68}\text{O}_6$ entlang $[2\bar{1}\bar{1}0]$. Die grünen Oktaeder sind die regelmäßigen $\text{Al}(1,2)/\text{Sc}(2,3)$ -O₆-Oktaeder. Die hellblauen Oktaeder sind die der Sc(1)-Position. Die rote Linie kennzeichnet die Elementarzellenbegrenzung.

3.4 Strukturverfeinerung von ScGaO₃

ScGaO₃ kristallisiert in der Raumgruppe *Pba2* (Nr. 32) und ist isostrukturell zu dem quaternären Oxid In₂Fe₂Ga₂O₉ [47]. Es ist bisher das einzig bekannte ternäre Oxid, welches in dieser Struktur kristallisiert. Eine Besonderheit dieser Verbindung liegt in den stark unterschiedlich langen Achsen. Mit 3.1254 Å ist die *c*-Achse die kürzeste der Verbindung, die *b*-Achse beträgt 7.1859 Å und die *a*-Achse ist mit 19.0040 Å die längste Achse von ScGaO₃. Aufgrund der langen *a*-Achse ist es u.a. nicht möglich, unter konvergenten Einstrahlbedingungen erzeugte Beugungsbilder ohne Überlappung der einzelnen Beugungsscheiben der [001]-Richtung aufzunehmen. Infolge dessen musste in den Richtungen, bei denen die Reflexe des Typs [hkl] mit $h \neq 0$ erscheinen, auf die konventionelle Methode zurückgegriffen werden. In allen anderen Richtungen wurde die hier neu entwickelte Methode angewendet. Ausgehend von den Atompositionen in Tabelle 36 wurde bei den Beugungsbildern eine Vorverfeinerung durchgeführt.

Atom	<i>x/a</i>	<i>y/b</i>	<i>z/c</i>	Debye-Waller Faktor	Besetzungsfaktor	Wyk hoff
Sc(1)	0.6344	0.4632	0.330	2	1	4c
Sc(2)	0.7552	0.2232	0.816	2	0.6	4c
Ga(1)	0.4973	0.2754	0.824	2	1	4c
Ga(2)	0.6159	0.9818	0.313	2	1	4c
Ga(3)	0.7552	0.2232	0.816	2	0.4	4c
O(1)	0.1805	0.325	0.294	2	1	4c
O(2)	0.4319	0.473	0.844	2	1	4c
O(3)	0.2096	0.034	0.838	2	1	4c
O(4)	0.3212	0.233	0.355	2	1	4c
O(5)	0.0419	0.270	0.327	2	1	4c
O(6)	0.4273	0.077	0.847	2	1	4c

Tabelle 36: Startwerte der Strukturverfeinerung von ScGaO₃ nach [54].

Die Atompositionen wurden von dem quaternären Oxid In₂Fe₂Ga₂O₉ abgeleitet [54] (s. Tabelle 36). Dabei werden die reinen Indium-Positionen im In₂Fe₂Ga₂O₉ im ScGaO₃ mit Scandium belegt (Sc(1) in Tabelle 36). Die im In₂Fe₂Ga₂O₉ mit drei Kationen (In, Fe und Ga) besetzte Lage wird im ScGaO₃ mit Scandium (Sc2) und Gallium (Ga3) okkupiert. Die beiden verbleibenden Positionen, die in In₂Fe₂Ga₂O₉ jeweils von Eisen

und Gallium belegt sind, werden im ScGaO_3 von Gallium (Ga1) und (Ga2) besetzt. Es befinden sich damit acht Formeleinheiten in der Elementarzelle.

3.4.1 Die zur Strukturverfeinerung verwendeten Zonenachsen von ScGaO_3

Wie beim $\text{Sc}_3\text{Sc}_{0.32}\text{Al}_{0.68}\text{O}_6$ wurde auch beim ScGaO_3 darauf geachtet, dass die gesamten Beugungsaufnahmen ein dreidimensionales Gesamtbild erzeugen. Das heißt im einzelnen, dass neben den Hauptzonenachsen [100], [010] und [001] auch eine Zwischenachse [101] und eine vom Typ [120] zur Strukturverfeinerung herangezogen wurden.

Die unter konvergenten Einstrahlbedingungen erzeugten Beugungsbilder wurden mit einer $30\ \mu\text{m}$ großen C2-Kondensorblende erstellt. Für einen kleinen Elektronenstrahldurchmesser wurde eine „spot size“ von 11 gewählt. Die mit einem parallelen Elektronenstrahl aufgenommenen Punktbeugungsbilder wurden mit einer $10\ \mu\text{m}$ großen C2-Kondensorblende und einer „spot size“ von 9 aufgezeichnet.

Es werden im folgenden nur die Anzahl der Reflexe von den zentral ausgelesenen Datensätzen genannt, da nur diese Datensätze zur Strukturverfeinerung herangezogen wurden.

3.4.1.1 Zonenachse [100] von ScGaO_3

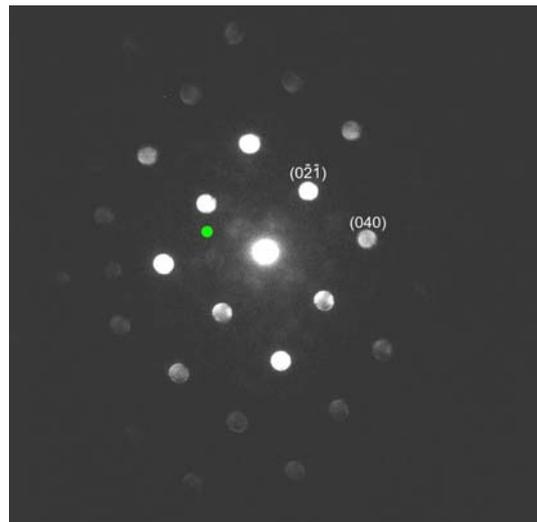
Die [100] Zonenachse kann aufgrund des ausreichend großen Abstandes der (0k0)-, (00l)- und der (0kl)-Reflexe in den Beugungsbildern unter konvergenten Einstrahlbedingungen erstellt werden. Es wurden drei Beugungsbilder mit unterschiedlichen Kristalldicken zur Strukturverfeinerung herangezogen.

Der zentral ausgelesene Datensatz des Beugungsbildes in der Abbildung 56a enthält 80, der zur Abbildung 56b 94 und der zur Abbildung 56c ebenfalls 94 verwertbare Reflexe.

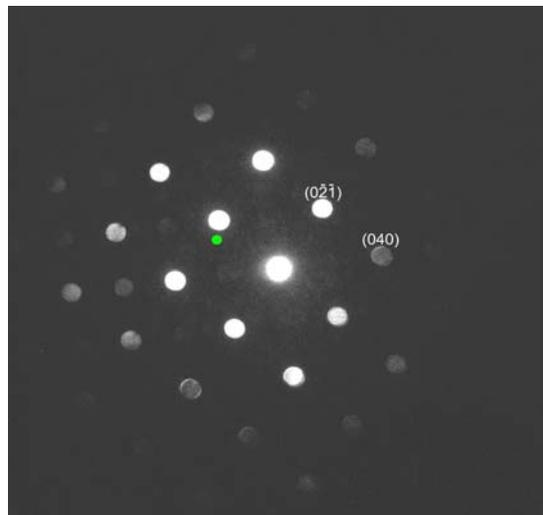
Der für diese Zonenachse verwendete Kristall ist in Abbildung 57 gezeigt. Die weißen Rahmen in Abbildung 57 kennzeichnen den Bereich, der für die Aufnahmen der Beugungsbilder herangezogen wurde. Das Gebiet des Steges musste dabei ausgespart werden, da der amorphe Kohlenstoff die Beugungsintensitäten verfälscht.

Die verfeinerten Werte der Skalenfaktoren und die der Kristalldicken konvergieren zu den selben Werten. Die Resultate der Lauezentren werden mit ansteigender Kristalldicke exakter (s. Tabellen 37 bis 39). Die *l*-Koordinate der Lauezentren reproduziert die verschiedenen Einstrahlrichtungen der unterschiedlichen Datensätze trotz der mit $78(3)\ \text{\AA}$ geringsten Probendicke in dieser Zonenachse aber auch schon bei den Ergebnissen der Datensätze in der Tabelle 37 ausreichend gut wider.

a)



b)



c)

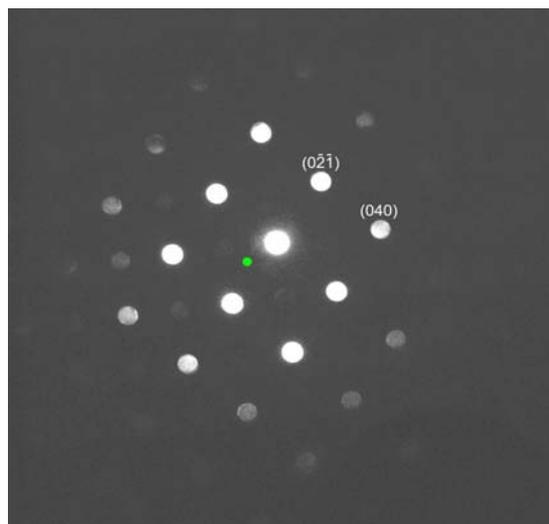
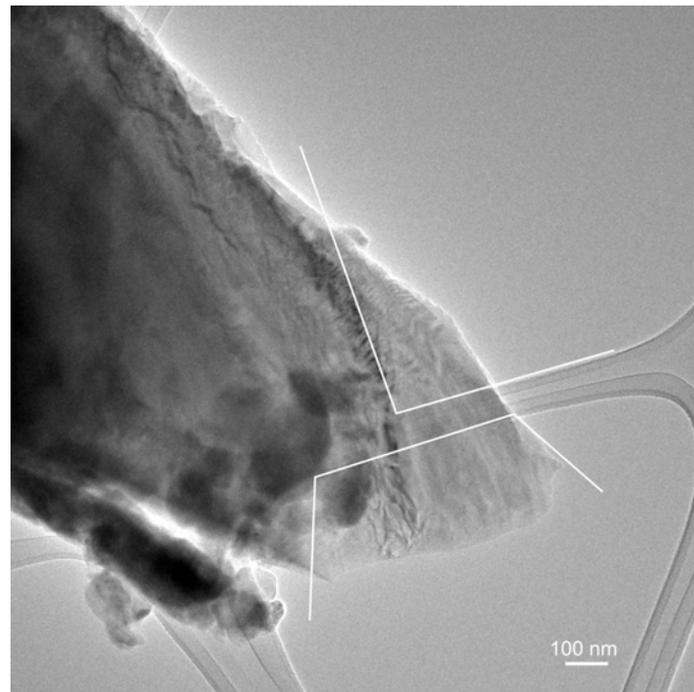


Abbildung 56: Drei Beugungsbilder von ScGaO_3 in $[100]$. Kameralänge: 780 mm. Grüner Kreis: Verfeinertes Lauezentrum des zentral ausgelesenen Datensatzes (Auslesekoordinaten x und $y = 0$). Größe des grünen Kreises = Fehler in der Bestimmung des Lauezentrums (s. Tabellen 37 bis 39). Aufnahmezeit: 1 s.

Abbildung 57: Hellfeldabbildung des untersuchten ScGaO₃-Kristalls in [100].

Datensatz Nr.	Auslesekoord.		Lauzentrum			Dicke [Å]	Skalen- faktor	R- Wert
	<i>x</i>	<i>y</i>	<i>h</i>	<i>k</i>	<i>l</i>			
1	0	0	0	-2.2(2)	-0.54(5)	78(3)	0.977(19)	2.5%
2	-10	0	0	-1.72(20)	-0.42(5)	81(3)	0.969(20)	3.0%
3	10	0	0	-2.2(2)	-0.58(5)	79(3)	0.977(20)	2.7%
4	0	-10	0	-2.22(20)	-0.46(5)	76(2)	0.974(17)	2.5%
5	0	10	0	-2.0(2)	-0.55(5)	78(3)	0.976(19)	3.0%
6	5	5	0	-2.2(2)	-0.59(5)	78(3)	0.973(19)	2.8%
7	-5	-5	0	-1.89(19)	-0.43(5)	78(3)	0.967(18)	2.7%

Datensatz Nr.	Auslesekoord.		Lauzentrum			Dicke [Å]	Skalen- faktor	R- Wert
	<i>x</i>	<i>y</i>	<i>h</i>	<i>k</i>	<i>l</i>			
1	0	0	0	-2.07(19)	-0.76(4)	112(3)	0.96(2)	3.8%
2	-10	0	0	-1.75(15)	-0.73(4)	117(3)	0.96(2)	3.7%
3	10	0	0	-2.48(18)	-0.80(4)	110(3)	0.97(2)	3.0%
4	0	-10	0	-1.98(18)	-0.70(4)	114(3)	0.97(2)	3.9%
5	0	10	0	-2.4(2)	-0.87(6)	105(3)	0.98(3)	3.9%
6	5	5	0	-2.42(20)	-0.82(5)	107(3)	0.97(2)	3.5%
7	-5	-5	0	-1.80(16)	-0.72(4)	115(3)	0.98(2)	3.7%

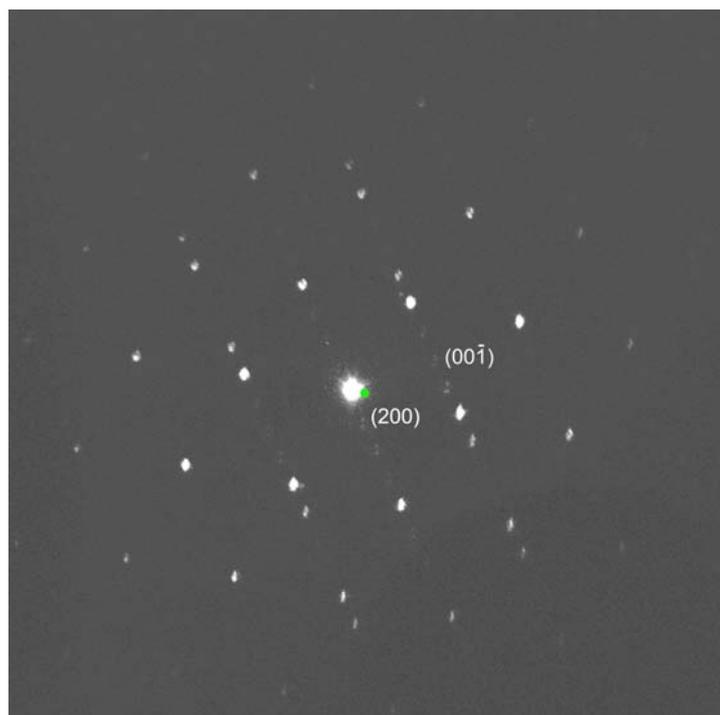
Datensatz Nr.	Auslesekoord.		Lauezentrum			Dicke [Å]	Skalen- faktor	R- Wert
	<i>x</i>	<i>y</i>	<i>h</i>	<i>k</i>	<i>l</i>			
1	0	0	0	-0.78(6)	0.321(18)	138(2)	0.945(19)	3.1%
2	-10	0	0	-0.76(6)	0.310(17)	139(2)	0.944(19)	3.6%
3	10	0	0	-0.82(7)	0.334(18)	135(2)	0.950(20)	3.1%
4	0	-10	0	-0.79(5)	0.403(16)	142(2)	0.950(17)	2.2%
5	0	10	0	-0.79(8)	0.26(2)	130(2)	0.95(2)	3.8%
6	5	5	0	-0.79(8)	0.29(2)	132(3)	0.95(2)	3.9%
7	-5	-5	0	-0.73(6)	0.355(17)	142(2)	0.949(19)	2.6%

Tabellen 37 bis 39: Berechnete Werte einer Vorverfeinerung mit Intensitätsdaten in der [100]-Zonenachse von ScGaO₃. Tabelle 37 gehört zur Abbildung 56a, Tabelle 38 zur Abbildung 56b und Tabelle 39 zur Abbildung 56c.

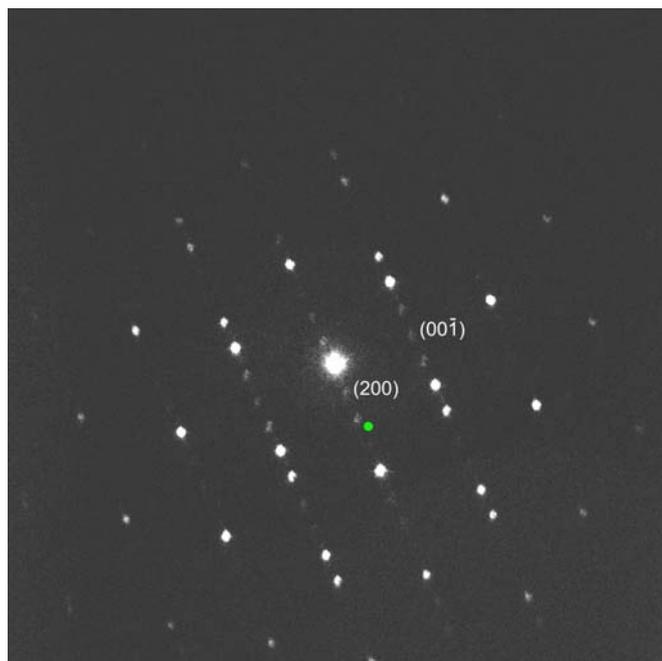
3.4.1.2 Zonenachse [010] von ScGaO₃

Die [010] Zonenachse kann infolge der relativ kleinen Abstände der (h00)- Reflexe nicht unter konvergenten Einstrahlbedingungen aufgenommen werden. Es wurden drei Punktbeugungsbilder, die von Bereichen mit unterschiedlichen Kristalldicken herrühren, zur Strukturverfeinerung herangezogen.

a)



b)



c)

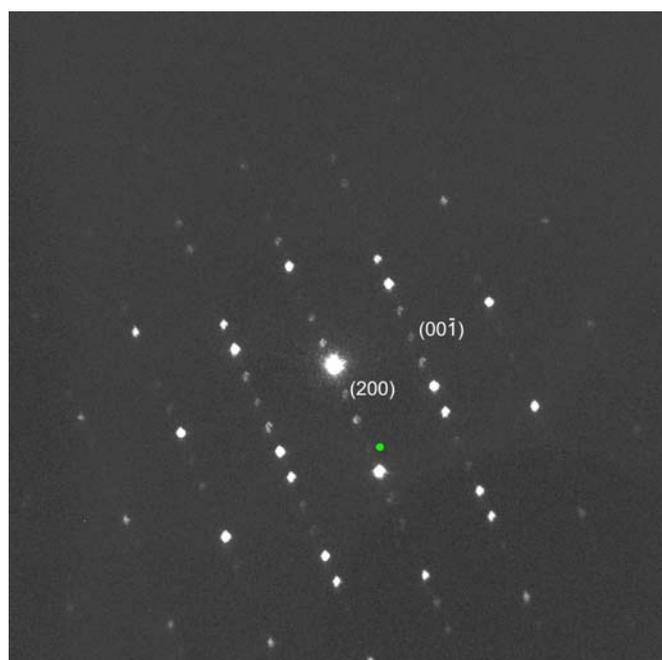


Abbildung 58: Drei Beugungsbilder von ScGaO₃ in [010]. Kameralänge: 780 mm. Grüner Kreis: Verfeinertes Lauezentrum. Größe des grünen Kreises = Fehler in der Bestimmung des Lauezentrums
Aufnahmezeit: 2 s.

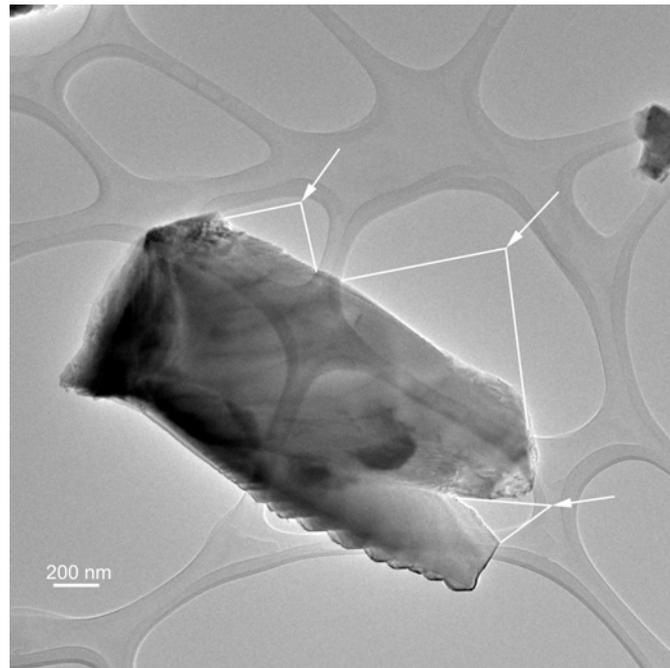


Abbildung 59: Hellfeldabbildung des untersuchten ScGaO₃-Kristalls in [010].

Abbildung Nr.	Anzahl Reflexe	Skalen- faktor	Lauenzentrum			Dicke [Å]	R- Wert
			<i>h</i>	<i>k</i>	<i>l</i>		
58a	37	1.056(29)	0.9(7)	0	-0.17	69(3)	1.1%
58b	96	1.00(2)	5.4(6)	0	-0.15(4)	107(2)	4.2 %
58c	102	0.97(3)	6.3(6)	0	-0.16(3)	136(3)	5.7 %

Tabelle 40: Ergebnisse der Vorverfeinerung mit Datensätzen in der [010]-Zonenachse von ScGaO₃.

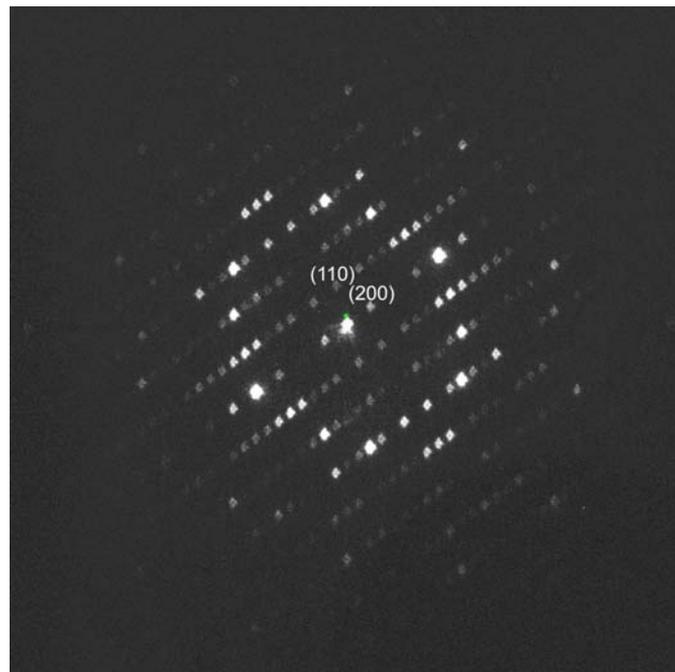
Der für diese Zonenachse verwendete Kristall ist in Abbildung 59 gezeigt. Die in dieser Abbildung mit weißen Begrenzungen gekennzeichneten Bereiche wurden für die Aufnahmen der Beugungsbilder verwendet. Wie bei den Aufnahmen in der Zonenachse [100] wurden die Bereiche des Trägermaterials bei der Aufzeichnung der Daten nicht mit beleuchtet.

Die Werte der Lauenzentren lassen den Rückschluss zu, dass sie in die richtige Richtung verschoben worden sind (Startwerte $h = k = 0$). Man erkennt in Abbildung 58a, dass das Beugungsbild fast exakt in die Zonenachse einorientiert ist. Bei den Beugungsbildern in den Abbildung 58b und c stimmen die verfeinerten Lauenzentren ebenfalls mit der Indizierung der Beugungsbilder überein. Die guten R-Werte (s. Tabelle 40) sind schließlich letzte Kriterien für die Verwendung solcher Datensätze zur Strukturverfeinerung.

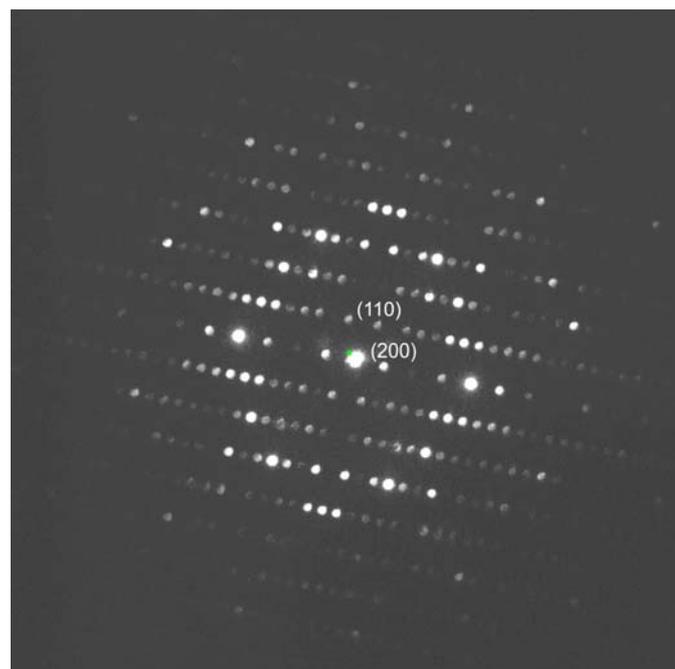
3.4.1.3 Zonenachse [001] von ScGaO₃

Die [001]-Zonenachse kann analog zur [010]-Zonenachse infolge der relativ kleinen Abstände der (h00)-Reflexe nicht unter konvergenten Einstrahlbedingungen erstellt werden. Es wurden drei Punktbeugungsbilder, die von Bereichen mit unterschiedlichen Kristalldicken stammen, zur Strukturverfeinerung herangezogen.

a)



b)



c)

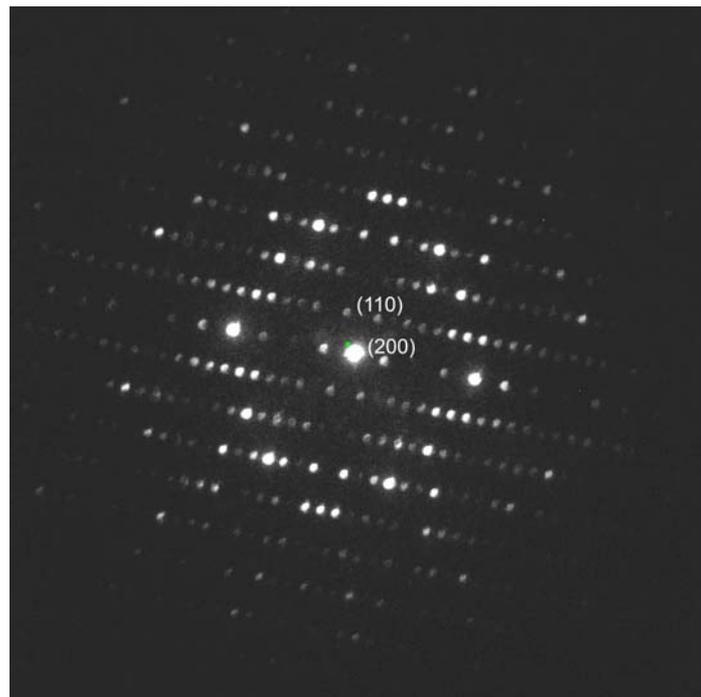


Abbildung 60: Beugungsbilder von ScGaO₃ in [001]. Kameralänge: 780 mm (a) bzw. 980 mm (b und c). Grüner Punkt: Verfeinertes Lauezentrum der Beugungsbildes. Aufnahmezeit: 1 s (a) bzw. 2 s (b und c).

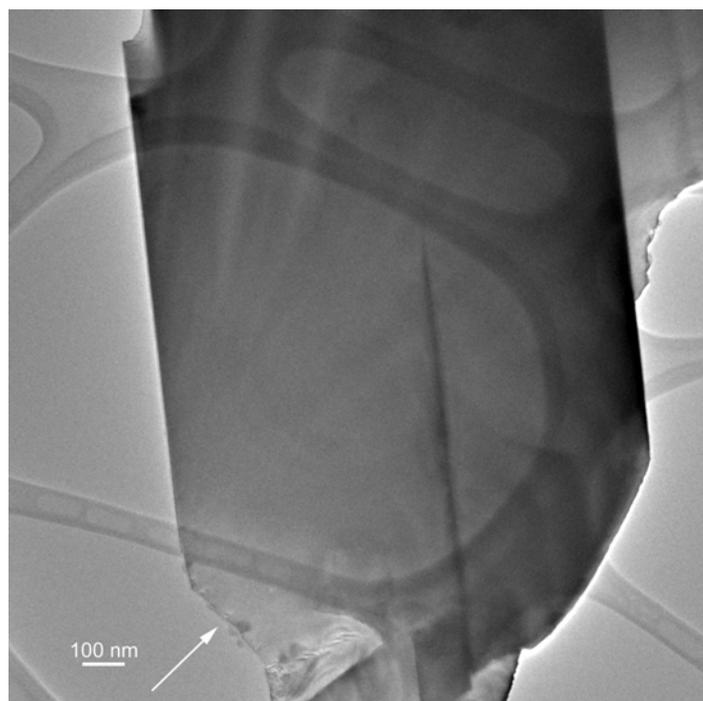
Abbildung Nr.	Anzahl Reflexe	Skalenfaktor	Lauezentrum			Dicke [Å]	R-Wert
			<i>h</i>	<i>k</i>	<i>l</i>		
60a	153	1.13(3)	0.06(4)	0.13(6)	0	181(2)	4.9 %
60b	422	0.998(16)	-0.13(4)	0.02(3)	0	185(1)	5.8 %
60c	416	1.023(18)	-0.40(3)	0.17(3)	0	189(1)	6.9 %

Tabelle 41: Ergebnisse der Vorverfeinerung mit Datensätzen in der [001]-Zonenachse von ScGaO₃.

Trotz der optimalen äusseren Form des Kristalls in Abbildung 61a, die durch eine kontinuierliche Steigerung der Proben Dicke nach rechts im Bild gegeben ist, ist es nicht gelungen, dünnere Probenbereiche als die zur Verfeinerung verwendeten 181(2) Å aufzufinden.

Auch bei den verfeinerten Werten der Kristalldicke in c-Richtung der beiden übrigen Aufnahmen in Abbildung 60b und c ist es nicht gelungen, dünnere Probenbereiche aufzufinden. Das Kristallwachstum entlang der c-Achse ist offenbar sehr viel größer als in die anderen Kristallrichtungen.

a)



b)

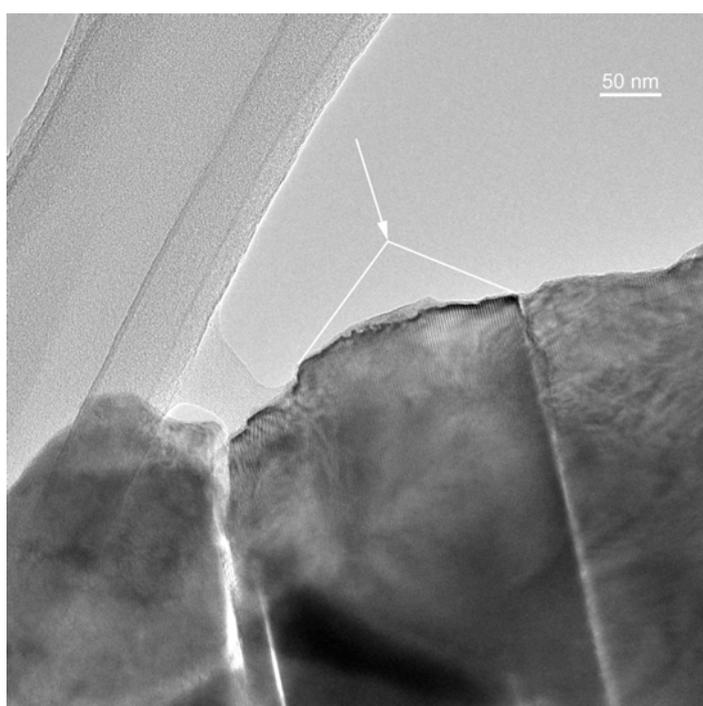
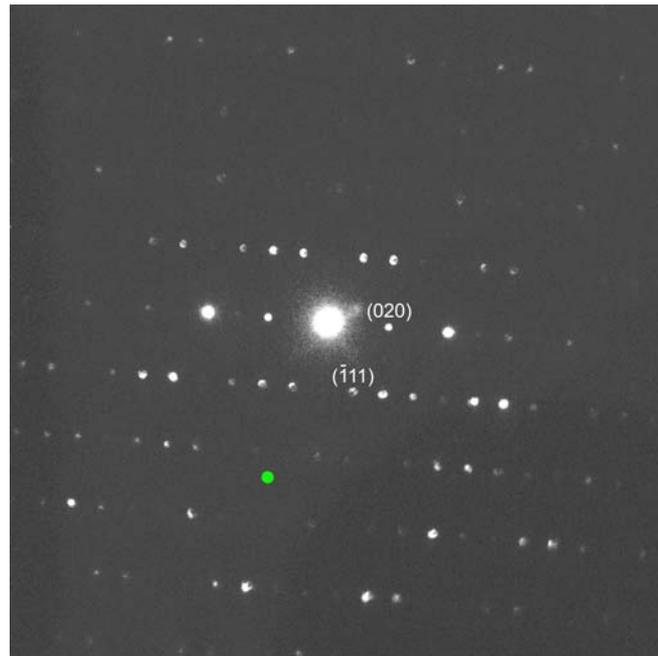


Abbildung 61: Hellfeldabbildungen des untersuchten ScGaO₃-Kristalls in [001]. Die Beugungsaufnahme der Abbildung 60a wurde von dem Kristall (a) und die Beugungsaufnahmen der Abbildung 60b und c von dem Kristall (b) erstellt.

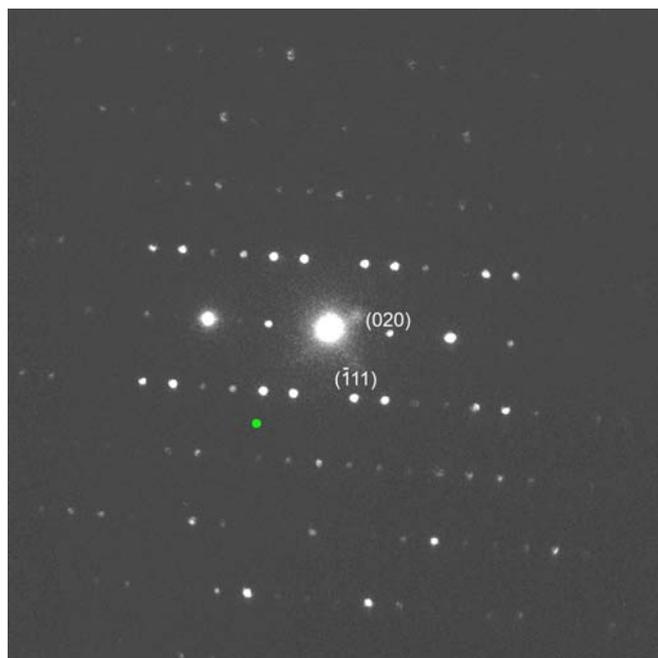
3.4.1.4 Zonenachse [101] von ScGaO₃

Auch in dieser Zonenachse ist es nicht möglich, quantitativ auswertbare unter konvergenten Einstrahlbedingungen erzeugte Beugungsbilder zu erstellen, da sich die Beugungsscheibchen überlappen. Demnach wurden wiederum Punktbeugungsbilder zur Strukturverfeinerung verwendet.

a)



b)



c)

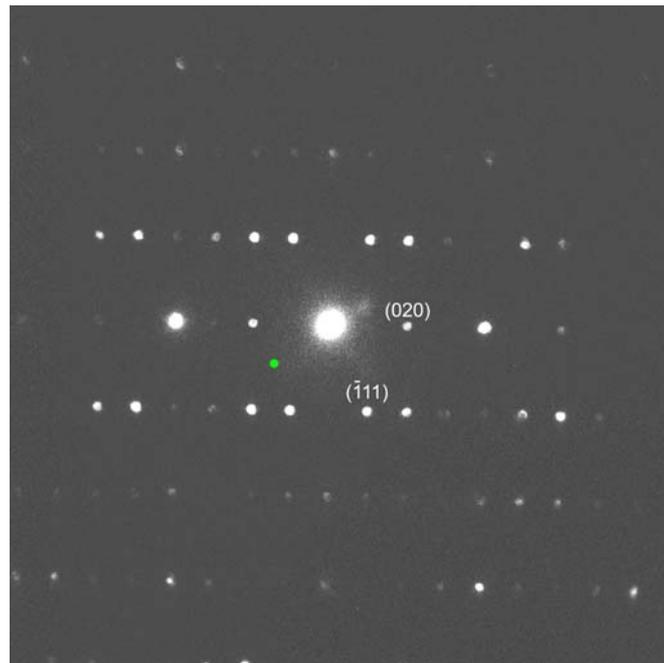


Abbildung 62: Drei Beugungsbilder von ScGaO₃ in [101]. Kameralänge: 780 mm (a und b) bzw. 980 mm (c). Grüner Kreis: Verfeinertes Lauezentrum der Beugungsbilder. Größe des grünen Kreises = Fehler in der Bestimmung des Lauezentrums (s. Tabelle 42). Aufnahmezeit: 1 s.

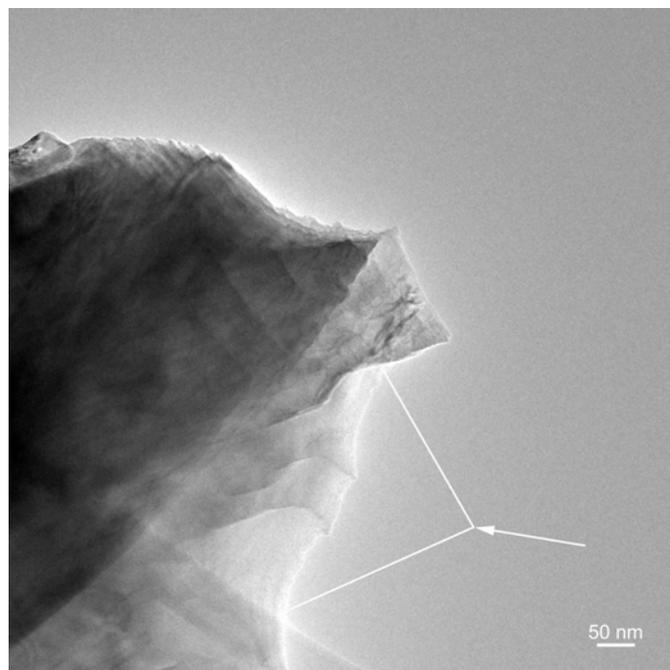


Abbildung 63: Hellfeldabbildung des untersuchten ScGaO₃-Kristalls in [101].

Abbildung Nr.	Anzahl Reflexe	Skalen- faktor	Lauzentrum			Dicke [Å]	R- Wert
			<i>h</i>	<i>k</i>	<i>l</i>		
62a	129	0.77(2)	-2.3(2)	-1.8(3)	2.3(2)	98(9)	3.3 %
62b	116	0.813(14)	-1.51(19)	-2.1(2)	1.51(19)	107(6)	1.9 %
62c	71	0.831(14)	-0.47(18)	1.47(15)	0.47(18)	133(6)	1.1 %

Tabelle 42: Ergebnisse der Vorverfeinerung der [101]-Zonenachse von ScGaO₃.

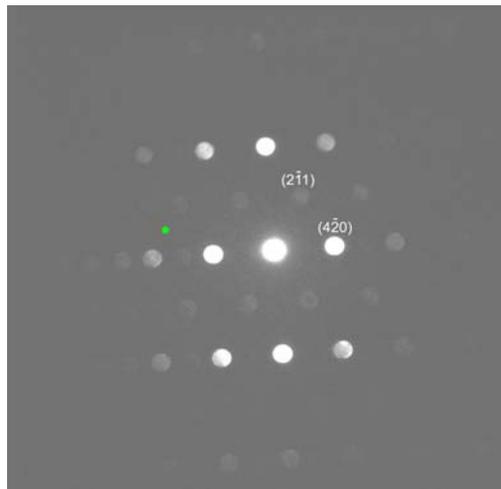
Die Punktbeugungsbilder der Abbildung 62a bis c wurden von dem mit einem weißen Pfeil markierten Bereich des Kristalls in Abbildung 63 erstellt.

Da die verfeinerten Werte des Lauzentrums in die richtige Richtung verschoben (s. Abbildung 62 a-c und Tabelle 42) wurden und die R-Werte klein sind, wurden die drei Datensätze zur Strukturverfeinerung verwendet.

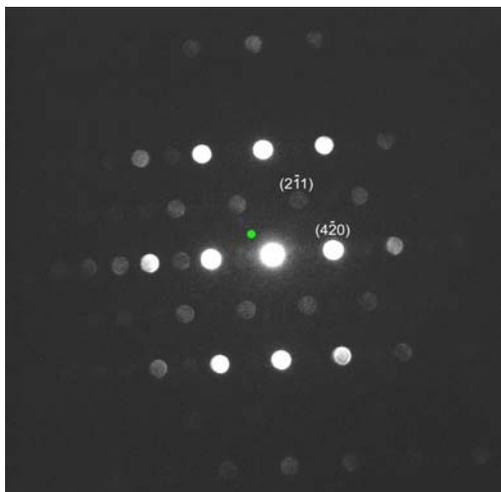
3.4.1.5 Zonenachse [120] von ScGaO₃

Diese Zonenachse konnte aufgrund des ausreichenden Abstandes der einzelnen Reflexscheiben unter konvergenter Einstrahlbedingung aufgezeichnet werden.

a)



b)



c)

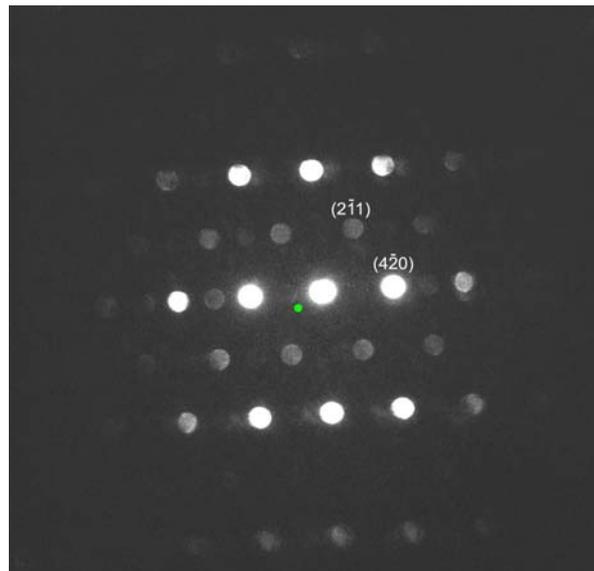


Abbildung 64: Drei Beugungsbilder von ScGaO_3 in $[120]$. Kameralänge: 780 mm. Grüner Kreis: Verfeinertes Lauezentrum des zentral ausgelesenen Datensatzes (Auslesekoordinaten x und $y = 0$). Größe des grünen Kreises = Fehler in der Bestimmung des Lauezentrums (s. Tabellen 43 bis 45).

Aufnahmezeit: 1 s.

Datensatz Nr.	Auslesekoor.		Lauezentrum			Dicke [Å]	Skalen- faktor	R- Wert
	x	y	h	k	l			
1	0	0	-7.34(19)	3.67(10)	-0.48(3)	54(1)	1.017(5)	0.1%
2	-10	0	-6.50(17)	3.25(8)	-0.29(2)	55(1)	1.017(4)	0.1%
3	10	0	-8.0(4)	4.0(2)	-0.76(9)	47(2)	1.017(7)	0.3%
4	0	-10	-7.8(2)	3.90(11)	-0.68(5)	52(1)	1.012(5)	0.2%
5	0	10	-6.36(16)	3.18(8)	-0.26(2)	58(1)	1.030(4)	0.1%
6	5	5	-6.59(18)	3.29(9)	-0.46(3)	54(1)	1.019(4)	0.1%
7	-5	-5	-7.5(2)	3.75(12)	-0.53(4)	54(2)	1.006(5)	0.2%

Datensatz Nr.	Auslesekoor.		Lauezentrum			Dicke [Å]	Skalen- faktor	R- Wert
	x	y	h	k	l			
1	0	0	-1.04(15)	0.52(7)	0.367(20)	78(1)	0.998(6)	0.2%
2	-10	0	-0.61(14)	0.31(7)	0.407(19)	77(1)	0.996(6)	0.2%
3	10	0	-1.54(11)	0.77(5)	0.332(14)	81(1)	1.003(5)	0.1%
4	0	-10	-1.32(13)	0.66(6)	0.189(15)	80(1)	0.995(5)	0.2%
5	0	10	-1.33(13)	0.67(7)	0.55(2)	76(1)	0.998(4)	0.1%
6	5	5	-1.21(11)	0.61(6)	0.395(15)	81(1)	0.999(5)	0.1%
7	-5	-5	-0.87(13)	0.43(7)	0.323(17)	78(1)	0.997(6)	0.2%

Datensatz Nr.	Auslesekoor.		Lauzentrum			Dicke [Å]	Skalen- faktor	R- Wert
	x	y	h	k	l			
1	0	0	-1.56(5)	0.78(2)	-0.183(7)	118(1)	0.990(5)	0.1%
2	-10	0	-1.48(4)	0.74(2)	-0.149(6)	110(1)	0.989(4)	0.1%
3	10	0	-1.52(4)	0.76(2)	-0.224(6)	123(1)	1.005(5)	0.1%
4	0	-10	-1.63(5)	0.81(2)	-0.215(7)	120(1)	0.999(5)	0.1%
5	0	10	-1.29(5)	0.65(2)	-0.149(7)	112(1)	0.986(4)	0.1%
6	5	5	-1.43(4)	0.71(2)	-0.195(7)	119(1)	0.993(5)	0.1%
7	-5	-5	-1.56(5)	0.78(3)	-0.194(8)	117(1)	0.991(5)	0.1%

Tabellen 43 bis 45: Berechnete Werte einer Vorverfeinerung mit Intensitätsdaten in der [120]-Zonenachse von ScGaO₃. Tabelle 43 gehört zur Abbildung 64a, Tabelle 44 zur Abbildung 64b und Tabelle 45 zur Abbildung 64c.

Der zentral ausgelesene Datensatz des Beugungsbildes, der zur Abbildung 64a gehört, enthält 64, der zur Abbildung 64b 73 und der zur Abbildung 64c 88 verwertbare Reflexe.

Die drei Beugungsbilder wurden von dem in Abbildung 57 gezeigten Kristall aufgezeichnet. Die weißen Rahmen markieren die Aufnahmebereiche der Beugungsaufnahmen der Abbildung 64a bis c.

Die verfeinerten Daten der Skalenfaktoren und die der Kristalldicke in den Tabellen 43 bis 45 sind zu den selben Werten konvergiert. Auch die verschiedenen Einstrahlrichtungen der unterschiedlichen Datensätze werden in den Werten der Lauzentren reproduziert.

3.4.2 Strukturverfeinerung von ScGaO₃

Zur Strukturverfeinerung von ScGaO₃ wurden von den unter konvergenten Einstrahlbedingungen erstellten Beugungsbildern die zentral ausgelesenen Datensätze verwendet. Insgesamt wurden damit 14 Datensätze mit 1947 Reflexen zur Strukturverfeinerung herangezogen. Die verfeinerten datensatzspezifischen Resultate aller berücksichtigten Aufnahmen sind in Tabelle 46 dargestellt. Die verfeinerte Struktur ist in Tabelle 47 gezeigt.

Zonen- achse	Lauezentrum			Proben- dicke	Skalen- faktor	R-Faktor [%]
	h	k	l			
[100]	0	-2.03(13)	-0.42(2)	84(1)	0.958(11)	2.1
[100]	0	-2.56(10)	-0.573(16)	117(2)	0.958(11)	1.5
[100]	0	-0.91(4)	0.277(9)	135(1)	0.966(11)	2.1
[010]	0.8(5)	0	-0.19(5)	63(2)	1.114(15)	0.8
[010]	5.1(3)	0	-0.17(3)	96(1)	1.044(15)	3.0
[010]	7.02(14)	0	-0.179(19)	129(2)	1.020(16)	4.2
[001]	0.05(3)	0.11(5)	0	181(1)	1.24(3)	1.7
[001]	-0.12(4)	0.05(5)	0	185(1)	1.11(3)	2.0
[001]	-0.39(4)	0.17(5)	0	188(1)	1.13(3)	2.4
[101]	-4.87(5)	-0.37(3)	4.87(5)	191(4)	0.743(11)	2.7
[101]	-1.92(8)	-2.93(13)	1.92(8)	92(2)	0.975(12)	1.3
[101]	-0.33(9)	-1.59(9)	0.33(9)	130(3)	0.937(11)	0.5
[120]	-7.4(4)	3.71(20)	-0.47(7)	54(3)	1.024(10)	0.2
[120]	-1.1(3)	0.57(13)	0.36(3)	77(2)	0.997(10)	0.2
Gesamt R-Wert 1947 Reflexe = 1.5 %						

Tabelle 46: Datensatzspezifische Resultate nach der Strukturverfeinerung der verwendeten Zonenachsen von ScGaO₃.

Atom	<i>x/a</i>	<i>y/b</i>	<i>z/c</i>	Debye-Waller Faktor	Besetzungs- faktor	Wyk- hoff
Sc(1)	0.6291(13)	0.463(3)	0.320(16)	2.00(8)	1	4c
Sc(2)	0.762(3)	0.193(5)	0.80(6)	2	0.33(7)	4c
Ga(1)	0.5008(9)	0.260(2)	0.819(16)	1.95(6)	1	4c
Ga(2)	0.6252(15)	0.981(3)	0.30(2)	1.94(7)	1	4c
Ga(3)	0.762(3)	0.193(5)	0.80(6)	2	0.67(7)	4c
O(1)	0.1781(17)	0.328(3)	0.29(3)	1.9(3)	1	4c
O(2)	0.4233(20)	0.466(4)	0.81(2)	1.3(3)	1	4c
O(3)	0.2016(19)	0.042(4)	0.83(2)	2.4(3)	1	4c
O(4)	0.3226(17)	0.246(4)	0.36(2)	2.6(4)	1	4c
O(5)	0.0451(20)	0.293(3)	0.30(3)	1.3(3)	1	4c
O(6)	0.4268(20)	0.067(4)	0.853(18)	2.4(4)	1	4c

Tabelle Nr. 47: Parameter der verfeinerten Kristallstruktur von ScGaO₃.

Atom Nr. 1	Atom Nr. 2	Abstand [Å]
Sc1	O1	2.291(33)
	O2	1.932(72)
	O2	1.985(72)
	O3	2.142(69)
	O4	2.287(37)
	O5	2.437(38)
Sc2/Ga3	O1	2.247(158)
	O1	2.292(161)
	O3	1.827(58)
	O3	2.226(53)
	O4	1.876(159)
	O4	2.281(172)
Ga1	O2	2.088(37)
	O2	2.441(36)
	O5	1.802(96)
	O5	1.908(98)
	O6	1.978(37)
	O6	2.725(35)
Ga2	O1	1.702(35)
	O4	1.919(39)
	O5	2.226(39)
	O6	1.778(75)
	O6	2.063(78)

 Tabelle 48: Bindungsabstände in ScGaO₃.

Die berechneten Fehler der datensatzspezifischen Ergebnisse sind im Vergleich zu den Fehlern in den Vorverfeinerungen kleiner geworden, was ein Hinweis auf eine gelungene Verfeinerung ist (vgl. Tabelle 46 mit Tabellen 37 bis 45). Alle aus den verfeinerten Ergebnissen resultierenden Bindungsabstände und –winkel liegen im üblichen Bereich für Oxide von Sc und Al (s. Tabelle 48). Auffällig dabei ist, dass die verfeinerte Atomposition der mit Sc(2) und Ga(3) besetzen Lage, einen um mindestens den Faktor 2 kleinere Genauigkeit liefert als die übrigen Atompositionen. Die Debye-Waller-Faktoren dieser Atome konnten nicht mit verfeinert werden, da sie nicht zu einem Wert konvergieren.

Die Genauigkeit in der Bestimmung der z-Koordinate liegt z.B. bei der O(1)-Position mit ± 9.4 pm ($= 0.03 \cdot 312.54$ pm) um 6.2 pm ($0.0017 \cdot 1900.4$ pm = 3.2 pm) bzw. 7.2 pm ($0.03 \cdot 718.59$ pm = 2.2 pm) über der Genauigkeit in der Festlegung der x-

Koordinate bzw. der y -Koordinate. Die größere Ungenauigkeit in der Bestimmung der z -Koordinate setzt sich auch bei den übrigen Atomen fort. Die aus den berechneten Fehlern in der Sc2/Ga3-Atomposition resultierenden Ungenauigkeiten in den Atomabstände sind ebenfalls im Vergleich zu den übrigen Werten mit einem um mindestens den Faktor 2 höheren Fehler behaftet (s. Tabelle 48). Demnach hat die Methode Schwierigkeiten, bei kürzeren Achsen die Atome auf bestimmte Werte festzulegen.

Ein ausgezeichnete R-Wert von 1.5 % über alle Reflexe zeigt eine erfolgreiche Strukturverfeinerung an.

3.4.2.1 Sc(1)-Koordination

Sc ist in dieser Verbindung siebenfach koordiniert. Sechs Sauerstoffatome liegen in den Ecken eines Oktaeders und eine Oktaederfläche ist mit einem Sauerstoffanion (O5) überkappt. Die Abstände und die jeweils eingeschlossenen Bindungswinkel sind in Abbildung 65 gezeigt. Dort ist auch die Scandiumsphäre als Polyeder dargestellt.

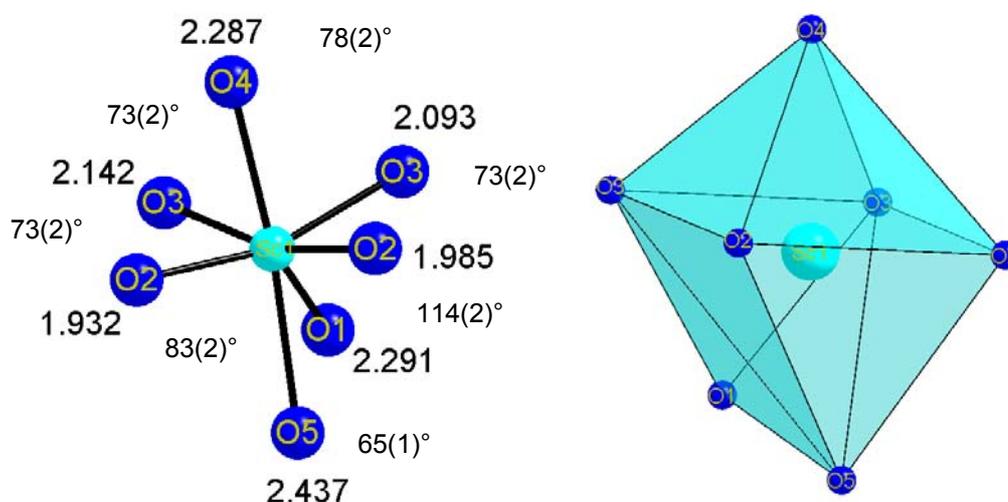


Abbildung 65: Sc(1)-Koordination im ScGaO₃ mit sieben O-Sc-O-Bindungswinkeln. Rechts als Sc(1)O₇-Polyeder dargestellt.

Die noch fehlenden O(X)-Sc(1)-O(X)-Bindungswinkel sind im folgenden dargestellt. Zur eindeutigen Unterscheidung der Sauerstoffanionen wird bei gleicher Bezeichnung dieser der Sc(1)-O(X)-Abstand hinter dem entsprechenden Sauerstoffanion aufgeführt.

O(4)-Sc(1)-O(1)	= 132(1)°,	O(4)-Sc(1)-O(2)1.932	= 85(2)°,
O(4)-Sc(1)-O(2)1.985	= 91(2)°,	O(4)-Sc(1)-O(5)	= 162(1)°,

O(3)2.093-Sc(1)O(3)2.142 = 99(3)°,	O(3)2.093-Sc(1)-O(2)1.932 = 163(3)°,
O(3)2.093-Sc(1)O(1) = 72(2)°,	O(3)2.142-Sc(1)-O(2)1.985 = 164(3)°,
O(3)2.142-Sc(1)-O(1) = 76(2)°,	O(3)2.142-Sc(1)-O(5) = 115(2)°,
O(2)1.932-Sc(1)-O(2)1.985 = 110(3)°,	O(2)1.932-Sc(1)-O(1) = 119(2)° und
O(2)1.985-Sc(1)-O(5) = 81(2)°.	

3.4.2.2 Ga(1)/Ga(2)-Koordination

Die Galliumatome sind auf ihren beiden Positionen jeweils trigonal-pyramidal durch Sauerstoffatome koordiniert. Sie besitzen unterschiedliche Ga(X)-O(X)-Atomabstände, welche in Abbildung 66 und in Tabelle 48 gezeigt sind. Im Falle des Ga(1)-Polyeders ist der Ga-O(5)-Abstand mit 2.441(36) Å im Vergleich zu dem 2.226(39) Å langen Ga(2)-O(5)-Abstand erhöht, weshalb man gerade bei dem Ga(1) auch von einer einfach überkappten verzerrt tetraedrischen Koordination sprechen kann.

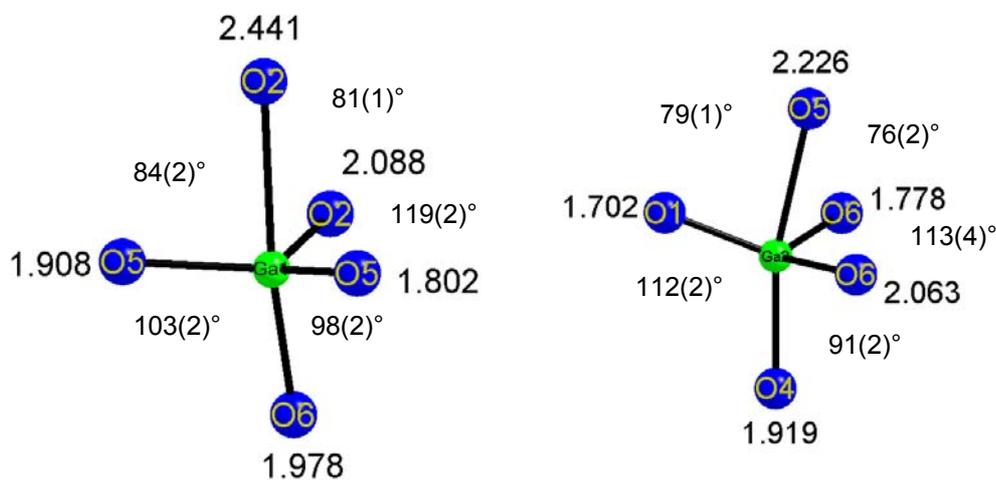


Abbildung 66: Ga(1)- (links) und Ga(2)-Koordination (rechts) mit den entsprechenden Atomabständen und fünf O(x)-Ga(1/2)-O(x)-Bindungswinkel im ScGaO₃.

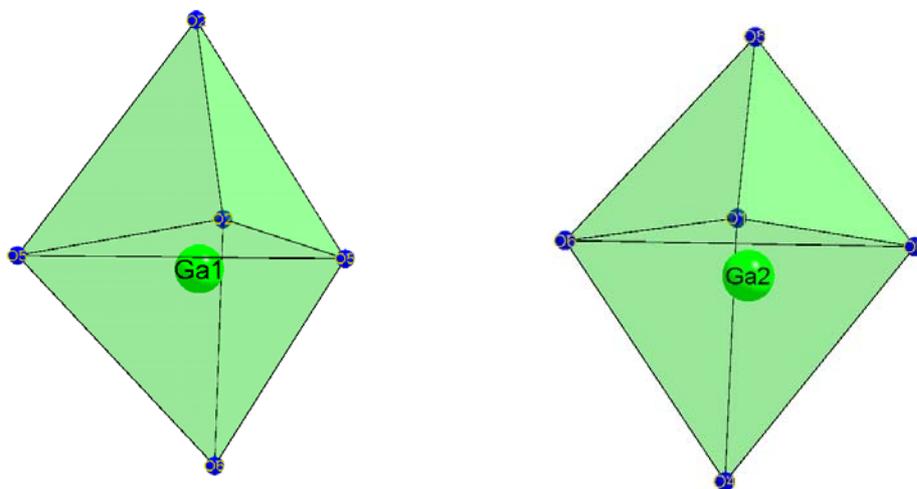


Abbildung 67: Ga(1/2)-Polyeder.

Die weiteren O(X)-Ga(1)-O(X) Bindungswinkel sind:

$$\begin{aligned} \text{O(2)2.441-Ga(1)-O(5)1.802} &= 85(2)^\circ, & \text{O(2)2.441-Ga(1)-O(6)} &= 171(1)^\circ, \\ \text{O(5)1.908-Ga(1)-O(2)2.088} &= 116(2)^\circ, & \text{O(5)1.908-Ga(1)-O(5)1.802} &= 120(5)^\circ \text{ und} \\ \text{O(6)1.978-Ga(1)-O(2)2.088} &= 90(1)^\circ. \end{aligned}$$

Die weiteren O(X)-Ga(2)-O(X)-Bindungswinkel sind:

$$\begin{aligned} \text{O(5)-Ga(2)-O(6)2.063} &= 78(2)^\circ, & \text{O(5)-Ga(2)-O(4)} &= 167(2)^\circ, \\ \text{O(1)-Ga(2)-O(6)1.778} &= 118(2)^\circ, & \text{O(1)-Ga(2)-O(6)2.063} &= 116(2)^\circ \text{ und} \\ \text{O(6)1.778-Ga(2)-O(4)} &= 102(2)^\circ. \end{aligned}$$

3.4.2.3 Sc(2)/Ga(3)-Koordination

Die Kationen auf dieser Position sind oktaedrisch von sechs Sauerstoffatomen umgeben, wobei es sich dabei um drei kristallographisch unterschiedliche Sauerstoffatome (O1, O3 und O4) handelt. Vier Atomabstände sind dabei mit ca. 2.2 Å deutlich länger als die beiden kurzen mit 1.8 Å.

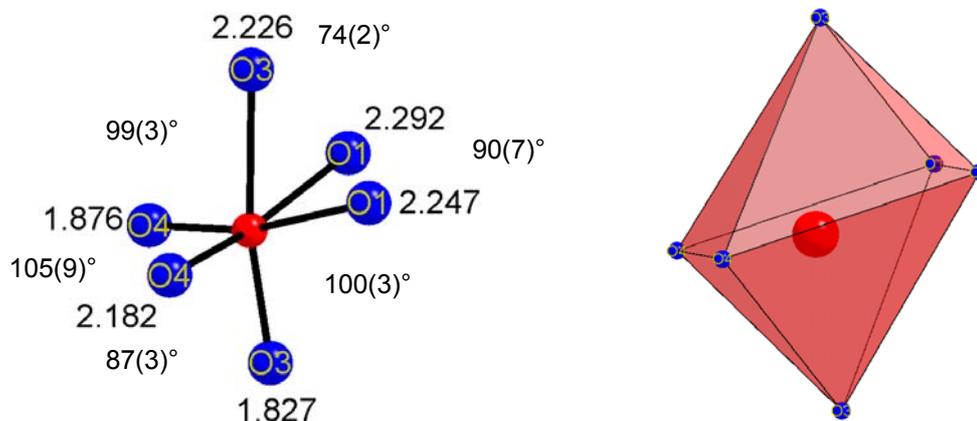


Abbildung 68: Sc(2)/Ga(3)-Koordination mit Atomabständen und sechs Kation-O(X)-Bindungswinkel. Rechts als Polyeder dargestellt.

Weitere Bindungswinkel sind im folgenden mit K = zentrale Kation dargestellt:

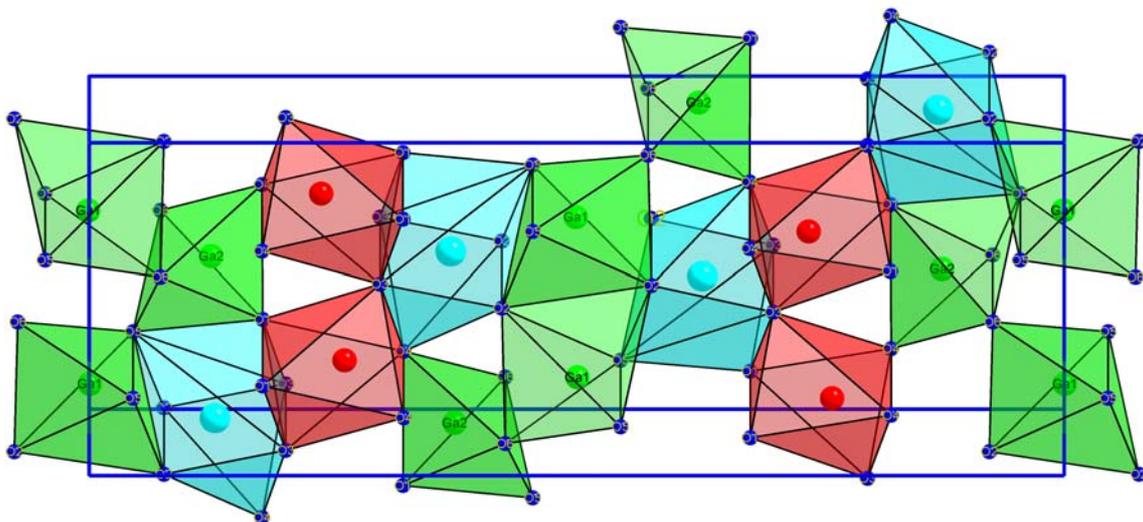
$$\begin{aligned} \text{O(3)2.226-K-O(1)2.247} &= 70(2)^\circ, & \text{O(3)2.226-K-O(4)2.182} &= 94(3)^\circ, \\ \text{O(3)2.226-K-O(3)1.827} &= 170(3)^\circ, & \text{O(4)1.876-K-O(1)2.292} &= 84(3)^\circ, \\ \text{O(4)1.876-K-O(1)2.247} &= 169(5)^\circ, & \text{O(4)1.876-K-O(3)1.827} &= 91(3)^\circ, \\ \text{O(1)2.292-K-O(4)2.182} &= 166(5)^\circ, & \text{O(1)2.292-K-O(3)1.827} &= 104(3)^\circ \text{ und} \\ \text{O(1)2.247-K-O(4)2.182} &= 79(3)^\circ. \end{aligned}$$

3.4.2.4 Elementarzelle von ScGaO₃

Die beste Einsicht in die Struktur von ScGaO₃ wird entlang [011] erhalten. Zur besseren Übersicht über die Verknüpfungsmuster wird einmal die komplette Sc(1)-Koordination als überkappter Oktaeder in die Elementarzelle gezeichnet (s. Abbildung 69a), wohingegen die überkappenden Sauerstoffionen O(5) in Abbildung 69b nicht mit in die Sc(1)-Polyeder eingeschlossen sind.

Die Sc(1)-Polyeder (cyan) teilen mit zwei Sc(2)/Ga(3)-Oktaedern (rot) eine gemeinsame Kante. Die entsprechenden Atomabstände sind mit Ga(3)/Sc(2)-O(1) = 2.292(161) Å und Sc(1)-O(1) = 2.291(33) Å zu einem Sauerstoffion an der gemeinsamen Kante und mit Ga(3)/Sc(2)-O(3) = 2.226(53) Å und Sc(1)-O(3) = 2.142(69) Å zu dem anderen Sauerstoffion an der gemeinsamen Kante im üblichen Bereich für kantenverknüpfte Sc- bzw. Ga-Oktaeder (s. Kap. 3.5). Betrachtet man den Sc(1)-Polyeder nur als Oktaeder, so ist dieser mit zwei weiteren Ga(1)-Oktaedern über dieselbe Ecke O(2) verknüpft. Wird das überkappende Sauerstoffion O(5) des Sc(1)-Polyeders mit in diese Betrachtung einbezogen, besitzt der Sc(1)-Polyeder mit dem Ga(1)-Polyeder eine gemeinsame Kante. Jeweils ein Kation-Sauerstoff-Abstand innerhalb jedes dieser Polyeder an dieser Kante ist aber mit 2.437(38) Å (= Abstand Sc(1)-O(5)) im Falle des Sc(1)-Polyeders und mit 2.441(36) Å (= Abstand Ga(1)-O(2)) im Falle des Ga(1)-Oktaeders deutlich gegenüber den übrigen Abständen innerhalb dieser Polyeder verlängert. Die Ga(2)-Polyeder (grün) sind über ein O(4)-Atom jeweils mit Sc(2)/Ga(3)-Oktaeder (rot) über eine Ecke verknüpft. Die Abstände liegen mit Ga(2)-O(4) = 1.919(39) Å und Sc(2)/Ga(3)-O(4) = 1.876(159) Å unter denen der kantenverknüpften Polyeder.

a)



b)

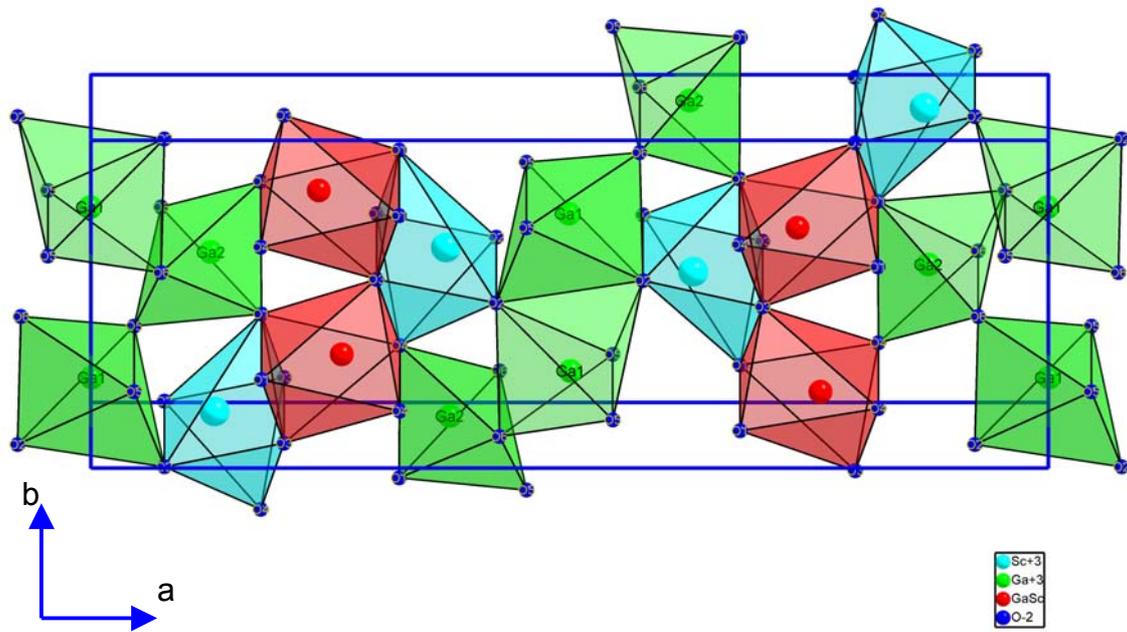


Abbildung 69: Die Elementarzelle von ScGaO₃ in [011]. In der oberen Abbildung (a) ist das überkappte Sc(1)-Oktaeder vollständig dargestellt, während in (b) die überkappenden Sauerstoffanionen O(5) nicht mit in den Polyeder einbezogen sind. Die blaue Linie kennzeichnet die Elementarzellenbegrenzung.

3.5 Vergleich der gefundenen Bindungsabstände mit den Oxiden β -Ga₂O₃, Sc₂O₃ und α - bzw. γ -Al₂O₃

In der Struktur des nur in einer Modifikation vorkommenden Scandium(III)oxids, lässt sich eine Variationsbreite der Bindungswinkel und –abstände des Scandiums in Bezug auf seine oktaedrische Umgebung ablesen. Sc₂O₃ kristallisiert in der Raumgruppe $Ia\bar{3}$ (Nr. 206) mit der Gitterkonstante 9.849 Å. Die eine vorkommende Scandiumposition ist ideal oktaedrisch - und die zweite Scandiumposition stark verzerrt oktaedrisch koordiniert. Innerhalb der ideal oktaedrischen Koordination ist ein Scandium-Sauerstoff-Bindungsabstand von 2.120 Å vorhanden. In dem verzerrten ScO₆-Oktaeder sind drei unterschiedliche Scandium-Sauerstoffabstände von 2.080 Å, 2.125 Å und 2.164 Å existent. Damit einher geht eine starke Variation der O-Sc-O-Bindungswinkel auf dieser Position. So tauchen Bindungswinkel innerhalb des verzerrten Oktaeders von 79°, 80°, 86°, 98°, 109°, 114°, 119°, 140° und 164° auf [55]. Im Sc₃Sc_{0.32}Al_{0.68}O₆ ist die Verzerrung der Sc(1)-Oktaeder in ähnlicher Weise ausgeprägt wie im Scandiumoxid. Auch die Eigenschaft des Scandium(III)ions, sich in idealer oktaedrischer Umgebung aufzuhalten, findet sich im Sc₃Sc_{0.32}Al_{0.68}O₆ auf der Sc(2)- bzw. der Sc(3)-Position wieder.

Das Vorhandensein von Scandium(III)ionen auf den ideal oktaedrisch koordinierten Positionen ist aber auch in der Kristallstruktur von Sc₃Sc_{0.32}Al_{0.68}O₆ intrinsisch existent, da im isostrukturellen Sc₃CrO₆ [48] das Chrom durch Scandium und Aluminium ersetzt wird. Werden die Besetzungsfaktoren mit den Ionenradien von Sc³⁺ (75 pm) und Al³⁺ (54 pm) multipliziert, so ist die Summe für die Sc(2)/Al(1)-Position gleich 62.6 pm (= 0.409 · 75 pm + 0.591 · 54 pm). Auf dieser Position wird das Chrom(III)ion (Ø 62 pm) im Mittel gröbengleich ersetzt. Mit 58.7 pm (= 0.226 · 75 pm + 0.774 · 54 pm) erhält man auf der Sc(3)/Al(2)-Position einen um 3.3 pm kleineren Wert (Alle angegebenen Ionenradien nach [53] in oktaedrischer Koordination).

Im ScGaO₃ ist das Scandium Sc(1) mit einer Koordinationszahl von sieben einfach überkappt oktaedrisch von Sauerstoffatomen umgeben. Die dort gefundenen Bindungsabstände zu den sechs nächsten Sauerstoffatomen innerhalb des Polyeders besitzen demnach mit 1.932 – 2.291 Å eine größere Variationsbreite im Vergleich zu den Sc-O-Bindungsabständen im Scandium(III)oxides. Die resultierenden O(X)-Sc-O(X)-Bindungswinkel von 73°-164° sind mit denen im Sc₂O₃ vergleichbar.

Scandium(III)ionen können folglich auch Positionen in ternären Verbindungen besetzen, die eine verzerrte oktaedrische Sphäre oder sogar eine siebenfache Koordination besitzen. Damit erhöht ein Scandium(III)ion die Anzahl von Verbindungen, bei denen die Kristallstruktur dieser Verbindung verschiedenartige

Verzerrungen oder eine Aufweitung der Koordinationszahl einer oktaedrischen Koordination für seine Stabilität benötigt.

Gallium ist in allen seinen bekannten Gallium(III)oxidmodifikationen entweder tetraedrisch- oder oktaedrisch von Sauerstoffatomen umgeben. In der thermodynamisch stabilsten Galliumoxidmodifikation β -Ga₂O₃ (Raumgruppe *C2/m* Nr. 12 $a = 12.23 \text{ \AA}$, $b = 3.04 \text{ \AA}$, $c = 5.80 \text{ \AA}$ und $\beta = 103.7^\circ$) liegen dabei die Gallium-Sauerstoff-Atomabstände der tetraedrisch koordinierten Galliumatome bei 1.803 \AA , 1.833 \AA und 1.853 \AA , während sie bei der oktaedrischen Koordination mit 1.978 \AA , 2.023 \AA und 2.077 \AA naturgemäß über den Werten für die tetraedrische Koordination liegen [56]. Die im ScGaO₃ gefundene fünffache Koordination der Galliumatome Ga(1) und Ga(2) (trigonal bipyramidal) benötigt demnach in diesem Falle Scandium als ein weiteres Kation. Jeweils ein Sauerstoff-Gallium-Bindungsabstand innerhalb der trigonalen Bipyramide ist aber mit 2.441 \AA (Ga(1)) und mit 2.226 \AA (Ga(2)) um mehr als 10 % größer als die übrigen gefundenen Bindungsabstände innerhalb der GaO₅-Polyeders. Aufgrund dieser Tatsache und der, dass das Gallium(III)ion in beiden Fällen aus der trigonalen Fläche etwas herausgerückt ist, ist eine Tendenz des Galliums, sich wie im binären Oxiden β -Ga₂O₃ in tetraedrischer Koordination aufzuhalten, auch im ScGaO₃ erkennbar. Die mit Scandium und Gallium im ScGaO₃ gemischt besetzte Lage, die verzerrt oktaedrisch koordiniert ist, bestätigt die getroffenen Aussagen über das Scandium und Gallium.

Das Aluminium im α -Al₂O₃ (Korund; thermodynamisch stabilste Aluminiumoxidmodifikation) weist als Element der dritten Hauptgruppe eine Ähnlichkeit zum Scandium als Element der dritten Nebengruppe in beiden Verbindungen Sc₃Sc_{0.32}Al_{0.68}O₆ und ScGaO₃ auf. Dort liegen zwar im Gegensatz zum Sc₃Sc_{0.32}Al_{0.68}O₆ und ScGaO₃ flächenverknüpfte Oktaeder vor, aber sie sind wie im Falle des α -Al₂O₃ stark verzerrt. Die Umgebung des Aluminium(III)ions im Sc₃Sc_{0.32}Al_{0.68}O₆ weist zur Sphäre des Aluminium(III)ions im Korund keine Ähnlichkeit auf, da es auf einer hoch symmetrischen Lage (3a und 3b) der Raumgruppe $R\bar{3}$ (Nr. 148) liegt und damit auch die AlO₆-Oktaeder-Symmetrie ohne Verlassen dieser speziellen Lage nicht erniedrigt werden kann. Aluminium zeigt aber z.B. in seinem binären Oxid γ -Al₂O₃ auch das Verhalten, sich auf hoch symmetrischen Lagen aufzuhalten. In dieser Defekt-Spinell-Struktur (Raumgruppe $Fd\bar{3}m$ Nr. 227) liegt das oktaedrisch koordinierte Aluminium(III)ion auf der 16 d-Lage und ein geringer Anteil auf der 32 e-Lage [57]. Aluminium ist demnach ebenfalls wie das Scandium in der Lage, sich auf mehreren kristallographisch unterschiedlichen Positionen in einer Verbindung aufzuhalten. Im Sc₃Sc_{0.32}Al_{0.68}O₆ wird es vom Scandium auf die höher symmetrische Lage verdrängt, was auf den größeren Ionenradius von Sc³⁺ zurückzuführen ist.

3.6 Vergleich der neu entwickelten mit der bisherigen Methode zur Aufnahme von Elektronenbeugungsdaten

Die Verwendung von Elektronenbeugungsbildern, welche unter konvergenten Einstrahlbedingungen zur Strukturverfeinerung erstellt werden, hat drei entscheidende Vorteile:

- Die im Falle paralleler Beleuchtung der Probe auftretende Übersättigung des CCD-Chips tritt bei konvergenter Beleuchtung nicht auf. Hierdurch ist es möglich, die im Primärstrahl vorhandenen Informationen für die Strukturverfeinerung zu nutzen.

- Die visuell sichtbare Variation der Intensitäten in den Beugungsscheiben der Beugungsbilder mit zunehmender Kristalldicke lässt eine Überprüfung der verfeinerten Kristalldicke durch Bildsimulationen zu. Die experimentellen Beugungsbilder können mit simulierten Bildern, die mit der verfeinerten Kristalldicke und dem verfeinerten Laugezentrum berechnet werden, sehr einfach verglichen werden. Der Wahrheitsgehalt der verfeinerten Kristalldicke und des Laugezentrums lässt sich demnach leicht überprüfen. Durch parallele Beleuchtung erstellte Punktbeugungsbilder ermöglichen zwar einen Vergleich, dieser ist aber nicht visuell sondern nur durch direkten Intensitätsvergleich der Reflexe des experimentellen Bildes mit einem simulierten Beugungsbild realisierbar. Dieses zeitaufwendige und mühsame Verfahren ist aber nicht praktikabel. Das Ergebnis eines verfeinerten Laugezentrums lässt sich allerdings auch bei Punktbeugungsbildern leicht innerhalb des Beugungsbildes auf seine Richtigkeit überprüfen.

- Schließlich gestattet die hier vorgestellte neue Methode zur Auswertung von unter konvergenten Einstrahlbedingungen erstellten Beugungsbildern die Überprüfung der in der Vorverfeinerung erhaltenen Ergebnisse eines Beugungsbildes auf Konsistenz (Skalenfaktoren und Kristalldicken, die von Datensätzen eines Beugungsbildes stammen, müssen zu den selben Werten konvergieren). Die Daten der Laugezentren sollten die Einstrahlrichtungen reproduzieren. Diese Kriterien zeigen die Verwendung eines Datensatzes zur Strukturverfeinerung an. Daten von Punktbeugungsbildern kann man nicht in dieser Weise analysieren. Manche Beugungsbilder, deren äußeres Erscheinungsbild keinen Anschein auf deren Nichtverwendbarkeit liefert, können durch diese Überprüfung von der Strukturverfeinerung ausgeschlossen werden.

Nicht unerwähnt soll hier ein Nachteil der neuen Methode bleiben. Da sich ab einer Gitterkonstante von 13.5 \AA die Beugungsscheiben bei den hier verwendeten Mikroskopparametern überlappen, können diese Beugungsbilder nicht mehr ausgewertet werden. Punktbeugungsbilder lassen sich dagegen auch noch bei größeren Gitterkonstanten problemlos auswerten.

3.7 Zusammenfassung

Die Verwendung von Punktbeugungsbildern, die durch einen parallelen Elektronenstrahl zur Strukturverfeinerung entstehen, beinhaltet zwei Nachteile:

- Der Primärstrahlreflex (000) in Punktbeugungsbildern ist immer übersättigt. Damit geht der Informationsgehalt dieses Reflexes verloren.
- Der kleinste erreichbare Elektronenstrahldurchmesser zur Erstellung von zur Strukturverfeinerung verwertbaren Punktbeugungsbildern ist mit 10 nm zu groß, da für geeignete Datensätze ein homogen dicker Kristallbereich beleuchtet werden muss.

Mit der Entwicklung einer neuen Methode zur Auswertung von Beugungsbildern, die unter konvergenten Einstrahlbedingungen erstellt werden, ist es gelungen, einen kleineren Elektronenstrahldurchmesser von ca. 5 nm auf der Probe bei der Erstellung von Beugungsbildern zu verwenden. Zudem ist die Primärstrahlreflexscheibe (000) bei diesen Beugungsbildern nicht mehr übersättigt.

Das hier entwickelte Programm mit der Bezeichnung *konv2.pro* liest Bragg-korrelierte Bereiche aus jeder Beugungsscheibe aus. Es können so mehrere Datensätze aus einem Beugungsbild gewonnen werden. In einer Vorverfeinerung müssen die Resultate der verschiedenen Datensätze eines Beugungsbildes einige Kriterien erfüllen, damit ein Datensatz eines Beugungsbildes zur anschließenden Strukturverfeinerung herangezogen werden kann:

- Die Ergebnisse der Skalenfaktoren und die der Kristalldicken müssen zu den selben Werten konvergieren.
- Die unterschiedlichen Positionen der verschiedenen Datensätze innerhalb der Beugungsscheiben (\equiv Einstrahlrichtungen) sollten sich in den verfeinerten Daten der Lauezentren widerspiegeln.

Im Vergleich zu den Punktbeugungsbildern, bei denen man nur den Wert des verfeinerten Lauezentrums und einen R-Wert als Entscheidungskriterium für die Verwendung zur Strukturverfeinerung hat, bieten unter konvergenten Einstrahlbedingungen erstellte Beugungsbilder zusätzliche Prüfsteine für die Verwendbarkeit von Datensätzen. Die Funktionsfähigkeit der neuen Methode wurde an einem berechneten und an einem experimentellen Beugungsbild von α -Al₂O₃ (Korund) in [0001] überprüft.

Eine Strukturverfeinerung wurde an der bekannten Verbindung Ta₂P mit Elektronenbeugungsdaten, die unter konvergenten Einstrahlbedingungen erstellt sind,

durchgeführt. Es konnten in der Literatur noch nicht erwähnte innenzentrierte Ta9-Würfel innerhalb der Kristallstruktur aufgefunden werden.

Strukturvorschläge von zwei unbekanntem Verbindungen ($\text{Sc}_3\text{Sc}_{0.32}\text{Al}_{0.68}\text{O}_6$ und ScGaO_3) konnten erfolgreich verfeinert werden. Der R-Wert der Strukturverfeinerung von $\text{Sc}_3\text{Sc}_{0.32}\text{Al}_{0.68}\text{O}_6$ liegt bei 0.7 % (1253 Reflexe) und der von ScGaO_3 bei 1.5 % (1947 Reflexe). Dabei wurden zur Strukturverfeinerung von $\text{Sc}_3\text{Sc}_{0.32}\text{Al}_{0.68}\text{O}_6$ nur unter konvergenten Einstrahlbedingungen erstellte Beugungsbilder verwendet. Zur Strukturverfeinerung von ScGaO_3 mussten auch Punktbeugungsbilder verwendet werden, da die lange a -Achse mit 19.004 Å unter konvergenten Einstrahlbedingungen zu einer Überlappung der Beugungsscheiben führt. Die Ergebnisse lieferten bei beiden Verbindungen Atomabstände und Bindungswinkel, die im üblichen Bereich vergleichbarer Verbindungen liegen.

3.8 Ausblick

Mit der Entwicklung von MSLS ist die Elektronenkristallographie einen großen Schritt vorangekommen. Dennoch kann man auch dieses Programm in einigen Punkten verbessern.

Innerhalb von MSLS können nur isotrope Debye-Waller-Faktoren verfeinert werden. In der Röntgenkristallographie (inklusive mit Daten aus Pulveraufnahmen) ist es aber schon länger etabliert, die Debye-Waller-Faktoren anisotrop zu bestimmen. Dazu müssten aber nicht nur die reinen Intensitätsdaten der Reflexe (hkl) zur Strukturverfeinerung herangezogen werden, sondern auch die Gestalt der Reflexe. Anhand der zusätzlichen Information der nicht symmetrischen Verbreiterung der Reflexe, können so die anisotropen Debye-Waller-Faktoren bestimmt werden.

Eine Erhöhung der Anzahl von Datensätzen zur Strukturverfeinerung (bis jetzt 14 Datensätze) ist zudem angebracht, damit man für jede Verbindung die nötige Überbestimmung für eine Strukturverfeinerung erreichen kann.

Ein weiterer Nachteil der hier gezeigten Methode zur Strukturverfeinerung ist, dass sie nur nach der R-Wert-Funktion die strukturellen Parameter optimiert. In der Praxis von Strukturverfeinerungen kommt es sehr häufig vor, dass ein lokales Minimum der R-Wert-Funktion aufgefunden wird, welches zu chemisch und physikalisch nicht sinnvollen Bindungsabständen führt. Liegt dieses lokale Minimum im Vergleich zu seiner Umgebung tief, ist es möglich, dass dieses während einer Verfeinerung nicht mehr verlassen wird, so dass kein vernünftiges Resultat erzielt werden kann. Diese Resultierenden zu kleinen Bindungsabstände können vermieden werden, indem in die Strukturverfeinerung auch eine Berechnung der potenziellen Energie des Systems einfließen lässt. *Putz* zeigte bei einer Strukturlösung mittels Röntgendaten aus Pulvern, dass eine Kombination von gemessenen mit simulierten Diffraktogrammen mit einer Berechnung der potenziellen Energie des Systems zu vernünftigen Ergebnissen führt [58]. Es stehen bereits Algorithmen zur Verfügung, die mit Atompotenzialen chemisch vernünftige Bindungsabstände und -winkel von noch nicht dargestellten Verbindungen vorhersagen können (z.B. „Simulated Annealing“ Methode [59]). Durch diese zusätzlichen Informationen wird der Wert eines lokalen Minimums der R-Wert-Funktion angehoben, so dass eine notwendige Verschiebung der Atome während einer Strukturverfeinerung möglich wird. Durch Entwicklung eines passenden Algorithmus und die zukünftig zur erwartende Steigerung der Rechnerleistung könnte ein Programm entwickelt werden, welches auf beide Funktionen zurückgreift.

4 Literaturverzeichnis

- [1] Tivol, W. F., Dorset, D.L., McCourt, M.P. & Turner, J.N. *Microsc. Soc. Am Bull* **23**(1) (1993) 91-98
- [2] Vincent, R. & Exelby, D.R. *Philos. Mag. Lett.* **63** (1991) 31-38
- [3] Vincent, R. & Exelby, D.R. *Philos. Mag. Lett.* **B68** (1993) 513-528
- [4] Vincent, R. & Midgeley, P.A. *Ultramicroscopy* **53** (1994) 271-282
- [5] Gjønnes, J. & Moodie, A.F. *Acta Cryst.* **19** (1965) 65-67
- [6] Buxton, B.F., Eades, J.A., Steeds, J.W. & Rackham, G.M. *Philos. Trans. R. Soc. London* **281** (1970) 171-194
- [7] Vainshtein, B. K., *Advances in Structure Research by Diffraction Methods*, edited by R. Brill, Vol. **1** (1964) 24-54 New York: Wiley-Interscience
- [8] Voigt-Martin, I. G., Yan, D. H., Wortmann, R. & Elich, K. *Ultramicroscopy* **57** (1995) 29-43
- [9] Voigt-Martin, I.G., Yan, D. H., Yakimansky, A., Schollmeyer, D., Gilmore, C. J. & Bricogne, G. *Acta Cryst.* **A51** (1995) 849-868
- [10] Dorset, D. L. *Structural Electron Crystallography* (1995) New York: Plenum Press
- [11] Vainshtein, B. K., D'yakon, I. A. & Ablov, A.V. *Sov. Phys. Dokl.* **15** (1971) 645-647
- [12] Vincent, R. & Exelby, D.R. *Philos. Mag. Lett* **63** (1991) 31-38
- [13] Vincent, R. & Exelby, D.R. *Philos. Mag. Lett* **B68** (1993) 513-528
- [14] Coene, W., Thust, A., Op de Beck, M. & v. Dyck, D. *Ultramicroscopy* **64** (1996) 167
- [15] Weirich, T. E., Ramlau, R., Simon, A. & Hovmöller, S. *Nature* **382** (1996) 144-146
- [16] Hu, J. J., Li, F. H. & Fan, H. F. *Ultramicroscopy* **41** (1992) 387-397
- [17] Dorset, D. L., Jap, B. K., Ho, M. S. & Glaeser, R. M. *Acta Cryst.* **A35** (1979) 1001-1009
- [18] Moss, B. & Dorset, D. L. *Acta Cryst.* **A38** (1982) 201-211
- [19] Karle, J. & Karle, I. L. *Acta Cryst.* **17** (1964) 835-841
- [20] Dorset, D. L. & Hauptmann, H. A. *Ultramicroscopy* **1** (1976) 195-201
- [21] Dorset, D. L. *Acta Cryst.* **B52** (1996) 753-769
- [22] Karle, J. & Hauptmann, H. *Acta Cryst.* **9** (1956) 635-651
- [23] Sayre, D. *Acta Cryst.* **5** (1952) 60-65
- [24] Bricogne, G. & Gilmore, C. J. *Acta Cryst.* **A46** (1990) 284-297
- [25] Gilmore, C. J., Shankland, K. & Fryer, J. R. *Ultramicroscopy* **49** (1993) 132-146
- [26] Gilmore, C. J., Shankland, K. & Fryer, J. R. *Ultramicroscopy* **49** (1993) 132-146

- [27] Li, F. H. *Electron Crystallography of Organic Molecules* edited by J. R. Fryer & D. L. Dorset S. 153-167. Dordrecht: Kluwer Academic Press
- [28] Fan, H. F. (1991) *Direct Methods of Solving Crystal Structures*, edited by H. Schenk, 265-272. New York: Plenum Press
- [29] Woolfson, M. & Fan, H. F. (1995) *Physical and Nonphysical Methods of Solving Crystal Structures* 214-233. Cambridge: University Press
- [30] Xiang, S. B., Fan, H. F., Wu, X. J., Li, F. H. & Pan, Q. *Acta Cryst.* **A46** (1990) 929-934
- [31] Mo. Y. D., Cheng, T. Z., Fan, H. F., Li, J. Q., Sha, B. D., Zheng, C. D., Li, F. H. & Zhao, Z. X. *Supercond. Sci Technol.* **5** (1992) 69-72
- [32] Cowley, J. M. *Acta Cryst.* **6** (1953) 522-529
- [33] Dorset, D. L. *Acta Cryst.* **A48** (1992) 568-574
- [34] Zou, X. D., Hovmöller, S., Parras, M., Gonzalez-Calbet, J. M., Vallet-Regi, M. & Grenier, J. C. *Acta Cryst.* **A49** (1993) 27-35
- [35] Dorset, D. L: *Structural Electron Crystallography*. New York: Plenum Press
- [36] Campbell Smith, P. J. & Arnott, S. *Acta Cryst.* **A34** (1978) 3-11
- [37] Jansen, J., Tang, D., Zandbergen, H. W. & Schenk, H. *Acta Cryst.* **A54** (1998) 91-101
- [38] IDL Interactive Data Language Reserves System Inc. Boulder, Colorado USA
- [39] Stadelmann, P. *Ultramicroscopy* **21** (1987) 131
- [40] Lutterrotti, L., Scardi, P., *Journal of applied Crystallography* **23** (1990) 246
- [41] D.W. Susnitzky und C.B. Carter *Mat. Res. Soc. Symp. Proc.* **Vol. 82** 1987
Surface Morphology of single crystals ceramics
- [42] Nylund, A. *Acta Chimica Scandinavica* **20** (1966) 2393-3401
- [43] Weirich, T. E., *Crystallography Reports* **Vol 43** 1998 Iss 6 956-967
- [44] Simon, A. *Kondensierte Metall-Cluster Angew. Chem.* **93** (1981) 23-44
- [45] Holleman, A.F., Wiberg, E. *Lehrbuch der Anorganischen Chemie*, Walter de Gruyter Berlin New York 1995 101. Auflage 1430
- [46] H. Schulz, G. Bayer *Acta Cryst.* **B27** (1971), 815
- [47] Dissertation Daniel Müller Universität Bonn 2002
- [48] Schneider, S. J., et al, *J. Res. NBS* **65 A** (1961) 345
- [49] Schultz, H., Bayer, G. *Naturwissenschaften* **57** (1970) 393
- [50] Bayer, G. *Naturwissenschaften* **55** (1968) 33
- [51] Kasper, H. *Z. anorg. Chem* **356** (1968) 329
- [52] Bayer, G. *Z. Kristallogr.* **124** (1967) 131
- [53] R.D. Shannon Revised effective Ionic Radii and systematic studies of interatomic distances in halides and chalcogenides *Acta Cryst. A* **32** (1976) 751
- [54] Nodari, I., Malaman, B. & Evard, O. *Mat. Res. Bull.* **Vol 20** (1985) 687-695
- [55] Norrestham, R. *ARKEA* **29** (1968) 343-349

-
- [56] Geller, S. Crystal structure of β -Ga₂O₃ Journal of chemical physics **33** (1960) 676-684
- [57] Zhou, R-S, Snyder, R.L. Structures and transforming mechanism of the η , γ and θ transition aluminas Acta Cryst **B47** (1991) 617-630
- [58] Putz, H. Acta Cryst. **A55** (1999) Anhang: IUCR Glasgow 1999 Beitrag: M12.OD.004
- [59] Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P. Science **220** **4598** (1983) 671-680
- [60] Heimendahl, von M. Einführung in die Elektronenmikroskopie Vieweg & Sohn, Braunschweig (1970)
- [61] Krischner, H. Einführung in die Röntgen-Feinstruktur-Analyse, Vieweg & Sohn, Braunschweig 4. Auflage (1990)
- [62] Spence, J., C., H. und Zuo, J., M. Electron Microdiffraction, Plenum Press, New York (1992)
- [63] Reimer, L. Transmission Electron Microscopy, 4th Edition Springer-Verlag Berlin Heidelberg New York (1997)
- [64] Cowley, J., M. & Moodie, A., F. Acta Cryst **10** (1957) 609-619
- [65] Giacovazzo, C. Monaco, H., L., Viterbo, D., Scordari, F., Gilli, G., Zanotti, G. & Cati, M. Fundamentals of Crystallography, International Union of Crystallography/Oxford University Press (1992) 90-98
- [66] Philips Manual (1999)
- [67] Mündliche Mitteilung Dr. Thomas Walter Universität Bonn
- [68] Loewe, A. Diplomarbeit, Bonn (1994)



5 Anhang

5.1 Mathematischer Zusammenhang zwischen dem Atomformfaktor der Elektronenbeugung mit dem der Röntgenbeugung

Mit Hilfe der Poissongleichung, die das elektrostatische Potenzial $V(r)$ mit seiner Ladungsdichteverteilung $\rho(r)$ verknüpft

$$\nabla^2 V(r) = -\frac{|e|}{\epsilon_0}(\rho_n(r) - \rho(r)) \quad (10)$$

mit ϵ_0 = Dielektrizitätskonstante,

ρ_n = Ladungsdichte der Kerne,

$\rho(r)$ = Ladungsdichte der Elektronen [Elektronen/Volumeneinheit]

und e = Elementarladung

erhält man für den Elektronenstreufaktor:

$$f^e(s) = -\frac{1}{16\pi^2 s^2} \int V(r) \nabla^2 \exp(-4 \cdot \pi i s \cdot r) d\tau, \quad (11)$$

$$= -\frac{1}{16\pi^2 s^2} \int \exp(-4 \cdot \pi \cdot i \cdot s \cdot r) \nabla^2 V(r) d\tau \quad (12)$$

$$= \frac{|e|}{16\pi^2 \epsilon_0 s^2} \int [Z\delta(r) - \rho(r)] \exp(-4 \cdot \pi \cdot i \cdot s \cdot r) d\tau \quad (13)$$

$$= \frac{|e|}{16\pi^2 \epsilon_0 s^2} \int [Z - f^x(s)] \quad (14)$$

mit Z = Ordnungszahl des Atoms

und $V(r) = 0$ für große r .

5.2 Programm *konv2.pro*

;optimiertes Programm zum Ausschneiden von Intensitaeten aus konvergenten Beugungs-
;bildern mit einer Untergrundkorrektur

dimccd = 2048
kopfccd = 24
namelccd = '1za7801s.j2' ;Name des Bildes
namerccd = "

nameldec = '1za7801s.j2'
namerccd = '.unt'

nullschwelle = 30
untergrundfaktor = 4

schnittpktprimaer = ftarr(2,1)
schnittpkt1 = ftarr(2,1)
schnittpkt2=ftarr(2,1)

schnittpktprimaer(0) = 915
schnittpktprimaer(1) = 981

schnittpkt1(0) = 1041
schnittpkt1(1) = 1015

schnittpkt2(0) =856
schnittpkt2(1) =1155.5

durchmesser = 14 ;Kreisdurchmesser

verschiebungx = 0
verschiebungy = 0

ringejanein = 0 ; Erzeugung der Ringe 0 = NEIN, 1 = JA
unterkorr = 0 ; Untergrundkorrektur? 0 = NEIN, 1 = JA

durchmesserunter1 = 110
durchmesserunter2 = 90

untergrossdurchmesser1 = 10
untergrossdurchmesser2 = 30

gesamtuntergrundint = 0
dimdec = dimccd
kopfdec = kopfccd

; 1. Unterteilung des GATAN-Formats

name = namelccd

```
get_lun, unit
openr, unit, name
status = fstat(unit)
groesse = status.size
schwanzlaenge = groesse - 24. - ( float(dimccd) )^2*2
schwanzstart = 24. + 2*float(dimccd)^2
schwanz0 = assoc(unit, bytarr(schwanzlaenge), schwanzstart)
schwanz = schwanz0(0)
```

```

kopf0 = assoc(unit, bytarr(kopfccd) )
kopf = kopf0(0)
inhalt = assoc(unit, intarr(dimccd,dimccd),kopfccd)
bild = inhalt(0)
close, unit
free_lun, unit

print, 'SWAP_ENDIAN wird benutzt.'
bild = swap_endian(bild)

```

; 2. Intensitaeten im Beugungsbild suchen

;Bestimmung der Pixel mit Untergrundintensitaet

```

untergrund = bild le nullschwelle ; 51 Bilderzeugung mit Werten 1 und 0 je nach Pixelwert
positionenunter = where(untergrund) ; Kodierte x und y-Koordinaten
nuntergrund = (size(positionenunter))(1); Anzahl Pixel kleiner Nullschwelle
yuntergrund = positionenunter / dimccd ; Dekodierung der Eintraege fuer y
xuntergrund = positionenunter mod dimccd; und y

```

```

bildunterint = bild(xuntergrund,yuntergrund);Bildintensitaet der einzelnen Pixel
summebildunter = total(bildunterint) ;gesamte Bildintensitaet
mittelunter = summebildunter/nuntergrund ;Mittelwert des Untergrundes
fehler = meanabsdev(bildunterint) ;Mittlere Abweichung des Untergrundes
untergneg = mittelunter - fehler
untergpos = mittelunter + fehler

```

```

print, 'Der Schnittpunktprimaer liegt bei x = ', schnittpktprimaer(0), $
' und y = ', schnittpktprimaer(1), '!

```

```

print, 'Der Schnittpunkt1 liegt bei x = ', schnittpkt1(0), $
' und y = ', schnittpkt1(1), '!

```

```

print, 'Der Schnittpunkt2 liegt bei x = ', schnittpkt2(0), $
' und y = ', schnittpkt2(1), '!

```

;Aufstellen des Indizierungsgitter

```

vektorreflex1 = schnittpkt1 - schnittpktprimaer
vektorreflex2 = schnittpkt2 - schnittpktprimaer

```

```

schnittpkt3 = schnittpkt2 + vektorreflex2
schnittpkt4 = schnittpkt3 + vektorreflex2
schnittpkt5 = schnittpkt4 + vektorreflex2
schnittpkt6 = schnittpkt5 + vektorreflex2
schnittpkt7 = schnittpkt6 + vektorreflex2
schnittpkt8 = schnittpkt7 + vektorreflex2
schnittpkt9 = schnittpkt8 + vektorreflex2
schnittpkt10 = schnittpkt9 + vektorreflex2
schnittpkt11 = schnittpkt10 + vektorreflex2
schnittpkt12 = schnittpkt11 + vektorreflex2
schnittpkt13 = schnittpkt12 + vektorreflex2
schnittpkt14 = schnittpkt13 + vektorreflex2
schnittpkt15 = schnittpkt14 + vektorreflex2
schnittpkt16 = schnittpkt15 + vektorreflex2
schnittpkt17 = schnittpkt16 + vektorreflex2
schnittpkt18 = schnittpkt17 + vektorreflex2

```

```

schnittpktm2 = schnittpktprimaer - vektorreflex2

```

schnittpktm3 = schnittpktm2 - vektorreflex2
schnittpktm4 = schnittpktm3 - vektorreflex2
schnittpktm5 = schnittpktm4 - vektorreflex2
schnittpktm6 = schnittpktm5 - vektorreflex2
schnittpktm7 = schnittpktm6 - vektorreflex2
schnittpktm8 = schnittpktm7 - vektorreflex2
schnittpktm9 = schnittpktm8 - vektorreflex2
schnittpktm10 = schnittpktm9 - vektorreflex2
schnittpktm11 = schnittpktm10 - vektorreflex2
schnittpktm12 = schnittpktm11 - vektorreflex2
schnittpktm13 = schnittpktm12 - vektorreflex2
schnittpktm14 = schnittpktm13 - vektorreflex2
schnittpktm15 = schnittpktm14 - vektorreflex2
schnittpktm16 = schnittpktm15 - vektorreflex2
schnittpktm17 = schnittpktm16 - vektorreflex2
schnittpktm18 = schnittpktm17 - vektorreflex2

vektorr1 = fltarr(2,42)
vektorr2 = fltarr(2,42)
vektorr3 = fltarr(2,42)
vektorr4 = fltarr(2,42)
vektorr5 = fltarr(2,42)
vektorr6 = fltarr(2,42)
vektorr7 = fltarr(2,42)
vektorr8 = fltarr(2,42)
vektorr9 = fltarr(2,42)
vektorr10 = fltarr(2,42)
vektorr11 = fltarr(2,42)
vektorr12 = fltarr(2,42)
vektorr13 = fltarr(2,42)
vektorr14 = fltarr(2,42)
vektorr15 = fltarr(2,42)
vektorr16 = fltarr(2,42)
vektorr17 = fltarr(2,42)
vektorr18 = fltarr(2,42)

vektorm2 = fltarr(2,42)
vektorm3 = fltarr(2,42)
vektorm4 = fltarr(2,42)
vektorm5 = fltarr(2,42)
vektorm6 = fltarr(2,42)
vektorm7 = fltarr(2,42)
vektorm8 = fltarr(2,42)
vektorm9 = fltarr(2,42)
vektorm10 = fltarr(2,42)
vektorm11 = fltarr(2,42)
vektorm12 = fltarr(2,42)
vektorm13 = fltarr(2,42)
vektorm14 = fltarr(2,42)
vektorm15 = fltarr(2,42)
vektorm16 = fltarr(2,42)
vektorm17 = fltarr(2,42)
vektorm18 = fltarr(2,42)

multi = -20

for lauf = 0, 80, 2 do begin
vektorr1(lauf) = schnittpktprimaer + multi * vektorreflex1 ;R1-Reihe
vektorr2(lauf) = schnittpkt2 + multi * vektorreflex1 ;R2-Reihe
vektorr3(lauf) = schnittpkt3 + multi * vektorreflex1 ;R3-Reihe
vektorr4(lauf) = schnittpkt4 + multi * vektorreflex1 ;R4-Reihe
vektorr5(lauf) = schnittpkt5 + multi * vektorreflex1 ;R5-Reihe

```

vektorr6(lauf) = schnittpkt6 + multi * vektorreflex1 ;R6-Reihe
vektorr7(lauf) = schnittpkt7 + multi * vektorreflex1 ;R7-Reihe
vektorr8(lauf) = schnittpkt8 + multi * vektorreflex1 ;R8-Reihe
vektorr9(lauf) = schnittpkt9 + multi * vektorreflex1 ;R9-Reihe
vektorr10(lauf)= schnittpkt10 + multi * vektorreflex1 ;R10-Reihe
vektorr11(lauf)= schnittpkt11 + multi * vektorreflex1 ;R11-Reihe
vektorr12(lauf)= schnittpkt12 + multi * vektorreflex1 ;R12-Reihe
vektorr13(lauf)= schnittpkt13 + multi * vektorreflex1 ;R13-Reihe
vektorr14(lauf)= schnittpkt14 + multi * vektorreflex1 ;R14-Reihe
vektorr15(lauf)= schnittpkt15 + multi * vektorreflex1 ;R15-Reihe
vektorr16(lauf)= schnittpkt16 + multi * vektorreflex1 ;R16-Reihe
vektorr17(lauf)= schnittpkt17 + multi * vektorreflex1 ;R17-Reihe
vektorr18(lauf)= schnittpkt18 + multi * vektorreflex1 ;R18-Reihe

```

```

vektorm2(lauf)= schnittpktm2 + multi * vektorreflex1 ;R-2-Reihe
vektorm3(lauf)= schnittpktm3 + multi * vektorreflex1 ;R-3-Reihe
vektorm4(lauf)= schnittpktm4 + multi * vektorreflex1 ;R-4-Reihe
vektorm5(lauf)= schnittpktm5 + multi * vektorreflex1 ;R-5-Reihe
vektorm6(lauf)= schnittpktm6 + multi * vektorreflex1 ;R-6-Reihe
vektorm7(lauf)= schnittpktm7 + multi * vektorreflex1 ;R-7-Reihe
vektorm8(lauf)= schnittpktm8 + multi * vektorreflex1 ;R-8-Reihe
vektorm9(lauf)= schnittpktm9 + multi * vektorreflex1 ;R-9-Reihe
vektorm10(lauf)= schnittpktm10 + multi * vektorreflex1 ;R-10-Reihe
vektorm11(lauf)= schnittpktm11 + multi * vektorreflex1 ;R-11-Reihe
vektorm12(lauf)= schnittpktm12 + multi * vektorreflex1 ;R-12-Reihe
vektorm13(lauf)= schnittpktm13 + multi * vektorreflex1 ;R-13-Reihe
vektorm14(lauf)= schnittpktm14 + multi * vektorreflex1 ;R-14-Reihe
vektorm15(lauf)= schnittpktm15 + multi * vektorreflex1 ;R-15-Reihe
vektorm16(lauf)= schnittpktm16 + multi * vektorreflex1 ;R-16-Reihe
vektorm17(lauf)= schnittpktm17 + multi * vektorreflex1 ;R-17-Reihe
vektorm18(lauf)= schnittpktm18 + multi * vektorreflex1 ;R-18-Reihe
multi = multi + 1
endfor

```

;Addition des Verschiebungsvektors

```

verschieb = fltarr(2,42)
for lauf = 0, 81, 2 do begin
    verschieb(lauf) = verschiebungx
    verschieb(lauf+1) = verschiebungy
endfor

```

```

vektorr1 = vektorr1 + verschieb
vektorr2 = vektorr2 + verschieb
vektorr3 = vektorr3 + verschieb
vektorr4 = vektorr4 + verschieb
vektorr5 = vektorr5 + verschieb
vektorr6 = vektorr6 + verschieb
vektorr7 = vektorr7 + verschieb
vektorr8 = vektorr8 + verschieb
vektorr9 = vektorr9 + verschieb
vektorr10 = vektorr10 + verschieb
vektorr11 = vektorr11 + verschieb
vektorr12 = vektorr12 + verschieb
vektorr13 = vektorr13 + verschieb
vektorr14 = vektorr14 + verschieb
vektorr15 = vektorr15 + verschieb
vektorr16 = vektorr16 + verschieb
vektorr17 = vektorr17 + verschieb
vektorr18 = vektorr18 + verschieb

```

```

vektorm2 = vektorm2 + verschieb

```

vektorm3 = vektorm3 + verschieb
vektorm4 = vektorm4 + verschieb
vektorm5 = vektorm5 + verschieb
vektorm6 = vektorm6 + verschieb
vektorm7 = vektorm7 + verschieb
vektorm8 = vektorm8 + verschieb
vektorm9 = vektorm9 + verschieb
vektorm10 = vektorm10 + verschieb
vektorm11 = vektorm11 + verschieb
vektorm12 = vektorm12 + verschieb
vektorm13 = vektorm13 + verschieb
vektorm14 = vektorm14 + verschieb
vektorm15 = vektorm15 + verschieb
vektorm16 = vektorm16 + verschieb
vektorm17 = vektorm17 + verschieb
vektorm18 = vektorm18 + verschieb

; Herausfinden der nuetzlichen Positionen

vektorlow = vektorr1 ge 0
vektorhigh = vektorr1 le dimccd
vektormulti = vektorlow * vektorhigh
vektorr1a = vektorr1 * vektormulti

vektorlow = vektorr2 ge 0
vektorhigh = vektorr2 le dimccd
vektormulti = vektorlow * vektorhigh
vektorr2a = vektorr2 * vektormulti

vektorlow = vektorr3 ge 0
vektorhigh = vektorr3 le dimccd
vektormulti = vektorlow * vektorhigh
vektorr3a = vektorr3 * vektormulti

vektorlow = vektorr4 ge 0
vektorhigh = vektorr4 le dimccd
vektormulti = vektorlow * vektorhigh
vektorr4a = vektorr4 * vektormulti

vektorlow = vektorr5 ge 0
vektorhigh = vektorr5 le dimccd
vektormulti = vektorlow * vektorhigh
vektorr5a = vektorr5 * vektormulti

vektorlow = vektorr6 ge 0
vektorhigh = vektorr6 le dimccd
vektormulti = vektorlow * vektorhigh
vektorr6a = vektorr6 * vektormulti

vektorlow = vektorr7 ge 0
vektorhigh = vektorr7 le dimccd
vektormulti = vektorlow * vektorhigh
vektorr7a = vektorr7 * vektormulti

vektorlow = vektorr8 ge 0
vektorhigh = vektorr8 le dimccd
vektormulti = vektorlow * vektorhigh
vektorr8a = vektorr8 * vektormulti

vektorlow = vektorr9 ge 0
vektorhigh = vektorr9 le dimccd
vektormulti = vektorlow * vektorhigh

vektorr9a	= vektorr9 * vektormulti
vektorlow	= vektorr10 ge 0
vektorhigh	= vektorr10 le dimccd
vektormulti	= vektorlow * vektorhigh
vektorr10a	= vektorr10 * vektormulti
vektorlow	= vektorr11 ge 0
vektorhigh	= vektorr11 le dimccd
vektormulti	= vektorlow * vektorhigh
vektorr11a	= vektorr11 * vektormulti
vektorlow	= vektorr12 ge 0
vektorhigh	= vektorr12 le dimccd
vektormulti	= vektorlow * vektorhigh
vektorr12a	= vektorr12 * vektormulti
vektorlow	= vektorr13 ge 0
vektorhigh	= vektorr13 le dimccd
vektormulti	= vektorlow * vektorhigh
vektorr13a	= vektorr13 * vektormulti
vektorlow	= vektorr14 ge 0
vektorhigh	= vektorr14 le dimccd
vektormulti	= vektorlow * vektorhigh
vektorr14a	= vektorr14 * vektormulti
vektorlow	= vektorr15 ge 0
vektorhigh	= vektorr15 le dimccd
vektormulti	= vektorlow * vektorhigh
vektorr15a	= vektorr15 * vektormulti
vektorlow	= vektorr16 ge 0
vektorhigh	= vektorr16 le dimccd
vektormulti	= vektorlow * vektorhigh
vektorr16a	= vektorr16 * vektormulti
vektorlow	= vektorr17 ge 0
vektorhigh	= vektorr17 le dimccd
vektormulti	= vektorlow * vektorhigh
vektorr17a	= vektorr17 * vektormulti
vektorlow	= vektorr18 ge 0
vektorhigh	= vektorr18 le dimccd
vektormulti	= vektorlow * vektorhigh
vektorr18a	= vektorr18 * vektormulti
vektorlow	= vektorrm2 ge 0
vektorhigh	= vektorrm2 le dimccd
vektormulti	= vektorlow * vektorhigh
vektorrm2a	= vektorrm2 * vektormulti
vektorlow	= vektorrm3 ge 0
vektorhigh	= vektorrm3 le dimccd
vektormulti	= vektorlow * vektorhigh
vektorrm3a	= vektorrm3 * vektormulti
vektorlow	= vektorrm4 ge 0
vektorhigh	= vektorrm4 le dimccd
vektormulti	= vektorlow * vektorhigh
vektorrm4a	= vektorrm4 * vektormulti

vektorlow = vektorrm5 ge 0
vektorhigh = vektorrm5 le dimccd
vektormulti = vektorlow * vektorhigh
vektorrm5a = vektorrm5 * vektormulti

vektorlow = vektorrm6 ge 0
vektorhigh = vektorrm6 le dimccd
vektormulti = vektorlow * vektorhigh
vektorrm6a = vektorrm6 * vektormulti

vektorlow = vektorrm7 ge 0
vektorhigh = vektorrm7 le dimccd
vektormulti = vektorlow * vektorhigh
vektorrm7a = vektorrm7 * vektormulti

vektorlow = vektorrm8 ge 0
vektorhigh = vektorrm8 le dimccd
vektormulti = vektorlow * vektorhigh
vektorrm8a = vektorrm8 * vektormulti

vektorlow = vektorrm9 ge 0
vektorhigh = vektorrm9 le dimccd
vektormulti = vektorlow * vektorhigh
vektorrm9a = vektorrm9 * vektormulti

vektorlow = vektorrm10 ge 0
vektorhigh = vektorrm10 le dimccd
vektormulti = vektorlow * vektorhigh
vektorrm10a = vektorrm10 * vektormulti

vektorlow = vektorrm11 ge 0
vektorhigh = vektorrm11 le dimccd
vektormulti = vektorlow * vektorhigh
vektorrm11a = vektorrm11 * vektormulti

vektorlow = vektorrm12 ge 0
vektorhigh = vektorrm12 le dimccd
vektormulti = vektorlow * vektorhigh
vektorrm12a = vektorrm12 * vektormulti

vektorlow = vektorrm13 ge 0
vektorhigh = vektorrm13 le dimccd
vektormulti = vektorlow * vektorhigh
vektorrm13a = vektorrm13 * vektormulti

vektorlow = vektorrm14 ge 0
vektorhigh = vektorrm14 le dimccd
vektormulti = vektorlow * vektorhigh
vektorrm14a = vektorrm14 * vektormulti

vektorlow = vektorrm15 ge 0
vektorhigh = vektorrm15 le dimccd
vektormulti = vektorlow * vektorhigh
vektorrm15a = vektorrm15 * vektormulti

vektorlow = vektorrm16 ge 0
vektorhigh = vektorrm16 le dimccd
vektormulti = vektorlow * vektorhigh
vektorrm16a = vektorrm16 * vektormulti

vektorlow = vektorrm17 ge 0
vektorhigh = vektorrm17 le dimccd

vektormulti = vektorlow * vektorhigh
vektorm17a = vektorm17 * vektormulti

vektorlow = vektorm18 ge 0
vektorhigh = vektorm18 le dimccd
vektormulti = vektorlow * vektorhigh
vektorm18a = vektorm18 * vektormulti

;Herausfiltern der nuetzlichen Positionen

```
for lauf = 0, 80, 2 do begin
  if vektorr1a(lauf) eq 0 then vektorr1a(lauf+1) = 0
  if vektorr2a(lauf) eq 0 then vektorr2a(lauf+1) = 0
  if vektorr3a(lauf) eq 0 then vektorr3a(lauf+1) = 0
  if vektorr4a(lauf) eq 0 then vektorr4a(lauf+1) = 0
  if vektorr5a(lauf) eq 0 then vektorr5a(lauf+1) = 0
  if vektorr6a(lauf) eq 0 then vektorr6a(lauf+1) = 0
  if vektorr7a(lauf) eq 0 then vektorr7a(lauf+1) = 0
  if vektorr8a(lauf) eq 0 then vektorr8a(lauf+1) = 0
  if vektorr9a(lauf) eq 0 then vektorr9a(lauf+1) = 0
  if vektorr10a(lauf) eq 0 then vektorr10a(lauf+1) = 0
  if vektorr11a(lauf) eq 0 then vektorr11a(lauf+1) = 0
  if vektorr12a(lauf) eq 0 then vektorr12a(lauf+1) = 0
  if vektorr13a(lauf) eq 0 then vektorr13a(lauf+1) = 0
  if vektorr14a(lauf) eq 0 then vektorr14a(lauf+1) = 0
  if vektorr15a(lauf) eq 0 then vektorr15a(lauf+1) = 0
  if vektorr16a(lauf) eq 0 then vektorr16a(lauf+1) = 0
  if vektorr17a(lauf) eq 0 then vektorr17a(lauf+1) = 0
  if vektorr18a(lauf) eq 0 then vektorr18a(lauf+1) = 0
  if vektorm2a(lauf) eq 0 then vektorm2a(lauf+1) = 0
  if vektorm3a(lauf) eq 0 then vektorm3a(lauf+1) = 0
  if vektorm4a(lauf) eq 0 then vektorm4a(lauf+1) = 0
  if vektorm5a(lauf) eq 0 then vektorm5a(lauf+1) = 0
  if vektorm6a(lauf) eq 0 then vektorm6a(lauf+1) = 0
  if vektorm7a(lauf) eq 0 then vektorm7a(lauf+1) = 0
  if vektorm8a(lauf) eq 0 then vektorm8a(lauf+1) = 0
  if vektorm9a(lauf) eq 0 then vektorm9a(lauf+1) = 0
  if vektorm10a(lauf) eq 0 then vektorm10a(lauf+1) = 0
  if vektorm11a(lauf) eq 0 then vektorm11a(lauf+1) = 0
  if vektorm12a(lauf) eq 0 then vektorm12a(lauf+1) = 0
  if vektorm13a(lauf) eq 0 then vektorm13a(lauf+1) = 0
  if vektorm14a(lauf) eq 0 then vektorm14a(lauf+1) = 0
  if vektorm15a(lauf) eq 0 then vektorm15a(lauf+1) = 0
  if vektorm16a(lauf) eq 0 then vektorm16a(lauf+1) = 0
  if vektorm17a(lauf) eq 0 then vektorm17a(lauf+1) = 0
  if vektorm18a(lauf) eq 0 then vektorm18a(lauf+1) = 0

  if vektorr1a(lauf+1) eq 0 then vektorr1a(lauf) = 0
  if vektorr2a(lauf+1) eq 0 then vektorr2a(lauf) = 0
  if vektorr3a(lauf+1) eq 0 then vektorr3a(lauf) = 0
  if vektorr4a(lauf+1) eq 0 then vektorr4a(lauf) = 0
  if vektorr5a(lauf+1) eq 0 then vektorr5a(lauf) = 0
  if vektorr6a(lauf+1) eq 0 then vektorr6a(lauf) = 0
  if vektorr7a(lauf+1) eq 0 then vektorr7a(lauf) = 0
  if vektorr8a(lauf+1) eq 0 then vektorr8a(lauf) = 0
  if vektorr9a(lauf+1) eq 0 then vektorr9a(lauf) = 0
  if vektorr10a(lauf+1) eq 0 then vektorr10a(lauf) = 0
  if vektorr11a(lauf+1) eq 0 then vektorr11a(lauf) = 0
  if vektorr12a(lauf+1) eq 0 then vektorr12a(lauf) = 0
  if vektorr13a(lauf+1) eq 0 then vektorr13a(lauf) = 0
  if vektorr14a(lauf+1) eq 0 then vektorr14a(lauf) = 0
  if vektorr15a(lauf+1) eq 0 then vektorr15a(lauf) = 0
```

```
if vektorr16a(lauf+1) eq 0 then vektorr16a(lauf) = 0
if vektorr17a(lauf+1) eq 0 then vektorr17a(lauf) = 0
if vektorr18a(lauf+1) eq 0 then vektorr18a(lauf) = 0
if vektorrm2a(lauf+1) eq 0 then vektorrm2a(lauf) = 0
if vektorrm3a(lauf+1) eq 0 then vektorrm3a(lauf) = 0
if vektorrm4a(lauf+1) eq 0 then vektorrm4a(lauf) = 0
if vektorrm5a(lauf+1) eq 0 then vektorrm5a(lauf) = 0
if vektorrm6a(lauf+1) eq 0 then vektorrm6a(lauf) = 0
if vektorrm7a(lauf+1) eq 0 then vektorrm7a(lauf) = 0
if vektorrm8a(lauf+1) eq 0 then vektorrm8a(lauf) = 0
if vektorrm9a(lauf+1) eq 0 then vektorrm9a(lauf) = 0
if vektorrm10a(lauf+1) eq 0 then vektorrm10a(lauf) = 0
if vektorrm11a(lauf+1) eq 0 then vektorrm11a(lauf) = 0
if vektorrm12a(lauf+1) eq 0 then vektorrm12a(lauf) = 0
if vektorrm13a(lauf+1) eq 0 then vektorrm13a(lauf) = 0
if vektorrm14a(lauf+1) eq 0 then vektorrm14a(lauf) = 0
if vektorrm15a(lauf+1) eq 0 then vektorrm15a(lauf) = 0
if vektorrm16a(lauf+1) eq 0 then vektorrm16a(lauf) = 0
if vektorrm17a(lauf+1) eq 0 then vektorrm17a(lauf) = 0
if vektorrm18a(lauf+1) eq 0 then vektorrm18a(lauf) = 0
```

endfor

```
vektor1b      = where(vektor1a, anzahl1)
if anzahl1 eq 0 then goto, eins
vektor1b      = vektor1a(vektor1b)
```

eins:

```
vektor2b      = where(vektor2a, anzahl2)
if anzahl2 eq 0 then goto, zwei
vektor2b      = vektor2a(vektor2b)
```

zwei:

```
vektor3b      = where(vektor3a, anzahl3)
if anzahl3 eq 0 then goto, drei
vektor3b      = vektor3a(vektor3b)
```

drei:

```
vektor4b      = where(vektor4a, anzahl4)
if anzahl4 eq 0 then goto, vier
vektor4b      = vektor4a(vektor4b)
```

vier:

```
vektor5b      = where(vektor5a, anzahl5)
if anzahl5 eq 0 then goto, fuenf
vektor5b      = vektor5a(vektor5b)
```

fuenf:

```
vektorrm2b    = where(vektorrm2a, anzahlm2)
if anzahlm2 eq 0 then goto, sechs
vektorrm2b    = vektorrm2a(vektorrm2b)
```

sechs:

```
vektorrm3b    = where(vektorrm3a, anzahlm3)
if anzahlm3 eq 0 then goto, sieben
vektorrm3b    = vektorrm3a(vektorrm3b)
```

sieben:

vektorm4b = where(vektorm4a, anzahlm4)
if anzahlm4 eq 0 then goto, acht
vektorm4b = vektorm4a(vektorm4b)

acht:

vektorm5b = where(vektorm5a, anzahlm5)
if anzahlm5 eq 0 then goto, acht1
vektorm5b = vektorm5a(vektorm5b)

acht1:

vektorm6b = where(vektorm6a, anzahlm6)
if anzahlm6 eq 0 then goto, acht2
vektorm6b = vektorm6a(vektorm6b)

acht2:

vektorm7b = where(vektorm7a, anzahlm7)
if anzahlm7 eq 0 then goto, acht3
vektorm7b = vektorm7a(vektorm7b)

acht3:

vektorm8b = where(vektorm8a, anzahlm8)
if anzahlm8 eq 0 then goto, acht4
vektorm8b = vektorm8a(vektorm8b)

acht4:

vektorm9b = where(vektorm9a, anzahlm9)
if anzahlm9 eq 0 then goto, acht5
vektorm9b = vektorm9a(vektorm9b)

acht5:

vektorm10b = where(vektorm10a, anzahlm10)
if anzahlm10 eq 0 then goto, acht6
vektorm10b = vektorm10a(vektorm10b)

acht6:

vektorm11b = where(vektorm11a, anzahlm11)
if anzahlm11 eq 0 then goto, acht7
vektorm11b = vektorm11a(vektorm11b)

acht7:

vektorm12b = where(vektorm12a, anzahlm12)
if anzahlm12 eq 0 then goto, acht8
vektorm12b = vektorm12a(vektorm12b)

acht8:

vektorm13b = where(vektorm13a, anzahlm13)
if anzahlm13 eq 0 then goto, acht9
vektorm13b = vektorm13a(vektorm13b)

acht9:

```
vektorm14b = where(vektorm14a, anzahlm14)
if anzahlm14 eq 0 then goto, acht10
vektorm14b = vektorm14a(vektorm14b)
```

acht10:

```
vektorm15b = where(vektorm15a, anzahlm15)
if anzahlm15 eq 0 then goto, acht11
vektorm15b = vektorm15a(vektorm15b)
```

acht11:

```
vektorm16b = where(vektorm16a, anzahlm16)
if anzahlm16 eq 0 then goto, acht12
vektorm16b = vektorm16a(vektorm16b)
```

acht12:

```
vektorm17b = where(vektorm17a, anzahlm17)
if anzahlm17 eq 0 then goto, acht13
vektorm17b = vektorm17a(vektorm17b)
```

acht13:

```
vektorm18b = where(vektorm18a, anzahlm18)
if anzahlm18 eq 0 then goto, neun
vektorm18b = vektorm18a(vektorm18b)
```

neun:

```
vektorr6b = where(vektorr6a, anzahl6)
if anzahl6 eq 0 then goto, zehn
vektorr6b = vektorr6a(vektorr6b)
```

zehn:

```
vektorr7b = where(vektorr7a, anzahl7)
if anzahl7 eq 0 then goto, elf
vektorr7b = vektorr7a(vektorr7b)
```

elf:

```
vektorr8b = where(vektorr8a, anzahl8)
if anzahl8 eq 0 then goto, elf1
vektorr8b = vektorr8a(vektorr8b)
```

elf1:

```
vektorr9b = where(vektorr9a, anzahl9)
if anzahl9 eq 0 then goto, elf2
vektorr9b = vektorr9a(vektorr9b)
```

elf2:

```
vektorr10b = where(vektorr10a, anzahl10)
if anzahl10 eq 0 then goto, elf3
vektorr10b = vektorr10a(vektorr10b)
```

elf3:

```
vektorr11b = where(vektorr11a, anzahl11)
if anzahl11 eq 0 then goto, elf4
```

```

vektorr11b    = vektorr11a(vektorr11b)

elf4:

vektorr12b    = where(vektorr12a, anzahl12)
if anzahl12 eq 0 then goto, elf5
vektorr12b    = vektorr12a(vektorr12b)

elf5:

vektorr13b    = where(vektorr13a, anzahl13)
if anzahl13 eq 0 then goto, elf6
vektorr13b    = vektorr13a(vektorr13b)

elf6:

vektorr14b    = where(vektorr14a, anzahl14)
if anzahl14 eq 0 then goto, elf7
vektorr14b    = vektorr14a(vektorr14b)

elf7:

vektorr15b    = where(vektorr15a, anzahl15)
if anzahl15 eq 0 then goto, elf8
vektorr15b    = vektorr15a(vektorr15b)

elf8:

vektorr16b    = where(vektorr16a, anzahl16)
if anzahl16 eq 0 then goto, elf9
vektorr16b    = vektorr16a(vektorr16b)

elf9:

vektorr17b    = where(vektorr17a, anzahl17)
if anzahl17 eq 0 then goto, elf10
vektorr17b    = vektorr17a(vektorr17b)

elf10:

vektorr18b    = where(vektorr18a, anzahl18)
if anzahl18 eq 0 then goto, elf11
vektorr18b    = vektorr18a(vektorr18b)

elf11:

print, 'test'
; Erzeugung des Kreises

k      = indgen(durchmesser,durchmesser)
kx     = (k mod durchmesser) - durchmesser/2
ky     = (k/durchmesser) - durchmesser/2
kreis  = (kx^2 + ky^2) le (durchmesser/2)^2

;Dieser Kreis wird auf die Koordinaten der Reflexe gebracht

kreisbereich  = fitarr(dimccd,dimccd)

;tvscI, congrid(kreisbereich,512,512)

for lauf = 0, anzahl1-1, 2 do begin
    if vektorr1b(lauf) - (durchmesser/2) lt 1 then goto, zwoelf

```

```
    if vektorr1b(lauf+1) - (durchmesser/2) lt 1 then goto, zwoelf
    if vektorr1b(lauf) + (durchmesser/2) gt dimccd then goto, zwoelf
    if vektorr1b(lauf+1) + (durchmesser/2) gt dimccd then goto, zwoelf
    kreisbereich(vektorr1b(lauf)-(durchmesser/2):(vektorr1b(lauf)$
    +(durchmesser/2)-1),vektorr1b(lauf+1)-(durchmesser/2):$
    (vektorr1b(lauf+1)+(durchmesser/2)-1))=kreis
    print, vektorr1b(lauf), vektorr1b(lauf+1)
    zwoelf:
    ;tvsc1, congrid(kreisbereich,512,512)
    print, vektorr1b(lauf), vektorr1b(lauf+1)
endfor

tvsc1, congrid(kreisbereich,512,512)

for lauf = 0, anzahl2-1, 2 do begin
    if vektorr2b(lauf) - (durchmesser/2) lt 1 then goto, dreizehn
    if vektorr2b(lauf+1) - (durchmesser/2) lt 1 then goto, dreizehn
    if vektorr2b(lauf) + (durchmesser/2) gt dimccd then goto, dreizehn
    if vektorr2b(lauf+1) + (durchmesser/2) gt dimccd then goto, dreizehn
    kreisbereich(vektorr2b(lauf)-(durchmesser/2):(vektorr2b(lauf)$
    +(durchmesser/2)-1),vektorr2b(lauf+1)-(durchmesser/2):$
    (vektorr2b(lauf+1)+(durchmesser/2)-1))=kreis
    print, vektorr2b(lauf), vektorr2b(lauf+1)
    dreizehn:
    ;tvsc1, congrid(kreisbereich,512,512)
    print, vektorr2b(lauf), vektorr2b(lauf+1)
endfor

tvsc1, congrid(kreisbereich,512,512)

for lauf = 0, anzahl3-1, 2 do begin
    if vektorr3b(lauf) - (durchmesser/2) lt 1 then goto, vierzehn
    if vektorr3b(lauf+1) - (durchmesser/2) lt 1 then goto, vierzehn
    if vektorr3b(lauf) + (durchmesser/2) gt dimccd then goto, vierzehn
    if vektorr3b(lauf+1) + (durchmesser/2) gt dimccd then goto, vierzehn
    kreisbereich(vektorr3b(lauf)-(durchmesser/2):(vektorr3b(lauf)$
    +(durchmesser/2)-1),vektorr3b(lauf+1)-(durchmesser/2):$
    (vektorr3b(lauf+1)+(durchmesser/2)-1))=kreis
    print, vektorr3b(lauf), vektorr3b(lauf+1)
    vierzehn:
    ;tvsc1, congrid(kreisbereich,512,512)
    print, vektorr3b(lauf), vektorr3b(lauf+1)
endfor

tvsc1, congrid(kreisbereich,512,512)

for lauf = 0, anzahl4-1, 2 do begin
    if vektorr4b(lauf) - (durchmesser/2) lt 1 then goto, fuenfzehn
    if vektorr4b(lauf+1) - (durchmesser/2) lt 1 then goto, fuenfzehn
    if vektorr4b(lauf) + (durchmesser/2) gt dimccd then goto, fuenfzehn
    if vektorr4b(lauf+1) + (durchmesser/2) gt dimccd then goto, fuenfzehn
    kreisbereich(vektorr4b(lauf)-(durchmesser/2):(vektorr4b(lauf)$
    +(durchmesser/2)-1),vektorr4b(lauf+1)-(durchmesser/2):$
    (vektorr4b(lauf+1)+(durchmesser/2)-1))=kreis
    print, vektorr4b(lauf), vektorr4b(lauf+1)
    fuenfzehn:
    ;tvsc1, congrid(kreisbereich,512,512)
    print, vektorr4b(lauf), vektorr4b(lauf+1)
endfor

tvsc1, congrid(kreisbereich,512,512)
```

```

for lauf = 0, anzahl5-1, 2 do begin
  if vektorr5b(lauf) - (durchmesser/2) lt 1 then goto, sechzehn
  if vektorr5b(lauf+1) - (durchmesser/2) lt 1 then goto, sechzehn
  if vektorr5b(lauf) + (durchmesser/2) gt dimccd then goto, sechzehn
  if vektorr5b(lauf+1) + (durchmesser/2) gt dimccd then goto, sechzehn
  kreisbereich(vektorr5b(lauf)-(durchmesser/2):(vektorr5b(lauf)$
+(durchmesser/2)-1),vektorr5b(lauf+1)-(durchmesser/2):$
(vektorr5b(lauf+1)+(durchmesser/2)-1))=kreis
  print, vektorr5b(lauf), vektorr5b(lauf+1)
  sechzehn:
  ;tvsc1, congrid(kreisbereich,512,512)
  print, vektorr5b(lauf), vektorr5b(lauf+1)
endfor

```

```
tvsc1, congrid(kreisbereich,512,512)
```

```

for lauf = 0, anzahl6-1, 2 do begin
  if vektorr6b(lauf) - (durchmesser/2) lt 1 then goto, siebzehn
  if vektorr6b(lauf+1) - (durchmesser/2) lt 1 then goto, siebzehn
  if vektorr6b(lauf) + (durchmesser/2) gt dimccd then goto, siebzehn
  if vektorr6b(lauf+1) + (durchmesser/2) gt dimccd then goto, siebzehn
  kreisbereich(vektorr6b(lauf)-(durchmesser/2):(vektorr6b(lauf)$
+(durchmesser/2)-1),vektorr6b(lauf+1)-(durchmesser/2):$
(vektorr6b(lauf+1)+(durchmesser/2)-1))=kreis
  print, vektorr6b(lauf), vektorr6b(lauf+1)
  siebzehn:
  ;tvsc1, congrid(kreisbereich,512,512)
  print, vektorr6b(lauf), vektorr6b(lauf+1)
endfor

```

```
tvsc1, congrid(kreisbereich,512,512)
```

```

for lauf = 0, anzahl7-1, 2 do begin
  if vektorr7b(lauf) - (durchmesser/2) lt 1 then goto, achtzehn
  if vektorr7b(lauf+1) - (durchmesser/2) lt 1 then goto, achtzehn
  if vektorr7b(lauf) + (durchmesser/2) gt dimccd then goto, achtzehn
  if vektorr7b(lauf+1) + (durchmesser/2) gt dimccd then goto, achtzehn
  kreisbereich(vektorr7b(lauf)-(durchmesser/2):(vektorr7b(lauf)$
+(durchmesser/2)-1),vektorr7b(lauf+1)-(durchmesser/2):$
(vektorr7b(lauf+1)+(durchmesser/2)-1))=kreis
  print, vektorr7b(lauf), vektorr7b(lauf+1)
  achtzehn:
  ;tvsc1, congrid(kreisbereich,512,512)
  print, vektorr7b(lauf), vektorr7b(lauf+1)
endfor

```

```
tvsc1, congrid(kreisbereich,512,512)
```

```

for lauf = 0, anzahl8-1, 2 do begin
  if vektorr8b(lauf) - (durchmesser/2) lt 1 then goto, achtzehn1
  if vektorr8b(lauf+1) - (durchmesser/2) lt 1 then goto, achtzehn1
  if vektorr8b(lauf) + (durchmesser/2) gt dimccd then goto, achtzehn1
  if vektorr8b(lauf+1) + (durchmesser/2) gt dimccd then goto, achtzehn1
  kreisbereich(vektorr8b(lauf)-(durchmesser/2):(vektorr8b(lauf)$
+(durchmesser/2)-1),vektorr8b(lauf+1)-(durchmesser/2):$
(vektorr8b(lauf+1)+(durchmesser/2)-1))=kreis
  print, vektorr8b(lauf), vektorr8b(lauf+1)
  achtzehn1:
  ;tvsc1, congrid(kreisbereich,512,512)
  print, vektorr8b(lauf), vektorr8b(lauf+1)
endfor

```

```
tvsl, congrid(kreisbereich,512,512)
```

```
for lauf = 0, anzahl9-1, 2 do begin
  if vektorr9b(lauf) - (durchmesser/2) lt 1 then goto, achtzehn2
  if vektorr9b(lauf+1) - (durchmesser/2) lt 1 then goto, achtzehn2
  if vektorr9b(lauf) + (durchmesser/2) gt dimccd then goto, achtzehn2
  if vektorr9b(lauf+1) + (durchmesser/2) gt dimccd then goto, achtzehn2
  kreisbereich(vektorr9b(lauf)-(durchmesser/2):(vektorr9b(lauf)$
+(durchmesser/2)-1),vektorr9b(lauf+1)-(durchmesser/2):$
(vektorr9b(lauf+1)+(durchmesser/2)-1))=kreis
  print, vektorr9b(lauf), vektorr9b(lauf+1)
  achtzehn2:
  ;tvsl, congrid(kreisbereich,512,512)
  print, vektorr9b(lauf), vektorr9b(lauf+1)
endfor
```

```
tvsl, congrid(kreisbereich,512,512)
```

```
for lauf = 0, anzahl10-1, 2 do begin
  if vektorr10b(lauf) - (durchmesser/2) lt 1 then goto, achtzehn3
  if vektorr10b(lauf+1) - (durchmesser/2) lt 1 then goto, achtzehn3
  if vektorr10b(lauf) + (durchmesser/2) gt dimccd then goto, achtzehn3
  if vektorr10b(lauf+1) + (durchmesser/2) gt dimccd then goto, achtzehn3
  kreisbereich(vektorr10b(lauf)-(durchmesser/2):(vektorr10b(lauf)$
+(durchmesser/2)-1),vektorr10b(lauf+1)-(durchmesser/2):$
(vektorr10b(lauf+1)+(durchmesser/2)-1))=kreis
  print, vektorr10b(lauf), vektorr10b(lauf+1)
  achtzehn3:
  ;tvsl, congrid(kreisbereich,512,512)
  print, vektorr10b(lauf), vektorr10b(lauf+1)
endfor
```

```
tvsl, congrid(kreisbereich,512,512)
```

```
for lauf = 0, anzahl11-1, 2 do begin
  if vektorr11b(lauf) - (durchmesser/2) lt 1 then goto, achtzehn4
  if vektorr11b(lauf+1) - (durchmesser/2) lt 1 then goto, achtzehn4
  if vektorr11b(lauf) + (durchmesser/2) gt dimccd then goto, achtzehn4
  if vektorr11b(lauf+1) + (durchmesser/2) gt dimccd then goto, achtzehn4
  kreisbereich(vektorr11b(lauf)-(durchmesser/2):(vektorr11b(lauf)$
+(durchmesser/2)-1),vektorr11b(lauf+1)-(durchmesser/2):$
(vektorr11b(lauf+1)+(durchmesser/2)-1))=kreis
  print, vektorr11b(lauf), vektorr11b(lauf+1)
  achtzehn4:
  ;tvsl, congrid(kreisbereich,512,512)
  print, vektorr11b(lauf), vektorr11b(lauf+1)
endfor
```

```
tvsl, congrid(kreisbereich,512,512)
```

```
for lauf = 0, anzahl12-1, 2 do begin
  if vektorr12b(lauf) - (durchmesser/2) lt 1 then goto, achtzehn5
  if vektorr12b(lauf+1) - (durchmesser/2) lt 1 then goto, achtzehn5
  if vektorr12b(lauf) + (durchmesser/2) gt dimccd then goto, achtzehn5
  if vektorr12b(lauf+1) + (durchmesser/2) gt dimccd then goto, achtzehn5
  kreisbereich(vektorr12b(lauf)-(durchmesser/2):(vektorr12b(lauf)$
+(durchmesser/2)-1),vektorr12b(lauf+1)-(durchmesser/2):$
(vektorr12b(lauf+1)+(durchmesser/2)-1))=kreis
  print, vektorr12b(lauf), vektorr12b(lauf+1)
  achtzehn5:
  ;tvsl, congrid(kreisbereich,512,512)
  print, vektorr12b(lauf), vektorr12b(lauf+1)
```

```

endfor

tvsc1, congrid(kreisbereich,512,512)

for lauf = 0, anzahl13-1, 2 do begin
  if vektorr13b(lauf) - (durchmesser/2) lt 1 then goto, achtzehn6
  if vektorr13b(lauf+1) - (durchmesser/2) lt 1 then goto, achtzehn6
  if vektorr13b(lauf) + (durchmesser/2) gt dimccd then goto, achtzehn6
  if vektorr13b(lauf+1) + (durchmesser/2) gt dimccd then goto, achtzehn6
  kreisbereich(vektorr13b(lauf)-(durchmesser/2):(vektorr13b(lauf)$
+(durchmesser/2)-1),vektorr13b(lauf+1)-(durchmesser/2):$
(vektorr13b(lauf+1)+(durchmesser/2)-1))=kreis
  print, vektorr13b(lauf), vektorr13b(lauf+1)
  achtzehn6:
;tvsc1, congrid(kreisbereich,512,512)
  print, vektorr13b(lauf), vektorr13b(lauf+1)
endfor

tvsc1, congrid(kreisbereich,512,512)

for lauf = 0, anzahl14-1, 2 do begin
  if vektorr14b(lauf) - (durchmesser/2) lt 1 then goto, achtzehn7
  if vektorr14b(lauf+1) - (durchmesser/2) lt 1 then goto, achtzehn7
  if vektorr14b(lauf) + (durchmesser/2) gt dimccd then goto, achtzehn7
  if vektorr14b(lauf+1) + (durchmesser/2) gt dimccd then goto, achtzehn7
  kreisbereich(vektorr14b(lauf)-(durchmesser/2):(vektorr14b(lauf)$
+(durchmesser/2)-1),vektorr14b(lauf+1)-(durchmesser/2):$
(vektorr14b(lauf+1)+(durchmesser/2)-1))=kreis
  print, vektorr14b(lauf), vektorr14b(lauf+1)
  achtzehn7:
;tvsc1, congrid(kreisbereich,512,512)
  print, vektorr14b(lauf), vektorr14b(lauf+1)
endfor

tvsc1, congrid(kreisbereich,512,512)

for lauf = 0, anzahl15-1, 2 do begin
  if vektorr15b(lauf) - (durchmesser/2) lt 1 then goto, achtzehn8
  if vektorr15b(lauf+1) - (durchmesser/2) lt 1 then goto, achtzehn8
  if vektorr15b(lauf) + (durchmesser/2) gt dimccd then goto, achtzehn8
  if vektorr15b(lauf+1) + (durchmesser/2) gt dimccd then goto, achtzehn8
  kreisbereich(vektorr15b(lauf)-(durchmesser/2):(vektorr15b(lauf)$
+(durchmesser/2)-1),vektorr15b(lauf+1)-(durchmesser/2):$
(vektorr15b(lauf+1)+(durchmesser/2)-1))=kreis
  print, vektorr15b(lauf), vektorr15b(lauf+1)
  achtzehn8:
;tvsc1, congrid(kreisbereich,512,512)
  print, vektorr15b(lauf), vektorr15b(lauf+1)
endfor

tvsc1, congrid(kreisbereich,512,512)

for lauf = 0, anzahl16-1, 2 do begin
  if vektorr16b(lauf) - (durchmesser/2) lt 1 then goto, achtzehn9
  if vektorr16b(lauf+1) - (durchmesser/2) lt 1 then goto, achtzehn9
  if vektorr16b(lauf) + (durchmesser/2) gt dimccd then goto, achtzehn9
  if vektorr16b(lauf+1) + (durchmesser/2) gt dimccd then goto, achtzehn9
  kreisbereich(vektorr16b(lauf)-(durchmesser/2):(vektorr16b(lauf)$
+(durchmesser/2)-1),vektorr16b(lauf+1)-(durchmesser/2):$
(vektorr16b(lauf+1)+(durchmesser/2)-1))=kreis
  print, vektorr16b(lauf), vektorr16b(lauf+1)
  achtzehn9:

```

```
        ;tvsc1, congrid(kreisbereich,512,512)
        print, vektorr16b(lauf), vektorr16b(lauf+1)
    endfor

tvsc1, congrid(kreisbereich,512,512)

for lauf = 0, anzahl17-1, 2 do begin
    if vektorr17b(lauf) - (durchmesser/2) lt 1 then goto, achtzehn10
    if vektorr17b(lauf+1) - (durchmesser/2) lt 1 then goto, achtzehn10
    if vektorr17b(lauf) + (durchmesser/2) gt dimccd then goto, achtzehn10
    if vektorr17b(lauf+1) + (durchmesser/2) gt dimccd then goto, achtzehn10
    kreisbereich(vektorr17b(lauf)-(durchmesser/2):(vektorr17b(lauf)$
    +(durchmesser/2)-1),vektorr17b(lauf+1)-(durchmesser/2):$
    (vektorr17b(lauf+1)+(durchmesser/2)-1))=kreis
    print, vektorr17b(lauf), vektorr17b(lauf+1)
    achtzehn10:
    ;tvsc1, congrid(kreisbereich,512,512)
    print, vektorr17b(lauf), vektorr17b(lauf+1)
endfor

tvsc1, congrid(kreisbereich,512,512)

for lauf = 0, anzahl18-1, 2 do begin
    if vektorr18b(lauf) - (durchmesser/2) lt 1 then goto, achtzehn11
    if vektorr18b(lauf+1) - (durchmesser/2) lt 1 then goto, achtzehn11
    if vektorr18b(lauf) + (durchmesser/2) gt dimccd then goto, achtzehn11
    if vektorr18b(lauf+1) + (durchmesser/2) gt dimccd then goto, achtzehn11
    kreisbereich(vektorr18b(lauf)-(durchmesser/2):(vektorr18b(lauf)$
    +(durchmesser/2)-1),vektorr18b(lauf+1)-(durchmesser/2):$
    (vektorr18b(lauf+1)+(durchmesser/2)-1))=kreis
    print, vektorr18b(lauf), vektorr18b(lauf+1)
    achtzehn11:
    ;tvsc1, congrid(kreisbereich,512,512)
    print, vektorr18b(lauf), vektorr18b(lauf+1)
endfor

tvsc1, congrid(kreisbereich,512,512)

for lauf = 0, anzahlm2-1, 2 do begin
    if vektorrm2b(lauf) - (durchmesser/2) lt 1 then goto, neunzehn
    if vektorrm2b(lauf+1) - (durchmesser/2) lt 1 then goto, neunzehn
    if vektorrm2b(lauf) + (durchmesser/2) gt dimccd then goto, neunzehn
    if vektorrm2b(lauf+1) + (durchmesser/2) gt dimccd then goto, neunzehn
    kreisbereich(vektorrm2b(lauf)-(durchmesser/2):(vektorrm2b(lauf)$
    +(durchmesser/2)-1),vektorrm2b(lauf+1)-(durchmesser/2):$
    (vektorrm2b(lauf+1)+(durchmesser/2)-1))=kreis
    print, vektorrm2b(lauf), vektorrm2b(lauf+1)
    neunzehn:
    ;tvsc1, congrid(kreisbereich,512,512)
    print, vektorrm2b(lauf), vektorrm2b(lauf+1)
endfor

tvsc1, congrid(kreisbereich,512,512)

for lauf = 0, anzahlm3-1, 2 do begin
    if vektorrm3b(lauf) - (durchmesser/2) lt 1 then goto, zwanzig
    if vektorrm3b(lauf+1) - (durchmesser/2) lt 1 then goto, zwanzig
    if vektorrm3b(lauf) + (durchmesser/2) gt dimccd then goto, zwanzig
    if vektorrm3b(lauf+1) + (durchmesser/2) gt dimccd then goto, zwanzig
    kreisbereich(vektorrm3b(lauf)-(durchmesser/2):(vektorrm3b(lauf)$
    +(durchmesser/2)-1),vektorrm3b(lauf+1)-(durchmesser/2):$
    (vektorrm3b(lauf+1)+(durchmesser/2)-1))=kreis
```

```

    print, vektorrm3b(lauf), vektorrm3b(lauf+1)
    zwanzig:
    ;tvsc1, congrid(kreisbereich,512,512)
    print, vektorrm3b(lauf), vektorrm3b(lauf+1)
endfor

tvsc1, congrid(kreisbereich,512,512)

for lauf = 0, anzahlm4-1, 2 do begin
    if vektorrm4b(lauf) - (durchmesser/2) lt 1 then goto, einzwanzig
    if vektorrm4b(lauf+1) - (durchmesser/2) lt 1 then goto, einzwanzig
    if vektorrm4b(lauf) + (durchmesser/2) gt dimccd then goto, einzwanzig
    if vektorrm4b(lauf+1) + (durchmesser/2) gt dimccd then goto, einzwanzig
    kreisbereich(vektorrm4b(lauf)-(durchmesser/2):(vektorrm4b(lauf)$
    +(durchmesser/2)-1),vektorrm4b(lauf+1)-(durchmesser/2):$
    (vektorrm4b(lauf+1)+(durchmesser/2)-1))=kreis
    print, vektorrm4b(lauf), vektorrm4b(lauf+1)
    einzwanzig:
    ;tvsc1, congrid(kreisbereich,512,512)
    print, vektorrm4b(lauf), vektorrm4b(lauf+1)
endfor

tvsc1, congrid(kreisbereich,512,512)

for lauf = 0, anzahlm5-1, 2 do begin
    if vektorrm5b(lauf) - (durchmesser/2) lt 1 then goto, zweizwanzig
    if vektorrm5b(lauf+1) - (durchmesser/2) lt 1 then goto, zweizwanzig
    if vektorrm5b(lauf) + (durchmesser/2) gt dimccd then goto, zweizwanzig
    if vektorrm5b(lauf+1) + (durchmesser/2) gt dimccd then goto, zweizwanzig
    kreisbereich(vektorrm5b(lauf)-(durchmesser/2):(vektorrm5b(lauf)$
    +(durchmesser/2)-1),vektorrm5b(lauf+1)-(durchmesser/2):$
    (vektorrm5b(lauf+1)+(durchmesser/2)-1))=kreis
    print, vektorrm5b(lauf), vektorrm5b(lauf+1)
    zweizwanzig:
    ;tvsc1, congrid(kreisbereich,512,512)
    print, vektorrm5b(lauf), vektorrm5b(lauf+1)
endfor

tvsc1, congrid(kreisbereich,512,512)

for lauf = 0, anzahlm6-1, 2 do begin
    if vektorrm6b(lauf) - (durchmesser/2) lt 1 then goto, zweizwanzig1
    if vektorrm6b(lauf+1) - (durchmesser/2) lt 1 then goto, zweizwanzig1
    if vektorrm6b(lauf) + (durchmesser/2) gt dimccd then goto, zweizwanzig1
    if vektorrm6b(lauf+1) + (durchmesser/2) gt dimccd then goto, zweizwanzig1
    kreisbereich(vektorrm6b(lauf)-(durchmesser/2):(vektorrm6b(lauf)$
    +(durchmesser/2)-1),vektorrm6b(lauf+1)-(durchmesser/2):$
    (vektorrm6b(lauf+1)+(durchmesser/2)-1))=kreis
    print, vektorrm6b(lauf), vektorrm6b(lauf+1)
    zweizwanzig1:
    ;tvsc1, congrid(kreisbereich,512,512)
    print, vektorrm6b(lauf), vektorrm6b(lauf+1)
endfor

tvsc1, congrid(kreisbereich,512,512)

for lauf = 0, anzahlm7-1, 2 do begin
    if vektorrm7b(lauf) - (durchmesser/2) lt 1 then goto, zweizwanzig2
    if vektorrm7b(lauf+1) - (durchmesser/2) lt 1 then goto, zweizwanzig2
    if vektorrm7b(lauf) + (durchmesser/2) gt dimccd then goto, zweizwanzig2
    if vektorrm7b(lauf+1) + (durchmesser/2) gt dimccd then goto, zweizwanzig2
    kreisbereich(vektorrm7b(lauf)-(durchmesser/2):(vektorrm7b(lauf)$

```

```
+(durchmesser/2)-1),vektorm7b(lauf+1)-(durchmesser/2):$
(vektorm7b(lauf+1)+(durchmesser/2)-1)=kreis
print, vektorm7b(lauf), vektorm7b(lauf+1)
zweizwanzig2:
;tvsl, congrid(kreisbereich,512,512)
print, vektorm7b(lauf), vektorm7b(lauf+1)
endfor

tvsl, congrid(kreisbereich,512,512)

for lauf = 0, anzahlm8-1, 2 do begin
  if vektorm8b(lauf) - (durchmesser/2) lt 1 then goto, zweizwanzig3
  if vektorm8b(lauf+1) - (durchmesser/2) lt 1 then goto, zweizwanzig3
  if vektorm8b(lauf) + (durchmesser/2) gt dimccd then goto, zweizwanzig3
  if vektorm8b(lauf+1) + (durchmesser/2) gt dimccd then goto, zweizwanzig3
  kreisbereich(vektorm8b(lauf)-(durchmesser/2):(vektorm8b(lauf)$
+(durchmesser/2)-1),vektorm8b(lauf+1)-(durchmesser/2):$
(vektorm8b(lauf+1)+(durchmesser/2)-1))=kreis
  print, vektorm8b(lauf), vektorm8b(lauf+1)
  zweizwanzig3:
  ;tvsl, congrid(kreisbereich,512,512)
  print, vektorm8b(lauf), vektorm8b(lauf+1)
endfor

tvsl, congrid(kreisbereich,512,512)

for lauf = 0, anzahlm9-1, 2 do begin
  if vektorm9b(lauf) - (durchmesser/2) lt 1 then goto, zweizwanzig4
  if vektorm9b(lauf+1) - (durchmesser/2) lt 1 then goto, zweizwanzig4
  if vektorm9b(lauf) + (durchmesser/2) gt dimccd then goto, zweizwanzig4
  if vektorm9b(lauf+1) + (durchmesser/2) gt dimccd then goto, zweizwanzig4
  kreisbereich(vektorm9b(lauf)-(durchmesser/2):(vektorm9b(lauf)$
+(durchmesser/2)-1),vektorm9b(lauf+1)-(durchmesser/2):$
(vektorm9b(lauf+1)+(durchmesser/2)-1))=kreis
  print, vektorm9b(lauf), vektorm9b(lauf+1)
  zweizwanzig4:
  ;tvsl, congrid(kreisbereich,512,512)
  print, vektorm9b(lauf), vektorm9b(lauf+1)
endfor

tvsl, congrid(kreisbereich,512,512)

for lauf = 0, anzahlm10-1, 2 do begin
  if vektorm10b(lauf) - (durchmesser/2) lt 1 then goto, zweizwanzig5
  if vektorm10b(lauf+1) - (durchmesser/2) lt 1 then goto, zweizwanzig5
  if vektorm10b(lauf) + (durchmesser/2) gt dimccd then goto, zweizwanzig5
  if vektorm10b(lauf+1) + (durchmesser/2) gt dimccd then goto, zweizwanzig5
  kreisbereich(vektorm10b(lauf)-(durchmesser/2):(vektorm10b(lauf)$
+(durchmesser/2)-1),vektorm10b(lauf+1)-(durchmesser/2):$
(vektorm10b(lauf+1)+(durchmesser/2)-1))=kreis
  print, vektorm10b(lauf), vektorm10b(lauf+1)
  zweizwanzig5:
  ;tvsl, congrid(kreisbereich,512,512)
  print, vektorm10b(lauf), vektorm10b(lauf+1)
endfor

tvsl, congrid(kreisbereich,512,512)

for lauf = 0, anzahlm11-1, 2 do begin
  if vektorm11b(lauf) - (durchmesser/2) lt 1 then goto, zweizwanzig6
  if vektorm11b(lauf+1) - (durchmesser/2) lt 1 then goto, zweizwanzig6
  if vektorm11b(lauf) + (durchmesser/2) gt dimccd then goto, zweizwanzig6
```

```

    if vektorrm11b(lauf+1) + (durchmesser/2) gt dimccd then goto, zweizwanzig6
    kreisbereich(vektorrm11b(lauf)-(durchmesser/2):(vektorrm11b(lauf)$
    +(durchmesser/2)-1),vektorrm11b(lauf+1)-(durchmesser/2):$
    (vektorrm11b(lauf+1)+(durchmesser/2)-1))=kreis
    print, vektorrm11b(lauf), vektorrm11b(lauf+1)
    zweizwanzig6:
    ;tvsc1, congrid(kreisbereich,512,512)
    print, vektorrm11b(lauf), vektorrm11b(lauf+1)
endfor

tvsc1, congrid(kreisbereich,512,512)

for lauf = 0, anzahlm12-1, 2 do begin
    if vektorrm12b(lauf) - (durchmesser/2) lt 1 then goto, zweizwanzig7
    if vektorrm12b(lauf+1) - (durchmesser/2) lt 1 then goto, zweizwanzig7
    if vektorrm12b(lauf) + (durchmesser/2) gt dimccd then goto, zweizwanzig7
    if vektorrm12b(lauf+1) + (durchmesser/2) gt dimccd then goto, zweizwanzig7
    kreisbereich(vektorrm12b(lauf)-(durchmesser/2):(vektorrm12b(lauf)$
    +(durchmesser/2)-1),vektorrm12b(lauf+1)-(durchmesser/2):$
    (vektorrm12b(lauf+1)+(durchmesser/2)-1))=kreis
    print, vektorrm12b(lauf), vektorrm12b(lauf+1)
    zweizwanzig7:
    ;tvsc1, congrid(kreisbereich,512,512)
    print, vektorrm12b(lauf), vektorrm12b(lauf+1)
endfor

tvsc1, congrid(kreisbereich,512,512)

for lauf = 0, anzahlm13-1, 2 do begin
    if vektorrm13b(lauf) - (durchmesser/2) lt 1 then goto, zweizwanzig8
    if vektorrm13b(lauf+1) - (durchmesser/2) lt 1 then goto, zweizwanzig8
    if vektorrm13b(lauf) + (durchmesser/2) gt dimccd then goto, zweizwanzig8
    if vektorrm13b(lauf+1) + (durchmesser/2) gt dimccd then goto, zweizwanzig8
    kreisbereich(vektorrm13b(lauf)-(durchmesser/2):(vektorrm13b(lauf)$
    +(durchmesser/2)-1),vektorrm13b(lauf+1)-(durchmesser/2):$
    (vektorrm13b(lauf+1)+(durchmesser/2)-1))=kreis
    print, vektorrm13b(lauf), vektorrm13b(lauf+1)
    zweizwanzig8:
    ;tvsc1, congrid(kreisbereich,512,512)
    print, vektorrm13b(lauf), vektorrm13b(lauf+1)
endfor

tvsc1, congrid(kreisbereich,512,512)

for lauf = 0, anzahlm14-1, 2 do begin
    if vektorrm14b(lauf) - (durchmesser/2) lt 1 then goto, zweizwanzig9
    if vektorrm14b(lauf+1) - (durchmesser/2) lt 1 then goto, zweizwanzig9
    if vektorrm14b(lauf) + (durchmesser/2) gt dimccd then goto, zweizwanzig9
    if vektorrm14b(lauf+1) + (durchmesser/2) gt dimccd then goto, zweizwanzig9
    kreisbereich(vektorrm14b(lauf)-(durchmesser/2):(vektorrm14b(lauf)$
    +(durchmesser/2)-1),vektorrm14b(lauf+1)-(durchmesser/2):$
    (vektorrm14b(lauf+1)+(durchmesser/2)-1))=kreis
    print, vektorrm14b(lauf), vektorrm14b(lauf+1)
    zweizwanzig9:
    ;tvsc1, congrid(kreisbereich,512,512)
    print, vektorrm14b(lauf), vektorrm14b(lauf+1)
endfor

tvsc1, congrid(kreisbereich,512,512)

for lauf = 0, anzahlm15-1, 2 do begin
    if vektorrm15b(lauf) - (durchmesser/2) lt 1 then goto, zweizwanzig10

```

```
    if vektorrm15b(lauf+1) - (durchmesser/2) lt 1 then goto, zweizwanzig10
    if vektorrm15b(lauf) + (durchmesser/2) gt dimccd then goto, zweizwanzig10
    if vektorrm15b(lauf+1) + (durchmesser/2) gt dimccd then goto, zweizwanzig10
    kreisbereich(vektorrm15b(lauf)-(durchmesser/2):(vektorrm15b(lauf)$
    +(durchmesser/2)-1),vektorrm15b(lauf+1)-(durchmesser/2):$
    (vektorrm15b(lauf+1)+(durchmesser/2)-1))=kreis
    print, vektorrm15b(lauf), vektorrm15b(lauf+1)
    zweizwanzig10:
    ;tvsc1, congrid(kreisbereich,512,512)
    print, vektorrm15b(lauf), vektorrm15b(lauf+1)
endfor

tvsc1, congrid(kreisbereich,512,512)

for lauf = 0, anzahlm16-1, 2 do begin
    if vektorrm16b(lauf) - (durchmesser/2) lt 1 then goto, zweizwanzig11
    if vektorrm16b(lauf+1) - (durchmesser/2) lt 1 then goto, zweizwanzig11
    if vektorrm16b(lauf) + (durchmesser/2) gt dimccd then goto, zweizwanzig11
    if vektorrm16b(lauf+1) + (durchmesser/2) gt dimccd then goto, zweizwanzig11
    kreisbereich(vektorrm16b(lauf)-(durchmesser/2):(vektorrm16b(lauf)$
    +(durchmesser/2)-1),vektorrm16b(lauf+1)-(durchmesser/2):$
    (vektorrm16b(lauf+1)+(durchmesser/2)-1))=kreis
    print, vektorrm16b(lauf), vektorrm16b(lauf+1)
    zweizwanzig11:
    ;tvsc1, congrid(kreisbereich,512,512)
    print, vektorrm16b(lauf), vektorrm16b(lauf+1)
endfor

tvsc1, congrid(kreisbereich,512,512)

for lauf = 0, anzahlm17-1, 2 do begin
    if vektorrm17b(lauf) - (durchmesser/2) lt 1 then goto, zweizwanzig12
    if vektorrm17b(lauf+1) - (durchmesser/2) lt 1 then goto, zweizwanzig12
    if vektorrm17b(lauf) + (durchmesser/2) gt dimccd then goto, zweizwanzig12
    if vektorrm17b(lauf+1) + (durchmesser/2) gt dimccd then goto, zweizwanzig12
    kreisbereich(vektorrm17b(lauf)-(durchmesser/2):(vektorrm17b(lauf)$
    +(durchmesser/2)-1),vektorrm17b(lauf+1)-(durchmesser/2):$
    (vektorrm17b(lauf+1)+(durchmesser/2)-1))=kreis
    print, vektorrm17b(lauf), vektorrm17b(lauf+1)
    zweizwanzig12:
    ;tvsc1, congrid(kreisbereich,512,512)
    print, vektorrm17b(lauf), vektorrm17b(lauf+1)
endfor

tvsc1, congrid(kreisbereich,512,512)

for lauf = 0, anzahlm18-1, 2 do begin
    if vektorrm18b(lauf) - (durchmesser/2) lt 1 then goto, zweizwanzig13
    if vektorrm18b(lauf+1) - (durchmesser/2) lt 1 then goto, zweizwanzig13
    if vektorrm18b(lauf) + (durchmesser/2) gt dimccd then goto, zweizwanzig13
    if vektorrm18b(lauf+1) + (durchmesser/2) gt dimccd then goto, zweizwanzig13
    kreisbereich(vektorrm18b(lauf)-(durchmesser/2):(vektorrm18b(lauf)$
    +(durchmesser/2)-1),vektorrm18b(lauf+1)-(durchmesser/2):$
    (vektorrm18b(lauf+1)+(durchmesser/2)-1))=kreis
    print, vektorrm18b(lauf), vektorrm18b(lauf+1)
    zweizwanzig13:
    ;tvsc1, congrid(kreisbereich,512,512)
    print, vektorrm18b(lauf), vektorrm18b(lauf+1)
endfor

tvsc1, congrid(kreisbereich,512,512)
```

```

bildkonv = bild * kreisbereich ;ACHTUNG!!
tvsl, congrid(bildkonv,512,512)

; Untergrundkorrektur

if ringejanein eq 1 then begin
print, 'Nun werden die Untergrundringe erzeugt!'

;Zuruecksetzen des Verschiebungsvektors fuer die Untergrundkorrektur

vektorr1b = vektorr1b - verschieb
vektorr2b = vektorr2b - verschieb
vektorr3b = vektorr3b - verschieb
vektorr4b = vektorr4b - verschieb
vektorr5b = vektorr5b - verschieb
vektorr6b = vektorr6b - verschieb
vektorr7b = vektorr7b - verschieb
vektorr8b = vektorr8b - verschieb
vektorr9b = vektorr9b - verschieb
vektorr10b = vektorr10b - verschieb
vektorr11b = vektorr11b - verschieb
vektorr12b = vektorr12b - verschieb
vektorr13b = vektorr13b - verschieb
vektorr14b = vektorr14b - verschieb
vektorr15b = vektorr15b - verschieb
vektorr16b = vektorr16b - verschieb
vektorr17b = vektorr17b - verschieb
vektorr18b = vektorr18b - verschieb

vektorm2b = vektorm2b - verschieb
vektorm3b = vektorm3b - verschieb
vektorm4b = vektorm4b - verschieb
vektorm5b = vektorm5b - verschieb
vektorm6b = vektorm6b - verschieb
vektorm7b = vektorm7b - verschieb
vektorm8b = vektorm8b - verschieb
vektorm9b = vektorm9b - verschieb
vektorm10b = vektorm10b - verschieb
vektorm11b = vektorm11b - verschieb
vektorm12b = vektorm12b - verschieb
vektorm13b = vektorm13b - verschieb
vektorm14b = vektorm14b - verschieb
vektorm15b = vektorm15b - verschieb
vektorm16b = vektorm16b - verschieb
vektorm17b = vektorm17b - verschieb
vektorm18b = vektorm18b - verschieb

k = lindgen(durchmesserunter1,durchmesserunter1)
kx = (k mod durchmesserunter1) - durchmesserunter1/2
ky = (k/durchmesserunter1) - durchmesserunter1/2
kreis1 = (kx^2 + ky^2) le (durchmesserunter1/2)^2

kzwei = lindgen(durchmesserunter2,durchmesserunter2)
kxzwei = (kzwei mod durchmesserunter2) - durchmesserunter2/2
kyzwei = (kzwei/durchmesserunter2) - durchmesserunter2/2
kreiszwei = (kxzwei^2 + kyzwei^2) le (durchmesserunter2/2)^2

if anzahl1 eq 0 then anzahl1 = 2
vektorr1bunter = fltarr(anzahl1/2,1)
vektorr1bfehlerunter = fltarr(anzahl1/2,1)
summeunterintvektorr1b = fltarr(anzahl1/2,1)

```

kreisringvar = fltarr(dimccd,dimccd) ; Nur um den Fehler der 1. Darstellung zu verhindern
 laufeins = 0

for lauf = 0, anzahl1-1, 2 do begin

print, lauf, laufeins

if vektorr1b(lauf) - (durchmesserunter1/2) lt 1 then goto, dreizwanzig
 if vektorr1b(lauf+1) - (durchmesserunter1/2) lt 1 then goto, dreizwanzig
 if vektorr1b(lauf) + (durchmesserunter1/2) gt dimccd then goto, dreizwanzig
 if vektorr1b(lauf+1) + (durchmesserunter1/2) gt dimccd then goto, dreizwanzig

kreisbereichring1 = fltarr(dimccd,dimccd)
 kreisbereichring1((vektorr1b(lauf)-(durchmesserunter1/2)):\$
 (vektorr1b(lauf)+(durchmesserunter1/2)-1),(vektorr1b(lauf+1)-\$
 (durchmesserunter1/2)):(vektorr1b(lauf+1)+(durchmesserunter1/2)-1))\$
 =kreis1

kreisbereichring2 = fltarr(dimccd,dimccd)
 kreisbereichring2((vektorr1b(lauf)-(durchmesserunter2/2)):\$
 (vektorr1b(lauf)+(durchmesserunter2/2)-1),(vektorr1b(lauf+1)-\$
 (durchmesserunter2/2)):(vektorr1b(lauf+1)+(durchmesserunter2/2)-1))\$
 =kreiszwei

;Erzeugung des Rings und Bestimmung der mittleren Intensitaet

kreisringvar = kreisbereichring1 - kreisbereichring2
 ;ringbildvar = kreisringvar * bild
 ;ortuntervar = kreisringvar gt 0; kann ich mir schenken!!
 positionenuntervar = where(kreisringvar)
 nuntervar = (size(positionenuntervar))(1)
 yuntervar = positionenuntervar / dimccd
 xuntervar = positionenuntervar mod dimccd

unterintvektorr1bvar = bild(xuntervar,yuntervar)
 summeunterintvektorr1b(laufeins)= total(unterintvektorr1bvar)
 vektorr1bunter(laufeins)= summeunterintvektorr1b(laufeins)/nuntervar
 vektorr1bfehlerunter(laufeins)= meanabsdev(unterintvektorr1bvar)

if lauf eq 0 then ringe = kreisringvar else ringe = ringe +\$
 kreisringvar
 dreizwanzig:
 laufeins = laufeins+1
 if lauf eq 0 then ringe = kreisringvar
 ;if lauf eq 0 then laufeins = 1
 ;tvsl, congrid(ringe,512,512)

endfor

tvsl, congrid(ringe,512,512)

if anzahl2 eq 0 then anzahl2 = 2
 vektorr2bunter = fltarr(anzahl2/2,1)
 vektorr2bfehlerunter = fltarr(anzahl2/2,1)
 summeunterintvektorr2b = fltarr(anzahl2/2,1)

laufeins = 0

for lauf = 0, anzahl2-1, 2 do begin

if vektorr2b(lauf) - (durchmesserunter1/2) lt 1 then goto, vierzwanzig
 if vektorr2b(lauf+1) - (durchmesserunter1/2) lt 1 then goto, vierzwanzig
 if vektorr2b(lauf) + (durchmesserunter1/2) gt dimccd then goto, vierzwanzig
 if vektorr2b(lauf+1) + (durchmesserunter1/2) gt dimccd then goto, vierzwanzig
 kreisbereichring1 = fltarr(dimccd,dimccd)

```

kreisbereichring1((vektorr2b(lauf)-(durchmesserunter1/2)):$
(vektorr2b(lauf)+(durchmesserunter1/2)-1),(vektorr2b(lauf+1)-$
(durchmesserunter1/2)):(vektorr2b(lauf+1)+(durchmesserunter1/2)-1))$
=kreis1

kreisbereichring2      = fltarr(dimccd,dimccd)
kreisbereichring2((vektorr2b(lauf)-(durchmesserunter2/2)):$
(vektorr2b(lauf)+(durchmesserunter2/2)-1),(vektorr2b(lauf+1)-$
(durchmesserunter2/2)):(vektorr2b(lauf+1)+(durchmesserunter2/2)-1))$
=kreiszwei

;Erzeugung des Rings und Bestimmung der mittleren Intensitaet

kreisringvar          = kreisbereichring1 - kreisbereichring2
;ringbildvar          = kreisringvar * bild
;ortuntervar         = kreisringvar gt 0
positionenuntervar    = where(kreisringvar)
nuntervar             = (size(positionenuntervar))(1)
yuntervar            = positionenuntervar / dimccd
xuntervar            = positionenuntervar mod dimccd

unterintvektorr2bvar  = bild(xuntervar,yuntervar)
summeunterintvektorr2b(laufeins)= total(unterintvektorr2bvar)
vektorr2bunter(laufeins)= summeunterintvektorr2b(laufeins)/nuntervar
vektorr2bfehlerunter(laufeins)= meanabsdev(unterintvektorr2bvar)
ringe = ringe + kreisringvar
vierzwanzig:
laufeins = laufeins+1
;tvsc1, congrid(ringe,512,512)
print, lauf
endfor

tvsc1, congrid(ringe,512,512)

if anzahl3 eq 0 then anzahl3 = 2
vektorr3bunter        = fltarr(anzahl3/2,1)
vektorr3bfehlerunter  = fltarr(anzahl3/2,1)
summeunterintvektorr3b      = fltarr(anzahl3/2,1)

laufeins = 0

for lauf = 0, anzahl3-1, 2 do begin
  if vektorr3b(lauf) - (durchmesserunter1/2) lt 1 then goto, fuenfzwanzig
  if vektorr3b(lauf+1) - (durchmesserunter1/2) lt 1 then goto, fuenfzwanzig
  if vektorr3b(lauf) + (durchmesserunter1/2) gt dimccd then goto, fuenfzwanzig
  if vektorr3b(lauf+1) + (durchmesserunter1/2) gt dimccd then goto, fuenfzwanzig
  kreisbereichring1      = fltarr(dimccd,dimccd)
  kreisbereichring1((vektorr3b(lauf)-(durchmesserunter1/2)):$
(vektorr3b(lauf)+(durchmesserunter1/2)-1),(vektorr3b(lauf+1)-$
(durchmesserunter1/2)):(vektorr3b(lauf+1)+(durchmesserunter1/2)-1))$
=kreis1

  kreisbereichring2      = fltarr(dimccd,dimccd)
  kreisbereichring2((vektorr3b(lauf)-(durchmesserunter2/2)):$
(vektorr3b(lauf)+(durchmesserunter2/2)-1),(vektorr3b(lauf+1)-$
(durchmesserunter2/2)):(vektorr3b(lauf+1)+(durchmesserunter2/2)-1))$
=kreiszwei

;Erzeugung des Rings und Bestimmung der mittleren Intensitaet

kreisringvar          = kreisbereichring1 - kreisbereichring2
positionenuntervar    = where(kreisringvar)

```

```

nuntervar          = (size(positionenuntervar))(1)
yuntervar          = positionenuntervar / dimccd
xuntervar          = positionenuntervar mod dimccd

unterintvektorr3bvar = bild(xuntervar,yuntervar)
summeunterintvektorr3b(laufeins)= total(unterintvektorr3bvar)
vektorr3bunter(laufeins)= summeunterintvektorr3b(laufeins)/nuntervar
vektorr3bfehlerunter(laufeins)= meanabsdev(unterintvektorr3bvar)
ringe = ringe + kreisringvar
fuenfzwanzig:
laufeins = laufeins+1
;tvsc1, congrid(ringe,512,512)
print, lauf
endfor

tvsc1, congrid(ringe,512,512)

if anzahl4 eq 0 then anzahl4 = 2
vektorr4bunter      = fltarr(anzahl4/2,1)
vektorr4bfehlerunter = fltarr(anzahl4/2,1)
summeunterintvektorr4b = fltarr(anzahl4/2,1)

laufeins = 0

for lauf = 0, anzahl4-1, 2 do begin
  if vektorr4b(lauf) - (durchmesserunter1/2) lt 1 then goto, sechszwanzig
  if vektorr4b(lauf+1) - (durchmesserunter1/2) lt 1 then goto, sechszwanzig
  if vektorr4b(lauf) + (durchmesserunter1/2) gt dimccd then goto, sechszwanzig
  if vektorr4b(lauf+1) + (durchmesserunter1/2) gt dimccd then goto, sechszwanzig
  kreisbereichring1 = fltarr(dimccd,dimccd)
  kreisbereichring1((vektorr4b(lauf)-(durchmesserunter1/2)):$
  (vektorr4b(lauf)+(durchmesserunter1/2)-1),(vektorr4b(lauf+1)-$
  (durchmesserunter1/2)):(vektorr4b(lauf+1)+(durchmesserunter1/2)-1))$
  =kreis1

  kreisbereichring2 = fltarr(dimccd,dimccd)
  kreisbereichring2((vektorr4b(lauf)-(durchmesserunter2/2)):$
  (vektorr4b(lauf)+(durchmesserunter2/2)-1),(vektorr4b(lauf+1)-$
  (durchmesserunter2/2)):(vektorr4b(lauf+1)+(durchmesserunter2/2)-1))$
  =kreiszwei

  ;Erzeugung des Rings und Bestimmung der mittleren Intensitaet

  kreisringvar      = kreisbereichring1 - kreisbereichring2
  positionenuntervar = where(kreisringvar)
  nuntervar          = (size(positionenuntervar))(1)
  yuntervar          = positionenuntervar / dimccd
  xuntervar          = positionenuntervar mod dimccd

  unterintvektorr4bvar = bild(xuntervar,yuntervar)
  summeunterintvektorr4b(laufeins)= total(unterintvektorr4bvar)
  vektorr4bunter(laufeins)= summeunterintvektorr4b(laufeins)/nuntervar
  vektorr4bfehlerunter(laufeins)= meanabsdev(unterintvektorr4bvar)
  ringe = ringe + kreisringvar
  sechszwanzig:
  laufeins = laufeins+1
  ;tvsc1, congrid(ringe,512,512)
  print, lauf
endfor

tvsc1, congrid(ringe,512,512)

```

```

if anzahl5 eq 0 then anzahl5 = 2
vektorr5bunter      = fltarr(anzahl5/2,1)
vektorr5bfehlerunter = fltarr(anzahl5/2,1)
summeunterintvektorr5b      = fltarr(anzahl5/2,1)

laufeins = 0

for lauf = 0, anzahl5-1, 2 do begin
  if vektorr5b(lauf) - (durchmesserunter1/2) lt 1 then goto, siebzwanzig
  if vektorr5b(lauf+1) - (durchmesserunter1/2) lt 1 then goto, siebzwanzig
  if vektorr5b(lauf) + (durchmesserunter1/2) gt dimccd then goto, siebzwanzig
  if vektorr5b(lauf+1) + (durchmesserunter1/2) gt dimccd then goto, siebzwanzig
  kreisbereichring1      = fltarr(dimccd,dimccd)
  kreisbereichring1((vektorr5b(lauf)-(durchmesserunter1/2)):$
(vektorr5b(lauf)+(durchmesserunter1/2)-1),(vektorr5b(lauf+1)-$
(durchmesserunter1/2)):(vektorr5b(lauf+1)+(durchmesserunter1/2)-1))$
=kreis1

  kreisbereichring2      = fltarr(dimccd,dimccd)
  kreisbereichring2((vektorr5b(lauf)-(durchmesserunter2/2)):$
(vektorr5b(lauf)+(durchmesserunter2/2)-1),(vektorr5b(lauf+1)-$
(durchmesserunter2/2)):(vektorr5b(lauf+1)+(durchmesserunter2/2)-1))$
=kreiszwei

;Erzeugung des Rings und Bestimmung der mittleren Intensitaet

  kreisringvar          = kreisbereichring1 - kreisbereichring2
  ;ringbildvar          = kreisringvar * bild
  ;ortuntervar          = kreisringvar gt 0
  positionenuntervar    = where(kreisringvar)
  nuntervar             = (size(positionenuntervar))(1)
  yuntervar             = positionenuntervar / dimccd
  xuntervar             = positionenuntervar mod dimccd

  unterintvektorr5bvar  = bild(xuntervar,yuntervar)
  summeunterintvektorr5b(laufeins)= total(unterintvektorr5bvar)
  vektorr5bunter(laufeins)= summeunterintvektorr5b(laufeins)/nuntervar
  vektorr5bfehlerunter(laufeins)= meanabsdev(unterintvektorr5bvar)
  ringe = ringe + kreisringvar
  siebzwanzig:
  laufeins = laufeins+1
  ;tvsc, congrid(ringe,512,512)
  print, lauf
endfor

tvsc, congrid(ringe,512,512)

if anzahl6 eq 0 then anzahl6 = 2
vektorr6bunter      = fltarr(anzahl6/2,1)
vektorr6bfehlerunter = fltarr(anzahl6/2,1)
summeunterintvektorr6b      = fltarr(anzahl6/2,1)

laufeins = 0

for lauf = 0, anzahl6-1, 2 do begin
  if vektorr6b(lauf) - (durchmesserunter1/2) lt 1 then goto, achtzwanzig
  if vektorr6b(lauf+1) - (durchmesserunter1/2) lt 1 then goto, achtzwanzig
  if vektorr6b(lauf) + (durchmesserunter1/2) gt dimccd then goto, achtzwanzig
  if vektorr6b(lauf+1) + (durchmesserunter1/2) gt dimccd then goto, achtzwanzig

  kreisbereichring1      = fltarr(dimccd,dimccd)
  kreisbereichring1((vektorr6b(lauf)-(durchmesserunter1/2)):$

```

```
(vektorr6b(lauf)+(durchmesserunter1/2)-1),(vektorr6b(lauf+1)-$
(durchmesserunter1/2)):(vektorr6b(lauf+1)+(durchmesserunter1/2)-1))$
=kreis1
```

```
kreisbereichring2 = fltarr(dimccd,dimccd)
kreisbereichring2((vektorr6b(lauf)-(durchmesserunter2/2)):$
(vektorr6b(lauf)+(durchmesserunter2/2)-1),(vektorr6b(lauf+1)-$
(durchmesserunter2/2)):(vektorr6b(lauf+1)+(durchmesserunter2/2)-1))$
=kreiszwei
```

;Erzeugung des Rings und Bestimmung der mittleren Intensitaet

```
kreisringvar = kreisbereichring1 - kreisbereichring2
positionenuntervar = where(kreisringvar)
nuntervar = (size(positionenuntervar))(1)
yuntervar = positionenuntervar / dimccd
xuntervar = positionenuntervar mod dimccd
```

```
unterintvektorr6bvar = bild(xuntervar,yuntervar)
summeunterintvektorr6b(laufeins)= total(unterintvektorr6bvar)
vektorr6bunter(laufeins)= summeunterintvektorr6b(laufeins)/nuntervar
vektorr6bfehlerunter(laufeins)= meanabsdev(unterintvektorr6bvar)
ringe = ringe + kreisringvar
achtzwanzig:
laufeins = laufeins+1
;tvsc1, congrid(ringe,512,512)
print, lauf
```

endfor

tvsc1, congrid(ringe,512,512)

```
if anzahl7 eq 0 then anzahl7 = 2
vektorr7bunter = fltarr(anzahl7/2,1)
vektorr7bfehlerunter = fltarr(anzahl7/2,1)
summeunterintvektorr7b = fltarr(anzahl7/2,1)
```

laufeins = 0

for lauf = 0, anzahl7-1, 2 do begin

```
if vektor7b(lauf) - (durchmesserunter1/2) lt 1 then goto, neunzwanzig
if vektor7b(lauf+1) - (durchmesserunter1/2) lt 1 then goto, neunzwanzig
if vektor7b(lauf) + (durchmesserunter1/2) gt dimccd then goto, neunzwanzig
if vektor7b(lauf+1) + (durchmesserunter1/2) gt dimccd then goto, neunzwanzig
kreisbereichring1 = fltarr(dimccd,dimccd)
kreisbereichring1((vektorr7b(lauf)-(durchmesserunter1/2)):$
(vektorr7b(lauf)+(durchmesserunter1/2)-1),(vektorr7b(lauf+1)-$
(durchmesserunter1/2)):(vektorr7b(lauf+1)+(durchmesserunter1/2)-1))$
=kreis1
```

```
kreisbereichring2 = fltarr(dimccd,dimccd)
kreisbereichring2((vektorr7b(lauf)-(durchmesserunter2/2)):$
(vektorr7b(lauf)+(durchmesserunter2/2)-1),(vektorr7b(lauf+1)-$
(durchmesserunter2/2)):(vektorr7b(lauf+1)+(durchmesserunter2/2)-1))$
=kreiszwei
```

;Erzeugung des Rings und Bestimmung der mittleren Intensitaet

```
kreisringvar = kreisbereichring1 - kreisbereichring2
positionenuntervar = where(kreisringvar)
nuntervar = (size(positionenuntervar))(1)
yuntervar = positionenuntervar / dimccd
xuntervar = positionenuntervar mod dimccd
```

```

unterintvektorr7bvar = bild(xuntervar,yuntervar)
summeunterintvektorr7b(laufeins)= total(unterintvektorr7bvar)
vektorr7bunter(laufeins)= summeunterintvektorr7b(laufeins)/nuntervar
vektorr7bfehlerunter(laufeins)= meanabsdev(unterintvektorr7bvar)
ringe = ringe + kreisringvar
neunzwanzig:
laufeins = laufeins+1
;tvsc1, congrid(ringe,512,512)
print, lauf
endfor

tvsc1, congrid(ringe,512,512)

if anzahl8 eq 0 then anzahl8 = 2
vektorr8bunter = fltarr(anzahl8/2,1)
vektorr8bfehlerunter = fltarr(anzahl8/2,1)
summeunterintvektorr8b = fltarr(anzahl8/2,1)

laufeins = 0

for lauf = 0, anzahl8-1, 2 do begin
  if vektorr8b(lauf) - (durchmesserunter1/2) lt 1 then goto, neunzwanzig1
  if vektorr8b(lauf+1) - (durchmesserunter1/2) lt 1 then goto, neunzwanzig1
  if vektorr8b(lauf) + (durchmesserunter1/2) gt dimccd then goto, neunzwanzig1
  if vektorr8b(lauf+1) + (durchmesserunter1/2) gt dimccd then goto, neunzwanzig1
  kreisbereichring1 = fltarr(dimccd,dimccd)
  kreisbereichring1((vektorr8b(lauf)-(durchmesserunter1/2)):$
  (vektorr8b(lauf)+(durchmesserunter1/2)-1),(vektorr8b(lauf+1)-$
  (durchmesserunter1/2)):(vektorr8b(lauf+1)+(durchmesserunter1/2)-1))$
  =kreis1

  kreisbereichring2 = fltarr(dimccd,dimccd)
  kreisbereichring2((vektorr8b(lauf)-(durchmesserunter2/2)):$
  (vektorr8b(lauf)+(durchmesserunter2/2)-1),(vektorr8b(lauf+1)-$
  (durchmesserunter2/2)):(vektorr8b(lauf+1)+(durchmesserunter2/2)-1))$
  =kreiszwei
  ;Erzeugung des Rings und Bestimmung der mittleren Intensitaet

  kreisringvar = kreisbereichring1 - kreisbereichring2
  positionenuntervar = where(kreisringvar)
  nuntervar = (size(positionenuntervar))(1)
  yuntervar = positionenuntervar / dimccd
  xuntervar = positionenuntervar mod dimccd

  unterintvektorr8bvar = bild(xuntervar,yuntervar)
  summeunterintvektorr8b(laufeins)= total(unterintvektorr8bvar)
  vektorr8bunter(laufeins)= summeunterintvektorr8b(laufeins)/nuntervar
  vektorr8bfehlerunter(laufeins)= meanabsdev(unterintvektorr8bvar)
  ringe = ringe + kreisringvar
  neunzwanzig1:
  laufeins = laufeins+1
  ;tvsc1, congrid(ringe,512,512)
  print, lauf
endfor

tvsc1, congrid(ringe,512,512)

if anzahl9 eq 0 then anzahl9 = 2
vektorr9bunter = fltarr(anzahl9/2,1)
vektorr9bfehlerunter = fltarr(anzahl9/2,1)
summeunterintvektorr9b = fltarr(anzahl9/2,1)

```

laufeins = 0

for lauf = 0, anzahl9-1, 2 do begin

if vektorr9b(lauf) - (durchmesserunter1/2) lt 1 then goto, neunzwanzig2
 if vektorr9b(lauf+1) - (durchmesserunter1/2) lt 1 then goto, neunzwanzig2
 if vektorr9b(lauf) + (durchmesserunter1/2) gt dimccd then goto, neunzwanzig2
 if vektorr9b(lauf+1) + (durchmesserunter1/2) gt dimccd then goto, neunzwanzig2

kreisbereichring1 = fltarr(dimccd,dimccd)
 kreisbereichring1((vektorr9b(lauf)-(durchmesserunter1/2)):\$
 (vektorr9b(lauf)+(durchmesserunter1/2)-1),(vektorr9b(lauf+1)-\$
 (durchmesserunter1/2)):(vektorr9b(lauf+1)+(durchmesserunter1/2)-1))\$
 =kreis1

kreisbereichring2 = fltarr(dimccd,dimccd)
 kreisbereichring2((vektorr9b(lauf)-(durchmesserunter2/2)):\$
 (vektorr9b(lauf)+(durchmesserunter2/2)-1),(vektorr9b(lauf+1)-\$
 (durchmesserunter2/2)):(vektorr9b(lauf+1)+(durchmesserunter2/2)-1))\$
 =kreiszwei

;Erzeugung des Rings und Bestimmung der mittleren Intensitaet

kreisringvar = kreisbereichring1 - kreisbereichring2
 positionenuntervar = where(kreisringvar)
 nuntervar = (size(positionenuntervar))(1)
 yuntervar = positionenuntervar / dimccd
 xuntervar = positionenuntervar mod dimccd

unterintvektor9bvar = bild(xuntervar,yuntervar)
 summeunterintvektor9b(laufeins)= total(unterintvektor9bvar)
 vektorr9bunter(laufeins)= summeunterintvektor9b(laufeins)/nuntervar
 vektorr9bfehlerunter(laufeins)= meanabsdev(unterintvektor9bvar)
 ringe = ringe + kreisringvar
 neunzwanzig2:
 laufeins = laufeins+1
 ;tvsc1, congrid(ringe,512,512)
 print, lauf

endfor

tvsc1, congrid(ringe,512,512)

if anzahl10 eq 0 then anzahl10 = 2
 vektorr10bunter = fltarr(anzahl10/2,1)
 vektorr10bfehlerunter = fltarr(anzahl10/2,1)
 summeunterintvektor10b = fltarr(anzahl10/2,1)

laufeins = 0

for lauf = 0, anzahl10-1, 2 do begin

if vektorr10b(lauf) - (durchmesserunter1/2) lt 1 then goto, neunzwanzig3
 if vektorr10b(lauf+1) - (durchmesserunter1/2) lt 1 then goto, neunzwanzig3
 if vektorr10b(lauf) + (durchmesserunter1/2) gt dimccd then goto, neunzwanzig3
 if vektorr10b(lauf+1) + (durchmesserunter1/2) gt dimccd then goto, neunzwanzig3

kreisbereichring1 = fltarr(dimccd,dimccd)
 kreisbereichring1((vektorr10b(lauf)-(durchmesserunter1/2)):\$
 (vektorr10b(lauf)+(durchmesserunter1/2)-1),(vektorr10b(lauf+1)-\$
 (durchmesserunter1/2)):(vektorr10b(lauf+1)+(durchmesserunter1/2)-1))\$
 =kreis1

```

kreisbereichring2      = fltarr(dimccd,dimccd)
kreisbereichring2((vektorr10b(lauf)-(durchmesserunter2/2)):$
(vektorr10b(lauf)+(durchmesserunter2/2)-1),(vektorr10b(lauf+1)-$
(durchmesserunter2/2)):(vektorr10b(lauf+1)+(durchmesserunter2/2)-1))$
=kreiszwei

;Erzeugung des Rings und Bestimmung der mittleren Intensitaet

kreisringvar          = kreisbereichring1 - kreisbereichring2
positionenuntervar    = where(kreisringvar)
nuntervar             = (size(positionenuntervar))(1)
yuntervar             = positionenuntervar / dimccd
xuntervar             = positionenuntervar mod dimccd

unterintvektorr10bvar = bild(xuntervar,yuntervar)
summeunterintvektorr10b(laufeins)= total(unterintvektorr10bvar)
vektorr10bunter(laufeins)      = summeunterintvektorr10b(laufeins)/nuntervar
vektorr10bfehlerunter(laufeins)= meanabsdev(unterintvektorr10bvar)
ringe = ringe + kreisringvar
neunzwanzig3:
laufeins = laufeins+1
;tvsc1, congrid(ringe,512,512)
print, lauf
endfor

tvsc1, congrid(ringe,512,512)

if anzahl11 eq 0 then anzahl11 = 2
vektorr11bunter      = fltarr(anzahl11/2,1)
vektorr11bfehlerunter = fltarr(anzahl11/2,1)
summeunterintvektorr11b      = fltarr(anzahl11/2,1)

laufeins = 0

for lauf = 0, anzahl11-1, 2 do begin
if vektorr11b(lauf) - (durchmesserunter1/2) lt 1 then goto, neunzwanzig4
if vektorr11b(lauf+1) - (durchmesserunter1/2) lt 1 then goto, neunzwanzig4
if vektorr11b(lauf) + (durchmesserunter1/2) gt dimccd then goto, neunzwanzig4
if vektorr11b(lauf+1) + (durchmesserunter1/2) gt dimccd then goto, neunzwanzig4

kreisbereichring1      = fltarr(dimccd,dimccd)
kreisbereichring1((vektorr11b(lauf)-(durchmesserunter1/2)):$
(vektorr11b(lauf)+(durchmesserunter1/2)-1),(vektorr11b(lauf+1)-$
(durchmesserunter1/2)):(vektorr11b(lauf+1)+(durchmesserunter1/2)-1))$
=kreis1

kreisbereichring2      = fltarr(dimccd,dimccd)
kreisbereichring2((vektorr11b(lauf)-(durchmesserunter2/2)):$
(vektorr11b(lauf)+(durchmesserunter2/2)-1),(vektorr11b(lauf+1)-$
(durchmesserunter2/2)):(vektorr11b(lauf+1)+(durchmesserunter2/2)-1))$
=kreiszwei

;Erzeugung des Rings und Bestimmung der mittleren Intensitaet

kreisringvar          = kreisbereichring1 - kreisbereichring2
positionenuntervar    = where(kreisringvar)
nuntervar             = (size(positionenuntervar))(1)
yuntervar             = positionenuntervar / dimccd
xuntervar             = positionenuntervar mod dimccd

unterintvektorr11bvar = bild(xuntervar,yuntervar)

```

```

    summeunterintvektorr11b(laufeins)= total(unterintvektorr11bvar)
    vektorr11bunter(laufeins)      = summeunterintvektorr11b(laufeins)/nuntervar
    vektorr11bfehlerunter(laufeins)= meanabsdev(unterintvektorr11bvar)
    ringe = ringe + kreisringvar
    neunzwanzig4:
    laufeins = laufeins+1
    ;tvscf, congrid(ringe,512,512)
    print, lauf
endfor

tvscf, congrid(ringe,512,512)

if anzahl12 eq 0 then anzahl12 = 2
vektorr12bunter      = fltarr(anzahl12/2,1)
vektorr12bfehlerunter = fltarr(anzahl12/2,1)
summeunterintvektorr12b = fltarr(anzahl12/2,1)

laufeins = 0

for lauf = 0, anzahl12-1, 2 do begin
    if vektorr12b(lauf) - (durchmesserunter1/2) lt 1 then goto, neunzwanzig5
    if vektorr12b(lauf+1) - (durchmesserunter1/2) lt 1 then goto, neunzwanzig5
    if vektorr12b(lauf) + (durchmesserunter1/2) gt dimccd then goto, neunzwanzig5
    if vektorr12b(lauf+1) + (durchmesserunter1/2) gt dimccd then goto, neunzwanzig5

    kreisbereichring1      = fltarr(dimccd,dimccd)
    kreisbereichring1((vektorr12b(lauf)-(durchmesserunter1/2)):$
    (vektorr12b(lauf)+(durchmesserunter1/2)-1),(vektorr12b(lauf+1)-$
    (durchmesserunter1/2)):(vektorr12b(lauf+1)+(durchmesserunter1/2)-1))$
    =kreis1

    kreisbereichring2      = fltarr(dimccd,dimccd)
    kreisbereichring2((vektorr12b(lauf)-(durchmesserunter2/2)):$
    (vektorr12b(lauf)+(durchmesserunter2/2)-1),(vektorr12b(lauf+1)-$
    (durchmesserunter2/2)):(vektorr12b(lauf+1)+(durchmesserunter2/2)-1))$
    =kreiszwei

    ;Erzeugung des Rings und Bestimmung der mittleren Intensitaet

    kreisringvar          = kreisbereichring1 - kreisbereichring2
    positionenuntervar    = where(kreisringvar)
    nuntervar             = (size(positionenuntervar))(1)
    yuntervar             = positionenuntervar / dimccd
    xuntervar             = positionenuntervar mod dimccd

    unterintvektorr12bvar = bild(xuntervar,yuntervar)
    summeunterintvektorr12b(laufeins)= total(unterintvektorr12bvar)
    vektorr12bunter(laufeins)      = summeunterintvektorr12b(laufeins)/nuntervar
    vektorr12bfehlerunter(laufeins)= meanabsdev(unterintvektorr12bvar)
    ringe = ringe + kreisringvar
    neunzwanzig5:
    laufeins = laufeins+1
    ;tvscf, congrid(ringe,512,512)
    print, lauf
endfor

tvscf, congrid(ringe,512,512)

if anzahl13 eq 0 then anzahl13 = 2
vektorr13bunter      = fltarr(anzahl13/2,1)
vektorr13bfehlerunter = fltarr(anzahl13/2,1)

```

```

summeunterintvektorr13b      = fltarr(anzahl13/2,1)

laufeins = 0

for lauf = 0, anzahl13-1, 2 do begin
  if vektorr13b(lauf) - (durchmesserunter1/2) lt 1 then goto, neunzwanzig6
  if vektorr13b(lauf+1) - (durchmesserunter1/2) lt 1 then goto, neunzwanzig6
  if vektorr13b(lauf) + (durchmesserunter1/2) gt dimccd then goto, neunzwanzig6
  if vektorr13b(lauf+1) + (durchmesserunter1/2) gt dimccd then goto, neunzwanzig6

  kreisbereichring1      = fltarr(dimccd,dimccd)
  kreisbereichring1((vektorr13b(lauf)-(durchmesserunter1/2)):$
  (vektorr13b(lauf)+(durchmesserunter1/2)-1),(vektorr13b(lauf+1)-$
  (durchmesserunter1/2)):(vektorr13b(lauf+1)+(durchmesserunter1/2)-1))$
  =kreis1

  kreisbereichring2      = fltarr(dimccd,dimccd)
  kreisbereichring2((vektorr13b(lauf)-(durchmesserunter2/2)):$
  (vektorr13b(lauf)+(durchmesserunter2/2)-1),(vektorr13b(lauf+1)-$
  (durchmesserunter2/2)):(vektorr13b(lauf+1)+(durchmesserunter2/2)-1))$
  =kreiszwei

;Erzeugung des Rings und Bestimmung der mittleren Intensitaet

  kreisringvar          = kreisbereichring1 - kreisbereichring2
  positionenuntervar    = where(kreisringvar)
  nuntervar             = (size(positionenuntervar))(1)
  yuntervar             = positionenuntervar / dimccd
  xuntervar             = positionenuntervar mod dimccd

  unterintvektorr13bvar = bild(xuntervar,yuntervar)
  summeunterintvektorr13b(laufeins)= total(unterintvektorr13bvar)
  vektorr13bunter(laufeins)      = summeunterintvektorr13b(laufeins)/nuntervar
  vektorr13bfehlerunter(laufeins)= meanabsdev(unterintvektorr13bvar)

  ringe = ringe + kreisringvar

  neunzwanzig6:
  laufeins = laufeins+1
  ;tvsc1, congrid(ringe,512,512)
  print, lauf
endfor

tvsc1, congrid(ringe,512,512)

if anzahl14 eq 0 then anzahl14 = 2
vektorr14bunter      = fltarr(anzahl14/2,1)
vektorr14bfehlerunter = fltarr(anzahl14/2,1)
summeunterintvektorr14b      = fltarr(anzahl14/2,1)

laufeins = 0

for lauf = 0, anzahl14-1, 2 do begin
  if vektorr14b(lauf) - (durchmesserunter1/2) lt 1 then goto, neunzwanzig7
  if vektorr14b(lauf+1) - (durchmesserunter1/2) lt 1 then goto, neunzwanzig7
  if vektorr14b(lauf) + (durchmesserunter1/2) gt dimccd then goto, neunzwanzig7
  if vektorr14b(lauf+1) + (durchmesserunter1/2) gt dimccd then goto, neunzwanzig7

  kreisbereichring1      = fltarr(dimccd,dimccd)
  kreisbereichring1((vektorr14b(lauf)-(durchmesserunter1/2)):$
  (vektorr14b(lauf)+(durchmesserunter1/2)-1),(vektorr14b(lauf+1)-$

```

```
(durchmesserunter1/2):(vektorr14b(lauf+1)+(durchmesserunter1/2)-1))$
=kreis1
```

```
kreisbereichring2 = fltarr(dimccd,dimccd)
kreisbereichring2((vektorr14b(lauf)-(durchmesserunter2/2)):$
(vektorr14b(lauf)+(durchmesserunter2/2)-1),(vektorr14b(lauf+1)-$
(durchmesserunter2/2)):(vektorr14b(lauf+1)+(durchmesserunter2/2)-1))$
=kreiszwei
```

```
;Erzeugung des Rings und Bestimmung der mittleren Intensitaet
```

```
kreisringvar = kreisbereichring1 - kreisbereichring2
positionenuntervar = where(kreisringvar)
nuntervar = (size(positionenuntervar))(1)
yuntervar = positionenuntervar / dimccd
xuntervar = positionenuntervar mod dimccd
```

```
unterintvektorr14bvar = bild(xuntervar,yuntervar)
summeunterintvektorr14b(laufeins)= total(unterintvektorr14bvar)
vektorr14bunter(laufeins) = summeunterintvektorr14b(laufeins)/nuntervar
vektorr14bfehlerunter(laufeins)= meanabsdev(unterintvektorr14bvar)
ringe = ringe + kreisringvar
neunzwanzig7:
laufeins = laufeins+1
;tvsc1, congrid(ringe,512,512)
print, lauf
```

```
endfor
```

```
tvsc1, congrid(ringe,512,512)
```

```
if anzahl15 eq 0 then anzahl15 = 2
vektorr15bunter = fltarr(anzahl15/2,1)
vektorr15bfehlerunter = fltarr(anzahl15/2,1)
summeunterintvektorr15b = fltarr(anzahl15/2,1)
```

```
laufeins = 0
```

```
for lauf = 0, anzahl15-1, 2 do begin
if vektor15b(lauf) - (durchmesserunter1/2) lt 1 then goto, neunzwanzig8
if vektor15b(lauf+1) - (durchmesserunter1/2) lt 1 then goto, neunzwanzig8
if vektor15b(lauf) + (durchmesserunter1/2) gt dimccd then goto, neunzwanzig8
if vektor15b(lauf+1) + (durchmesserunter1/2) gt dimccd then goto, neunzwanzig8
```

```
kreisbereichring1 = fltarr(dimccd,dimccd)
kreisbereichring1((vektorr15b(lauf)-(durchmesserunter1/2)):$
(vektorr15b(lauf)+(durchmesserunter1/2)-1),(vektorr15b(lauf+1)-$
(durchmesserunter1/2)):(vektorr15b(lauf+1)+(durchmesserunter1/2)-1))$
=kreis1
```

```
kreisbereichring2 = fltarr(dimccd,dimccd)
kreisbereichring2((vektorr15b(lauf)-(durchmesserunter2/2)):$
(vektorr15b(lauf)+(durchmesserunter2/2)-1),(vektorr15b(lauf+1)-$
(durchmesserunter2/2)):(vektorr15b(lauf+1)+(durchmesserunter2/2)-1))$
=kreiszwei
```

```
;Erzeugung des Rings und Bestimmung der mittleren Intensitaet
```

```
kreisringvar = kreisbereichring1 - kreisbereichring2
positionenuntervar = where(kreisringvar)
nuntervar = (size(positionenuntervar))(1)
```

```

yuntervar          = positionenuntervar / dimccd
xuntervar          = positionenuntervar mod dimccd

unterintvektorr15bvar = bild(xuntervar,yuntervar)
summeunterintvektorr15b(laufeins)= total(unterintvektorr15bvar)
vektorr15bunter(laufeins)      = summeunterintvektorr15b(laufeins)/nuntervar
vektorr15bfehlerunter(laufeins)= meanabsdev(unterintvektorr15bvar)
ringe = ringe + kreisringvar
neunzwanzig8:
laufeins = laufeins+1
;tvsc1, congrid(ringe,512,512)
print, lauf
endfor

tvsc1, congrid(ringe,512,512)

if anzahl16 eq 0 then anzahl16 = 2
vektorr16bunter          = fltarr(anzahl16/2,1)
vektorr16bfehlerunter   = fltarr(anzahl16/2,1)
summeunterintvektorr16b = fltarr(anzahl16/2,1)

laufeins = 0

for lauf = 0, anzahl16-1, 2 do begin
  if vektorr16b(lauf) - (durchmesserunter1/2) lt 1 then goto, neunzwanzig9
  if vektorr16b(lauf+1) - (durchmesserunter1/2) lt 1 then goto, neunzwanzig9
  if vektorr16b(lauf) + (durchmesserunter1/2) gt dimccd then goto, neunzwanzig9
  if vektorr16b(lauf+1) + (durchmesserunter1/2) gt dimccd then goto, neunzwanzig9

  kreisbereichring1      = fltarr(dimccd,dimccd)
  kreisbereichring1((vektorr16b(lauf)-(durchmesserunter1/2)):$
  (vektorr16b(lauf)+(durchmesserunter1/2)-1),(vektorr16b(lauf+1)-$
  (durchmesserunter1/2)):(vektorr16b(lauf+1)+(durchmesserunter1/2)-1))$
  =kreis1

  kreisbereichring2      = fltarr(dimccd,dimccd)
  kreisbereichring2((vektorr16b(lauf)-(durchmesserunter2/2)):$
  (vektorr16b(lauf)+(durchmesserunter2/2)-1),(vektorr16b(lauf+1)-$
  (durchmesserunter2/2)):(vektorr16b(lauf+1)+(durchmesserunter2/2)-1))$
  =kreiszwei

  ;Erzeugung des Rings und Bestimmung der mittleren Intensitaet

  kreisringvar          = kreisbereichring1 - kreisbereichring2
  positionenuntervar    = where(kreisringvar)
  nuntervar             = (size(positionenuntervar))(1)
  yuntervar             = positionenuntervar / dimccd
  xuntervar             = positionenuntervar mod dimccd

  unterintvektorr16bvar = bild(xuntervar,yuntervar)
  summeunterintvektorr16b(laufeins)= total(unterintvektorr16bvar)
  vektorr16bunter(laufeins)      = summeunterintvektorr16b(laufeins)/nuntervar
  vektorr16bfehlerunter(laufeins)= meanabsdev(unterintvektorr16bvar)
  ringe = ringe + kreisringvar
  neunzwanzig9:
  laufeins = laufeins+1
  ;tvsc1, congrid(ringe,512,512)
  print, lauf
endfor

tvsc1, congrid(ringe,512,512)

```

```

if anzahl17 eq 0 then anzahl17 = 2
vektorr17bunter          = fltarr(anzahl17/2,1)
vektorr17bfehlerunter = fltarr(anzahl17/2,1)
summeunterintvektorr17b = fltarr(anzahl17/2,1)

laufeins = 0

for lauf = 0, anzahl17-1, 2 do begin
  if vektorr17b(lauf) - (durchmesserunter1/2) lt 1 then goto, neunzwanzig10
  if vektorr17b(lauf+1) - (durchmesserunter1/2) lt 1 then goto, neunzwanzig10
  if vektorr17b(lauf) + (durchmesserunter1/2) gt dimccd then goto, neunzwanzig10
  if vektorr17b(lauf+1) + (durchmesserunter1/2) gt dimccd then goto, neunzwanzig10

  kreisbereichring1      = fltarr(dimccd,dimccd)
  kreisbereichring1((vektorr17b(lauf)-(durchmesserunter1/2)):$
(vektorr17b(lauf)+(durchmesserunter1/2)-1),(vektorr17b(lauf+1)-$
(durchmesserunter1/2)):(vektorr17b(lauf+1)+(durchmesserunter1/2)-1))$
=kreis1

  kreisbereichring2      = fltarr(dimccd,dimccd)
  kreisbereichring2((vektorr17b(lauf)-(durchmesserunter2/2)):$
(vektorr17b(lauf)+(durchmesserunter2/2)-1),(vektorr17b(lauf+1)-$
(durchmesserunter2/2)):(vektorr17b(lauf+1)+(durchmesserunter2/2)-1))$
=kreiszwei

;Erzeugung des Rings und Bestimmung der mittleren Intensitaet

  kreisringvar          = kreisbereichring1 - kreisbereichring2
  positionenuntervar    = where(kreisringvar)
  nuntervar            = (size(positionenuntervar))(1)
  yuntervar            = positionenuntervar / dimccd
  xuntervar            = positionenuntervar mod dimccd

  unterintvektorr17bvar = bild(xuntervar,yuntervar)
  summeunterintvektorr17b(laufeins)= total(unterintvektorr17bvar)
  vektorr17bunter(laufeins)      = summeunterintvektorr17b(laufeins)/nuntervar
  vektorr17bfehlerunter(laufeins)= meanabsdev(unterintvektorr17bvar)
  ringe = ringe + kreisringvar
  neunzwanzig10:
  laufeins = laufeins+1
  ;tvsc1, congrid(ringe,512,512)
  print, lauf
endfor

tvsc1, congrid(ringe,512,512)

if anzahl18 eq 0 then anzahl18 = 2
vektorr18bunter          = fltarr(anzahl18/2,1)
vektorr18bfehlerunter = fltarr(anzahl18/2,1)
summeunterintvektorr18b = fltarr(anzahl18/2,1)

laufeins = 0

for lauf = 0, anzahl18-1, 2 do begin
  if vektorr18b(lauf) - (durchmesserunter1/2) lt 1 then goto, neunzwanzig11
  if vektorr18b(lauf+1) - (durchmesserunter1/2) lt 1 then goto, neunzwanzig11
  if vektorr18b(lauf) + (durchmesserunter1/2) gt dimccd then goto, neunzwanzig11
  if vektorr18b(lauf+1) + (durchmesserunter1/2) gt dimccd then goto, neunzwanzig11

  kreisbereichring1      = fltarr(dimccd,dimccd)
  kreisbereichring1((vektorr18b(lauf)-(durchmesserunter1/2)):$

```

```
(vektorr18b(lauf)+(durchmesserunter1/2)-1),(vektorr18b(lauf+1)-$
(durchmesserunter1/2)):(vektorr18b(lauf+1)+(durchmesserunter1/2)-1))$
=kreis1
```

```
kreisbereichring2 = fltarr(dimccd,dimccd)
kreisbereichring2((vektorr18b(lauf)-(durchmesserunter2/2)):$
(vektorr18b(lauf)+(durchmesserunter2/2)-1),(vektorr18b(lauf+1)-$
(durchmesserunter2/2)):(vektorr18b(lauf+1)+(durchmesserunter2/2)-1))$
=kreiszwei
```

```
;Erzeugung des Rings und Bestimmung der mittleren Intensitaet
```

```
kreisringvar = kreisbereichring1 - kreisbereichring2
positionenuntervar = where(kreisringvar)
nuntervar = (size(positionenuntervar))(1)
yuntervar = positionenuntervar / dimccd
xuntervar = positionenuntervar mod dimccd
```

```
unterintvektorr18bvar = bild(xuntervar,yuntervar)
summeunterintvektorr18b(laufeins)= total(unterintvektorr18bvar)
vektorr18bunter(laufeins) = summeunterintvektorr18b(laufeins)/nuntervar
vektorr18bfehlerunter(laufeins)= meanabsdev(unterintvektorr18bvar)
ringe = ringe + kreisringvar
neunzwanzig11:
laufeins = laufeins+1
;tvsc1, congrid(ringe,512,512)
print, lauf
```

```
endfor
```

```
tvsc1, congrid(ringe,512,512)
```

```
if anzahlm2 eq 0 then anzahlm2 = 2
vektorm2bunter = fltarr(anzahlm2/2,1)
vektorm2bfehlerunter = fltarr(anzahlm2/2,1)
summeunterintvektorm2b = fltarr(anzahlm2/2,1)
```

```
laufeins = 0
```

```
for lauf = 0, anzahlm2-1, 2 do begin
```

```
if vektorm2b(lauf) - (durchmesserunter1/2) lt 1 then goto, dreissig
if vektorm2b(lauf+1) - (durchmesserunter1/2) lt 1 then goto, dreissig
if vektorm2b(lauf) + (durchmesserunter1/2) gt dimccd then goto, dreissig
if vektorm2b(lauf+1) + (durchmesserunter1/2) gt dimccd then goto, dreissig
```

```
kreisbereichring1 = fltarr(dimccd,dimccd)
kreisbereichring1((vektorm2b(lauf)-(durchmesserunter1/2)):$
(vektorm2b(lauf)+(durchmesserunter1/2)-1),(vektorm2b(lauf+1)-$
(durchmesserunter1/2)):(vektorm2b(lauf+1)+(durchmesserunter1/2)-1))$
=kreis1
```

```
kreisbereichring2 = fltarr(dimccd,dimccd)
kreisbereichring2((vektorm2b(lauf)-(durchmesserunter2/2)):$
(vektorm2b(lauf)+(durchmesserunter2/2)-1),(vektorm2b(lauf+1)-$
(durchmesserunter2/2)):(vektorm2b(lauf+1)+(durchmesserunter2/2)-1))$
=kreiszwei
```

```
;Erzeugung des Rings und Bestimmung der mittleren Intensitaet
```

```
kreisringvar = kreisbereichring1 - kreisbereichring2
positionenuntervar = where(kreisringvar)
nuntervar = (size(positionenuntervar))(1)
```

```

yuntervar          = positionenuntervar / dimccd
xuntervar          = positionenuntervar mod dimccd

unterintvektorm2bvar = bild(xuntervar,yuntervar)
summeunterintvektorm2b(laufeins)= total(unterintvektorm2bvar)
vektorm2bunter(laufeins)      = summeunterintvektorm2b(laufeins)/nuntervar
vektorm2bfehlerunter(laufeins)= meanabsdev(unterintvektorm2bvar)
ringe = ringe + kreisringvar
dreissig:
laufeins = laufeins+1
;tvsc1, congrid(ringe,512,512)
print, lauf
endfor

tvsc1, congrid(ringe,512,512)

if anzahlm3 eq 0 then anzahlm3 = 2
vektorm3bunter          = fltarr(anzahlm3/2,1)
vektorm3bfehlerunter = fltarr(anzahlm3/2,1)
summeunterintvektorm3b = fltarr(anzahlm3/2,1)

laufeins = 0

for lauf = 0, anzahlm3-1, 2 do begin
  if vektorm3b(lauf) - (durchmesserunter1/2) lt 1 then goto, eindreissig
  if vektorm3b(lauf+1) - (durchmesserunter1/2) lt 1 then goto, eindreissig
  if vektorm3b(lauf) + (durchmesserunter1/2) gt dimccd then goto, eindreissig
  if vektorm3b(lauf+1) + (durchmesserunter1/2) gt dimccd then goto, eindreissig

  kreisbereichring1      = fltarr(dimccd,dimccd)
  kreisbereichring1((vektorm3b(lauf)-(durchmesserunter1/2)):$
  (vektorm3b(lauf)+(durchmesserunter1/2)-1),(vektorm3b(lauf+1)-$
  (durchmesserunter1/2)):(vektorm3b(lauf+1)+(durchmesserunter1/2)-1))$
  =kreis1

  kreisbereichring2      = fltarr(dimccd,dimccd)
  kreisbereichring2((vektorm3b(lauf)-(durchmesserunter2/2)):$
  (vektorm3b(lauf)+(durchmesserunter2/2)-1),(vektorm3b(lauf+1)-$
  (durchmesserunter2/2)):(vektorm3b(lauf+1)+(durchmesserunter2/2)-1))$
  =kreiszwei

  ;Erzeugung des Rings und Bestimmung der mittleren Intensitaet

  kreisringvar          = kreisbereichring1 - kreisbereichring2
  positionenuntervar    = where(kreisringvar)
  nuntervar             = (size(positionenuntervar))(1)
  yuntervar             = positionenuntervar / dimccd
  xuntervar             = positionenuntervar mod dimccd

  unterintvektorm3bvar = bild(xuntervar,yuntervar)
  summeunterintvektorm3b(laufeins)= total(unterintvektorm3bvar)
  vektorm3bunter(laufeins)      = summeunterintvektorm3b(laufeins)/nuntervar
  vektorm3bfehlerunter(laufeins)= meanabsdev(unterintvektorm3bvar)
  ringe = ringe + kreisringvar
  eindreissig:
  laufeins = laufeins+1
  ;tvsc1, congrid(ringe,512,512)
  print, lauf
endfor

tvsc1, congrid(ringe,512,512)

```

```

if anzahlm4 eq 0 then anzahlm4 = 2
vektorm4bunter          = fltarr(anzahlm4/2,1)
vektorm4bfehlerunter = fltarr(anzahlm4/2,1)
summeunterintvektorm4b  = fltarr(anzahlm4/2,1)

laufeins = 0

for lauf = 0, anzahlm4-1, 2 do begin
  if vektorm4b(lauf) - (durchmesserunter1/2) lt 1 then goto, zweidreissig
  if vektorm4b(lauf+1) - (durchmesserunter1/2) lt 1 then goto, zweidreissig
  if vektorm4b(lauf) + (durchmesserunter1/2) gt dimccd then goto, zweidreissig
  if vektorm4b(lauf+1) + (durchmesserunter1/2) gt dimccd then goto, zweidreissig

  kreisbereichring1      = fltarr(dimccd,dimccd)
  kreisbereichring1((vektorm4b(lauf)-(durchmesserunter1/2)):$
(vektorm4b(lauf)+(durchmesserunter1/2)-1),(vektorm4b(lauf+1)-$
(durchmesserunter1/2)):(vektorm4b(lauf+1)+(durchmesserunter1/2)-1))$
=kreis1

  kreisbereichring2      = fltarr(dimccd,dimccd)
  kreisbereichring2((vektorm4b(lauf)-(durchmesserunter2/2)):$
(vektorm4b(lauf)+(durchmesserunter2/2)-1),(vektorm4b(lauf+1)-$
(durchmesserunter2/2)):(vektorm4b(lauf+1)+(durchmesserunter2/2)-1))$
=kreiszwei

;Erzeugung des Rings und Bestimmung der mittleren Intensitaet

  kreisringvar          = kreisbereichring1 - kreisbereichring2
  positionenuntervar    = where(kreisringvar)
  nuntervar             = (size(positionenuntervar))(1)
  yuntervar             = positionenuntervar / dimccd
  xuntervar             = positionenuntervar mod dimccd

  unterintvektorm4bvar = bild(xuntervar,yuntervar)
  summeunterintvektorm4b(laufeins)= total(unterintvektorm4bvar)
  vektorm4bunter(laufeins)      = summeunterintvektorm4b(laufeins)/nuntervar
  vektorm4bfehlerunter(laufeins)= meanabsdev(unterintvektorm4bvar)
  ringe = ringe + kreisringvar
  zweidreissig:
  laufeins = laufeins+1
  ;tvsc1, congrid(ringe,512,512)
  print, lauf
endfor

tvsc1, congrid(ringe,512,512)

```

```

if anzahlm5 eq 0 then anzahlm5 = 2
vektorm5bunter          = fltarr(anzahlm5/2,1)
vektorm5bfehlerunter = fltarr(anzahlm5/2,1)
summeunterintvektorm5b  = fltarr(anzahlm5/2,1)

laufeins = 0

for lauf = 0, anzahlm5-1, 2 do begin
  if vektorm5b(lauf) - (durchmesserunter1/2) lt 1 then goto, dreidreissig
  if vektorm5b(lauf+1) - (durchmesserunter1/2) lt 1 then goto, dreidreissig
  if vektorm5b(lauf) + (durchmesserunter1/2) gt dimccd then goto, dreidreissig
  if vektorm5b(lauf+1) + (durchmesserunter1/2) gt dimccd then goto, dreidreissig

  kreisbereichring1      = fltarr(dimccd,dimccd)
  kreisbereichring1((vektorm5b(lauf)-(durchmesserunter1/2)):$

```

```

(vektorm5b(lauf)+(durchmesserunter1/2)-1),(vektorm5b(lauf+1)-$
(durchmesserunter1/2)):(vektorm5b(lauf+1)+(durchmesserunter1/2)-1))$
=kreis1

kreisbereichring2      = fltarr(dimccd,dimccd)
kreisbereichring2((vektorm5b(lauf)-(durchmesserunter2/2)):$
(vektorm5b(lauf)+(durchmesserunter2/2)-1),(vektorm5b(lauf+1)-$
(durchmesserunter2/2)):(vektorm5b(lauf+1)+(durchmesserunter2/2)-1))$
=kreiszwei

;Erzeugung des Rings und Bestimmung der mittleren Intensitaet

kreisringvar          = kreisbereichring1 - kreisbereichring2
positionenuntervar    = where(kreisringvar)
nuntervar             = (size(positionenuntervar))(1)
yuntervar             = positionenuntervar / dimccd
xuntervar             = positionenuntervar mod dimccd

unterintvektorm5bvar = bild(xuntervar,yuntervar)
summeunterintvektorm5b(laufeins)= total(unterintvektorm5bvar)
vektorm5bunter(laufeins)      = summeunterintvektorm5b(laufeins)/nuntervar
vektorm5bfehlerunter(laufeins)= meanabsdev(unterintvektorm5bvar)
ringe = ringe + kreisringvar
dreidreissig:
laufeins = laufeins+1
;tvsc1, congrid(ringe,512,512)
print, lauf
endfor

tvsc1, congrid(ringe,512,512)

if anzahlm6 eq 0 then anzahlm6 = 2
vektorm6bunter      = fltarr(anzahlm6/2,1)
vektorm6bfehlerunter = fltarr(anzahlm6/2,1)
summeunterintvektorm6b      = fltarr(anzahlm6/2,1)

laufeins = 0

for lauf = 0, anzahlm6-1, 2 do begin
  if vektorm6b(lauf) - (durchmesserunter1/2) lt 1 then goto, dreidreissig1
  if vektorm6b(lauf+1) - (durchmesserunter1/2) lt 1 then goto, dreidreissig1
  if vektorm6b(lauf) + (durchmesserunter1/2) gt dimccd then goto, dreidreissig1
  if vektorm6b(lauf+1) + (durchmesserunter1/2) gt dimccd then goto, dreidreissig1

  kreisbereichring1      = fltarr(dimccd,dimccd)
  kreisbereichring1((vektorm6b(lauf)-(durchmesserunter1/2)):$
(vektorm6b(lauf)+(durchmesserunter1/2)-1),(vektorm6b(lauf+1)-$
(durchmesserunter1/2)):(vektorm6b(lauf+1)+(durchmesserunter1/2)-1))$
=kreis1

  kreisbereichring2      = fltarr(dimccd,dimccd)
  kreisbereichring2((vektorm6b(lauf)-(durchmesserunter2/2)):$
(vektorm6b(lauf)+(durchmesserunter2/2)-1),(vektorm6b(lauf+1)-$
(durchmesserunter2/2)):(vektorm6b(lauf+1)+(durchmesserunter2/2)-1))$
=kreiszwei

;Erzeugung des Rings und Bestimmung der mittleren Intensitaet

kreisringvar          = kreisbereichring1 - kreisbereichring2
positionenuntervar    = where(kreisringvar)
nuntervar             = (size(positionenuntervar))(1)
yuntervar             = positionenuntervar / dimccd

```

```

xuntervar                = positionenuntervar mod dimccd

unterintvektorm6bvar = bild(xuntervar,yuntervar)
summeunterintvektorm6b(laufeins)= total(unterintvektorm6bvar)
vektorm6bunter(laufeins)      = summeunterintvektorm6b(laufeins)/nuntervar
vektorm6bfehlerunter(laufeins)= meanabsdev(unterintvektorm6bvar)
ringe = ringe + kreisringvar
dreidreissig1:
laufeins = laufeins+1
;tvsc1, congrid(ringe,512,512)
print, lauf
endfor

tvsc1, congrid(ringe,512,512)

if anzahlm7 eq 0 then anzahlm7 = 2
vektorm7bunter          = fltarr(anzahlm7/2,1)
vektorm7bfehlerunter = fltarr(anzahlm7/2,1)
summeunterintvektorm7b = fltarr(anzahlm7/2,1)

laufeins = 0

for lauf = 0, anzahlm7-1, 2 do begin
  if vektorm7b(lauf) - (durchmesserunter1/2) lt 1 then goto, dreidreissig2
  if vektorm7b(lauf+1) - (durchmesserunter1/2) lt 1 then goto, dreidreissig2
  if vektorm7b(lauf) + (durchmesserunter1/2) gt dimccd then goto, dreidreissig2
  if vektorm7b(lauf+1) + (durchmesserunter1/2) gt dimccd then goto, dreidreissig2

  kreisbereichring1      = fltarr(dimccd,dimccd)
  kreisbereichring1((vektorm7b(lauf)-(durchmesserunter1/2)):$
(vektorm7b(lauf)+(durchmesserunter1/2)-1),(vektorm7b(lauf+1)-$
(durchmesserunter1/2)):(vektorm7b(lauf+1)+(durchmesserunter1/2)-1))$
=kreis1

  kreisbereichring2      = fltarr(dimccd,dimccd)
  kreisbereichring2((vektorm7b(lauf)-(durchmesserunter2/2)):$
(vektorm7b(lauf)+(durchmesserunter2/2)-1),(vektorm7b(lauf+1)-$
(durchmesserunter2/2)):(vektorm7b(lauf+1)+(durchmesserunter2/2)-1))$
=kreiszwei

;Erzeugung des Rings und Bestimmung der mittleren Intensitaet

  kreisringvar          = kreisbereichring1 - kreisbereichring2
  positionenuntervar    = where(kreisringvar)
  nuntervar             = (size(positionenuntervar))(1)
  yuntervar             = positionenuntervar / dimccd
  xuntervar             = positionenuntervar mod dimccd

  unterintvektorm7bvar = bild(xuntervar,yuntervar)
  summeunterintvektorm7b(laufeins)= total(unterintvektorm7bvar)
  vektorm7bunter(laufeins)      = summeunterintvektorm7b(laufeins)/nuntervar
  vektorm7bfehlerunter(laufeins)= meanabsdev(unterintvektorm7bvar)
  ringe = ringe + kreisringvar
  dreidreissig2:
  laufeins = laufeins+1
  ;tvsc1, congrid(ringe,512,512)
  print, lauf
endfor

tvsc1, congrid(ringe,512,512)

if anzahlm8 eq 0 then anzahlm8 = 2

```

```

vektorm8bunter          = fltarr(anzahlm8/2,1)
vektorm8bfehlerunter = fltarr(anzahlm8/2,1)
summeunterintvektorm8b  = fltarr(anzahlm8/2,1)

laufeins = 0

for lauf = 0, anzahlm8-1, 2 do begin
  if vektorm8b(lauf) - (durchmesserunter1/2) lt 1 then goto, dreidreissig3
  if vektorm8b(lauf+1) - (durchmesserunter1/2) lt 1 then goto, dreidreissig3
  if vektorm8b(lauf) + (durchmesserunter1/2) gt dimccd then goto, dreidreissig3
  if vektorm8b(lauf+1) + (durchmesserunter1/2) gt dimccd then goto, dreidreissig3

  kreisbereichring1      = fltarr(dimccd,dimccd)
  kreisbereichring1((vektorm8b(lauf)-(durchmesserunter1/2)):$
(vektorm8b(lauf)+(durchmesserunter1/2)-1),(vektorm8b(lauf+1)-$
(durchmesserunter1/2)):(vektorm8b(lauf+1)+(durchmesserunter1/2)-1))$
=kreis1

  kreisbereichring2      = fltarr(dimccd,dimccd)
  kreisbereichring2((vektorm8b(lauf)-(durchmesserunter2/2)):$
(vektorm8b(lauf)+(durchmesserunter2/2)-1),(vektorm8b(lauf+1)-$
(durchmesserunter2/2)):(vektorm8b(lauf+1)+(durchmesserunter2/2)-1))$
=kreiszwei

;Erzeugung des Rings und Bestimmung der mittleren Intensitaet

  kreisringvar          = kreisbereichring1 - kreisbereichring2
  positionenuntervar    = where(kreisringvar)
  nuntervar             = (size(positionenuntervar))(1)
  yuntervar             = positionenuntervar / dimccd
  xuntervar             = positionenuntervar mod dimccd

  unterintvektorm8bvar = bild(xuntervar,yuntervar)
  summeunterintvektorm8b(laufeins)= total(unterintvektorm8bvar)
  vektorm8bunter(laufeins) = summeunterintvektorm8b(laufeins)/nuntervar
  vektorm8bfehlerunter(laufeins)= meanabsdev(unterintvektorm8bvar)

  ringe = ringe + kreisringvar

  dreidreissig3:
  laufeins = laufeins+1
  ;tvsc1, congrid(ringe,512,512)
  print, lauf
endfor

tvsc1, congrid(ringe,512,512)

if anzahlm9 eq 0 then anzahlm9 = 2
vektorm9bunter          = fltarr(anzahlm9/2,1)
vektorm9bfehlerunter = fltarr(anzahlm9/2,1)
summeunterintvektorm9b  = fltarr(anzahlm9/2,1)

laufeins = 0

for lauf = 0, anzahlm9-1, 2 do begin
  if vektorm9b(lauf) - (durchmesserunter1/2) lt 1 then goto, dreidreissig4
  if vektorm9b(lauf+1) - (durchmesserunter1/2) lt 1 then goto, dreidreissig4
  if vektorm9b(lauf) + (durchmesserunter1/2) gt dimccd then goto, dreidreissig4
  if vektorm9b(lauf+1) + (durchmesserunter1/2) gt dimccd then goto, dreidreissig4

  kreisbereichring1      = fltarr(dimccd,dimccd)

```

```

kreisbereichring1((vektorm9b(lauf)-(durchmesserunter1/2)):$
(vektorm9b(lauf)+(durchmesserunter1/2)-1),(vektorm9b(lauf+1)-$
(durchmesserunter1/2)):(vektorm9b(lauf+1)+(durchmesserunter1/2)-1))$
=kreis1

kreisbereichring2      = fltarr(dimccd,dimccd)
kreisbereichring2((vektorm9b(lauf)-(durchmesserunter2/2)):$
(vektorm9b(lauf)+(durchmesserunter2/2)-1),(vektorm9b(lauf+1)-$
(durchmesserunter2/2)):(vektorm9b(lauf+1)+(durchmesserunter2/2)-1))$
=kreiszwei

;Erzeugung des Rings und Bestimmung der mittleren Intensitaet

kreisringvar          = kreisbereichring1 - kreisbereichring2
positionenuntervar   = where(kreisringvar)
nuntervar            = (size(positionenuntervar))(1)
yuntervar            = positionenuntervar / dimccd
xuntervar            = positionenuntervar mod dimccd

unterintvektorm9bvar = bild(xuntervar,yuntervar)
summeunterintvektorm9b(laufeins)= total(unterintvektorm9bvar)
vektorm9bunter(laufeins)      = summeunterintvektorm9b(laufeins)/nuntervar
vektorm9bfehlerunter(laufeins)= meanabsdev(unterintvektorm9bvar)
ringe = ringe + kreisringvar
dreidreissig4:
laufeins = laufeins+1
;tvsc1, congrid(ringe,512,512)
print, lauf
endfor

tvsc1, congrid(ringe,512,512)

if anzahlm10 eq 0 then anzahlm10 = 2
vektorm10bunter      = fltarr(anzahlm10/2,1)
vektorm10bfehlerunter = fltarr(anzahlm10/2,1)
summeunterintvektorm10b = fltarr(anzahlm10/2,1)

laufeins = 0

for lauf = 0, anzahlm10-1, 2 do begin
if vektorm10b(lauf) - (durchmesserunter1/2) lt 1 then goto, dreidreissig5
if vektorm10b(lauf+1) - (durchmesserunter1/2) lt 1 then goto, dreidreissig5
if vektorm10b(lauf) + (durchmesserunter1/2) gt dimccd then goto, dreidreissig5
if vektorm10b(lauf+1) + (durchmesserunter1/2) gt dimccd then goto, dreidreissig5

kreisbereichring1      = fltarr(dimccd,dimccd)
kreisbereichring1((vektorm10b(lauf)-(durchmesserunter1/2)):$
(vektorm10b(lauf)+(durchmesserunter1/2)-1),(vektorm10b(lauf+1)-$
(durchmesserunter1/2)):(vektorm10b(lauf+1)+(durchmesserunter1/2)-1))$
=kreis1

kreisbereichring2      = fltarr(dimccd,dimccd)
kreisbereichring2((vektorm10b(lauf)-(durchmesserunter2/2)):$
(vektorm10b(lauf)+(durchmesserunter2/2)-1),(vektorm10b(lauf+1)-$
(durchmesserunter2/2)):(vektorm10b(lauf+1)+(durchmesserunter2/2)-1))$
=kreiszwei

;Erzeugung des Rings und Bestimmung der mittleren Intensitaet

kreisringvar          = kreisbereichring1 - kreisbereichring2
positionenuntervar   = where(kreisringvar)
nuntervar            = (size(positionenuntervar))(1)

```

```

yuntervar          = positionenuntervar / dimccd
xuntervar          = positionenuntervar mod dimccd

unterintvektorm10bvar = bild(xuntervar,yuntervar)
summeunterintvektorm10b(laufeins)= total(unterintvektorm10bvar)
vektorm10bunter(laufeins)      = summeunterintvektorm10b(laufeins)/nuntervar
vektorm10bfehlerunter(laufeins)= meanabsdev(unterintvektorm10bvar)
ringe = ringe + kreisringvar
dreidreissig5:
laufeins = laufeins+1
;tvsc1, congrid(ringe,512,512)
print, lauf
endfor

tvsc1, congrid(ringe,512,512)

if anzahlm11 eq 0 then anzahlm11 = 2
vektorm11bunter          = fltarr(anzahlm11/2,1)
vektorm11bfehlerunter    = fltarr(anzahlm11/2,1)
summeunterintvektorm11b = fltarr(anzahlm11/2,1)

laufeins = 0

for lauf = 0, anzahlm11-1, 2 do begin
  if vektorm11b(lauf) - (durchmesserunter1/2) lt 1 then goto, dreidreissig6
  if vektorm11b(lauf+1) - (durchmesserunter1/2) lt 1 then goto, dreidreissig6
  if vektorm11b(lauf) + (durchmesserunter1/2) gt dimccd then goto, dreidreissig6
  if vektorm11b(lauf+1) + (durchmesserunter1/2) gt dimccd then goto, dreidreissig6

  kreisbereichring1      = fltarr(dimccd,dimccd)
  kreisbereichring1((vektorm11b(lauf)-(durchmesserunter1/2)):$
  (vektorm11b(lauf)+(durchmesserunter1/2)-1),(vektorm11b(lauf+1)-$
  (durchmesserunter1/2)):(vektorm11b(lauf+1)+(durchmesserunter1/2)-1))$
  =kreis1

  kreisbereichring2      = fltarr(dimccd,dimccd)
  kreisbereichring2((vektorm11b(lauf)-(durchmesserunter2/2)):$
  (vektorm11b(lauf)+(durchmesserunter2/2)-1),(vektorm11b(lauf+1)-$
  (durchmesserunter2/2)):(vektorm11b(lauf+1)+(durchmesserunter2/2)-1))$
  =kreiszwei

  ;Erzeugung des Rings und Bestimmung der mittleren Intensitaet

  kreisringvar          = kreisbereichring1 - kreisbereichring2
  positionenuntervar    = where(kreisringvar)
  nuntervar             = (size(positionenuntervar))(1)
  yuntervar             = positionenuntervar / dimccd
  xuntervar             = positionenuntervar mod dimccd

  unterintvektorm11bvar = bild(xuntervar,yuntervar)
  summeunterintvektorm11b(laufeins)= total(unterintvektorm11bvar)
  vektorm11bunter(laufeins)      = summeunterintvektorm11b(laufeins)/nuntervar
  vektorm11bfehlerunter(laufeins)= meanabsdev(unterintvektorm11bvar)
  ringe = ringe + kreisringvar
  dreidreissig6:
  laufeins = laufeins+1
  ;tvsc1, congrid(ringe,512,512)
  print, lauf
endfor

tvsc1, congrid(ringe,512,512)

```

```

if anzahlm12 eq 0 then anzahlm12 = 2
vektorm12bunter          = fltarr(anzahlm12/2,1)
vektorm12bfehlerunter    = fltarr(anzahlm12/2,1)
summeunterintvektorm12b = fltarr(anzahlm12/2,1)

laufeins = 0

for lauf = 0, anzahlm12-1, 2 do begin
  if vektorm12b(lauf) - (durchmesserunter1/2) lt 1 then goto, dreidreissig7
  if vektorm12b(lauf+1) - (durchmesserunter1/2) lt 1 then goto, dreidreissig7
  if vektorm12b(lauf) + (durchmesserunter1/2) gt dimccd then goto, dreidreissig7
  if vektorm12b(lauf+1) + (durchmesserunter1/2) gt dimccd then goto, dreidreissig7

  kreisbereichring1      = fltarr(dimccd,dimccd)
  kreisbereichring1((vektorm12b(lauf)-(durchmesserunter1/2)):$
(vektorm12b(lauf)+(durchmesserunter1/2)-1),(vektorm12b(lauf+1)-$
(durchmesserunter1/2)):(vektorm12b(lauf+1)+(durchmesserunter1/2)-1))$
  =kreis1

  kreisbereichring2      = fltarr(dimccd,dimccd)
  kreisbereichring2((vektorm12b(lauf)-(durchmesserunter2/2)):$
(vektorm12b(lauf)+(durchmesserunter2/2)-1),(vektorm12b(lauf+1)-$
(durchmesserunter2/2)):(vektorm12b(lauf+1)+(durchmesserunter2/2)-1))$
  =kreiszwei

  ;Erzeugung des Rings und Bestimmung der mittleren Intensitaet

  kreisringvar          = kreisbereichring1 - kreisbereichring2
  positionenuntervar    = where(kreisringvar)
  nuntervar             = (size(positionenuntervar))(1)
  yuntervar             = positionenuntervar / dimccd
  xuntervar             = positionenuntervar mod dimccd

  unterintvektorm12bvar = bild(xuntervar,yuntervar)
  summeunterintvektorm12b(laufeins)= total(unterintvektorm12bvar)
  vektorm12bunter(laufeins)  = summeunterintvektorm12b(laufeins)/nuntervar
  vektorm12bfehlerunter(laufeins)= meanabsdev(unterintvektorm12bvar)
  ringe = ringe + kreisringvar
  dreidreissig7:
  laufeins = laufeins+1
  ;tvsc1, congrid(ringe,512,512)
  print, lauf
endfor

tvsc1, congrid(ringe,512,512)

```

```

if anzahlm13 eq 0 then anzahlm13 = 2
vektorm13bunter          = fltarr(anzahlm13/2,1)
vektorm13bfehlerunter    = fltarr(anzahlm13/2,1)
summeunterintvektorm13b = fltarr(anzahlm13/2,1)

laufeins = 0

for lauf = 0, anzahlm13-1, 2 do begin
  if vektorm13b(lauf) - (durchmesserunter1/2) lt 1 then goto, dreidreissig8
  if vektorm13b(lauf+1) - (durchmesserunter1/2) lt 1 then goto, dreidreissig8
  if vektorm13b(lauf) + (durchmesserunter1/2) gt dimccd then goto, dreidreissig8
  if vektorm13b(lauf+1) + (durchmesserunter1/2) gt dimccd then goto, dreidreissig8

  kreisbereichring1      = fltarr(dimccd,dimccd)
  kreisbereichring1((vektorm13b(lauf)-(durchmesserunter1/2)):$
(vektorm13b(lauf)+(durchmesserunter1/2)-1),(vektorm13b(lauf+1)-$

```

```
(durchmesserunter1/2):(vektorm13b(lauf+1)+(durchmesserunter1/2)-1))$
=kreis1
```

```
kreisbereichring2 = fltarr(dimccd,dimccd)
kreisbereichring2((vektorm13b(lauf)-(durchmesserunter2/2)):$
(vektorm13b(lauf)+(durchmesserunter2/2)-1),(vektorm13b(lauf+1)-$
(durchmesserunter2/2)):(vektorm13b(lauf+1)+(durchmesserunter2/2)-1))$
=kreiszwei
```

```
;Erzeugung des Rings und Bestimmung der mittleren Intensitaet
```

```
kreisringvar = kreisbereichring1 - kreisbereichring2
positionenuntervar = where(kreisringvar)
nuntervar = (size(positionenuntervar))(1)
yuntervar = positionenuntervar / dimccd
xuntervar = positionenuntervar mod dimccd
```

```
unterintvektorm13bvar = bild(xuntervar,yuntervar)
summeunterintvektorm13b(laufeins)= total(unterintvektorm13bvar)
vektorm13bunter(laufeins) = summeunterintvektorm13b(laufeins)/nuntervar
vektorm13bfehlerunter(laufeins)= meanabsdev(unterintvektorm13bvar)
ringe = ringe + kreisringvar
dreidreissig8:
laufeins = laufeins+1
;tvsc1, congrid(ringe,512,512)
print, lauf
```

```
endfor
```

```
tvsc1, congrid(ringe,512,512)
```

```
if anzahlm14 eq 0 then anzahlm14 = 2
vektorm14bunter = fltarr(anzahlm14/2,1)
vektorm14bfehlerunter = fltarr(anzahlm14/2,1)
summeunterintvektorm14b = fltarr(anzahlm14/2,1)
```

```
laufeins = 0
```

```
for lauf = 0, anzahlm14-1, 2 do begin
if vektorm14b(lauf) - (durchmesserunter1/2) lt 1 then goto, dreidreissig9
if vektorm14b(lauf+1) - (durchmesserunter1/2) lt 1 then goto, dreidreissig9
if vektorm14b(lauf) + (durchmesserunter1/2) gt dimccd then goto, dreidreissig9
if vektorm14b(lauf+1) + (durchmesserunter1/2) gt dimccd then goto, dreidreissig9
```

```
kreisbereichring1 = fltarr(dimccd,dimccd)
kreisbereichring1((vektorm14b(lauf)-(durchmesserunter1/2)):$
(vektorm14b(lauf)+(durchmesserunter1/2)-1),(vektorm14b(lauf+1)-$
(durchmesserunter1/2)):(vektorm14b(lauf+1)+(durchmesserunter1/2)-1))$
=kreis1
```

```
kreisbereichring2 = fltarr(dimccd,dimccd)
kreisbereichring2((vektorm14b(lauf)-(durchmesserunter2/2)):$
(vektorm14b(lauf)+(durchmesserunter2/2)-1),(vektorm14b(lauf+1)-$
(durchmesserunter2/2)):(vektorm14b(lauf+1)+(durchmesserunter2/2)-1))$
=kreiszwei
```

```
;Erzeugung des Rings und Bestimmung der mittleren Intensitaet
```

```
kreisringvar = kreisbereichring1 - kreisbereichring2
positionenuntervar = where(kreisringvar)
nuntervar = (size(positionenuntervar))(1)
yuntervar = positionenuntervar / dimccd
```

```

xuntervar                = positionenuntervar mod dimccd

unterintvektorm14bvar = bild(xuntervar,yuntervar)
summeunterintvektorm14b(laufeins)= total(unterintvektorm14bvar)
vektorm14bunter(laufeins)    = summeunterintvektorm14b(laufeins)/nuntervar
vektorm14bfehlerunter(laufeins)= meanabsdev(unterintvektorm14bvar)
ringe = ringe + kreisringvar
dreidreissig9:
laufeins = laufeins+1
;tvsc1, congrid(ringe,512,512)
print, lauf
endfor

tvsc1, congrid(ringe,512,512)

if anzahlm15 eq 0 then anzahlm15 = 2
vektorm15bunter          = fltarr(anzahlm15/2,1)
vektorm15bfehlerunter    = fltarr(anzahlm15/2,1)
summeunterintvektorm15b = fltarr(anzahlm15/2,1)

laufeins = 0

for lauf = 0, anzahlm15-1, 2 do begin
  if vektorm15b(lauf) - (durchmesserunter1/2) lt 1 then goto, dreidreissig10
  if vektorm15b(lauf+1) - (durchmesserunter1/2) lt 1 then goto, dreidreissig10
  if vektorm15b(lauf) + (durchmesserunter1/2) gt dimccd then goto, dreidreissig10
  if vektorm15b(lauf+1) + (durchmesserunter1/2) gt dimccd then goto, dreidreissig10

  kreisbereichring1      = fltarr(dimccd,dimccd)
  kreisbereichring1((vektorm15b(lauf)-(durchmesserunter1/2)):$
(vektorm15b(lauf)+(durchmesserunter1/2)-1),(vektorm15b(lauf+1)-$
(durchmesserunter1/2)):(vektorm15b(lauf+1)+(durchmesserunter1/2)-1))$
=kreis1

  kreisbereichring2      = fltarr(dimccd,dimccd)
  kreisbereichring2((vektorm15b(lauf)-(durchmesserunter2/2)):$
(vektorm15b(lauf)+(durchmesserunter2/2)-1),(vektorm15b(lauf+1)-$
(durchmesserunter2/2)):(vektorm15b(lauf+1)+(durchmesserunter2/2)-1))$
=kreiszwei

;Erzeugung des Rings und Bestimmung der mittleren Intensitaet

  kreisringvar          = kreisbereichring1 - kreisbereichring2
  positionenuntervar    = where(kreisringvar)
  nuntervar             = (size(positionenuntervar))(1)
  yuntervar             = positionenuntervar / dimccd
  xuntervar             = positionenuntervar mod dimccd

  unterintvektorm15bvar = bild(xuntervar,yuntervar)
  summeunterintvektorm15b(laufeins)= total(unterintvektorm15bvar)
  vektorm15bunter(laufeins)    = summeunterintvektorm15b(laufeins)/nuntervar
  vektorm15bfehlerunter(laufeins)= meanabsdev(unterintvektorm15bvar)
  ringe = ringe + kreisringvar
  dreidreissig10:
  laufeins = laufeins+1
  ;tvsc1, congrid(ringe,512,512)
  print, lauf
endfor

tvsc1, congrid(ringe,512,512)

if anzahlm16 eq 0 then anzahlm16 = 2

```

```
vektorm16bunter          = fltarr(anzahlm16/2,1)
vektorm16bfehlerunter   = fltarr(anzahlm16/2,1)
summeunterintvektorm16b = fltarr(anzahlm16/2,1)
```

laufeins = 0

```
for lauf = 0, anzahlm16-1, 2 do begin
  if vektorm16b(lauf) - (durchmesserunter1/2) lt 1 then goto, dreidreissig11
  if vektorm16b(lauf+1) - (durchmesserunter1/2) lt 1 then goto, dreidreissig11
  if vektorm16b(lauf) + (durchmesserunter1/2) gt dimccd then goto, dreidreissig11
  if vektorm16b(lauf+1) + (durchmesserunter1/2) gt dimccd then goto, dreidreissig11
```

```
  kreisbereichring1      = fltarr(dimccd,dimccd)
  kreisbereichring1((vektorm16b(lauf)-(durchmesserunter1/2)):$
(vektorm16b(lauf)+(durchmesserunter1/2)-1),(vektorm16b(lauf+1)-$
(durchmesserunter1/2)):(vektorm16b(lauf+1)+(durchmesserunter1/2)-1))$
  =kreis1
```

```
  kreisbereichring2      = fltarr(dimccd,dimccd)
  kreisbereichring2((vektorm16b(lauf)-(durchmesserunter2/2)):$
(vektorm16b(lauf)+(durchmesserunter2/2)-1),(vektorm16b(lauf+1)-$
(durchmesserunter2/2)):(vektorm16b(lauf+1)+(durchmesserunter2/2)-1))$
  =kreiszwei
```

;Erzeugung des Rings und Bestimmung der mittleren Intensitaet

```
  kreisringvar           = kreisbereichring1 - kreisbereichring2
  positionenuntervar     = where(kreisringvar)
  nuntervar              = (size(positionenuntervar))(1)
  yuntervar              = positionenuntervar / dimccd
  xuntervar              = positionenuntervar mod dimccd
```

```
  unterintvektorm16bvar = bild(xuntervar,yuntervar)
  summeunterintvektorm16b(laufeins)= total(unterintvektorm16bvar)
  vektorm16bunter(laufeins)  = summeunterintvektorm16b(laufeins)/nuntervar
  vektorm16bfehlerunter(laufeins)= meanabsdev(unterintvektorm16bvar)
  ringe = ringe + kreisringvar
  dreidreissig11:
  laufeins = laufeins+1
  ;tvsc1, congrid(ringe,512,512)
  print, lauf
```

endfor

tvsc1, congrid(ringe,512,512)

```
if anzahlm17 eq 0 then anzahlm17 = 2
vektorm17bunter          = fltarr(anzahlm17/2,1)
vektorm17bfehlerunter   = fltarr(anzahlm17/2,1)
summeunterintvektorm17b = fltarr(anzahlm17/2,1)
```

laufeins = 0

```
for lauf = 0, anzahlm17-1, 2 do begin
  if vektorm17b(lauf) - (durchmesserunter1/2) lt 1 then goto, dreidreissig12
  if vektorm17b(lauf+1) - (durchmesserunter1/2) lt 1 then goto, dreidreissig12
  if vektorm17b(lauf) + (durchmesserunter1/2) gt dimccd then goto, dreidreissig12
  if vektorm17b(lauf+1) + (durchmesserunter1/2) gt dimccd then goto, dreidreissig12
```

```
  kreisbereichring1      = fltarr(dimccd,dimccd)
  kreisbereichring1((vektorm17b(lauf)-(durchmesserunter1/2)):$
(vektorm17b(lauf)+(durchmesserunter1/2)-1),(vektorm17b(lauf+1)-$
(durchmesserunter1/2)):(vektorm17b(lauf+1)+(durchmesserunter1/2)-1))$
```

```

=kreis1

kreisbereichring2      = fltarr(dimccd,dimccd)
kreisbereichring2((vektorm17b(lauf)-(durchmesserunter2/2)):$
(vektorm17b(lauf)+(durchmesserunter2/2)-1),(vektorm17b(lauf+1)-$
(durchmesserunter2/2)):(vektorm17b(lauf+1)+(durchmesserunter2/2)-1))$
=kreiszwei

;Erzeugung des Rings und Bestimmung der mittleren Intensitaet

kreisringvar          = kreisbereichring1 - kreisbereichring2
positionenuntervar    = where(kreisringvar)
nuntervar             = (size(positionenuntervar))(1)
yuntervar            = positionenuntervar / dimccd
xuntervar            = positionenuntervar mod dimccd

unterintvektorm17bvar = bild(xuntervar,yuntervar)
summeunterintvektorm17b(laufeins)= total(unterintvektorm17bvar)
vektorm17bunter(laufeins)      = summeunterintvektorm17b(laufeins)/nuntervar
vektorm17bfehlerunter(laufeins)= meanabsdev(unterintvektorm17bvar)
ringe = ringe + kreisringvar
dreidreissig12:
laufeins = laufeins+1
;tvsc1, congrid(ringe,512,512)
print, lauf
endfor

tvsc1, congrid(ringe,512,512)

if anzahlm18 eq 0 then anzahlm18 = 2
vektorm18bunter      = fltarr(anzahlm18/2,1)
vektorm18bfehlerunter = fltarr(anzahlm18/2,1)
summeunterintvektorm18b = fltarr(anzahlm18/2,1)

laufeins = 0

for lauf = 0, anzahlm18-1, 2 do begin
  if vektorm18b(lauf) - (durchmesserunter1/2) lt 1 then goto, dreidreissig13
  if vektorm18b(lauf+1) - (durchmesserunter1/2) lt 1 then goto, dreidreissig13
  if vektorm18b(lauf) + (durchmesserunter1/2) gt dimccd then goto, dreidreissig13
  if vektorm18b(lauf+1) + (durchmesserunter1/2) gt dimccd then goto, dreidreissig13

  kreisbereichring1      = fltarr(dimccd,dimccd)
  kreisbereichring1((vektorm18b(lauf)-(durchmesserunter1/2)):$
(vektorm18b(lauf)+(durchmesserunter1/2)-1),(vektorm18b(lauf+1)-$
(durchmesserunter1/2)):(vektorm18b(lauf+1)+(durchmesserunter1/2)-1))$
=kreis1

  kreisbereichring2      = fltarr(dimccd,dimccd)
  kreisbereichring2((vektorm18b(lauf)-(durchmesserunter2/2)):$
(vektorm18b(lauf)+(durchmesserunter2/2)-1),(vektorm18b(lauf+1)-$
(durchmesserunter2/2)):(vektorm18b(lauf+1)+(durchmesserunter2/2)-1))$
=kreiszwei

;Erzeugung des Rings und Bestimmung der mittleren Intensitaet

kreisringvar          = kreisbereichring1 - kreisbereichring2
positionenuntervar    = where(kreisringvar)
nuntervar             = (size(positionenuntervar))(1)
yuntervar            = positionenuntervar / dimccd
xuntervar            = positionenuntervar mod dimccd

```

```
unterintvektorm18bvar = bild(xuntervar,yuntervar)
summeunterintvektorm18b(laufeins)= total(unterintvektorm18bvar)
vektorm18bunter(laufeins) = summeunterintvektorm18b(laufeins)/nuntervar
vektorm18bfehlerunter(laufeins)= meanabsdev(unterintvektorm18bvar)
ringe = ringe + kreisringvar
dreidreissig13:
laufeins = laufeins+1
;tvsc1, congrid(ringe,512,512)
print, lauf
endfor

tvsc1, congrid(ringe,512,512)
print, 'ENDE RINGE'
endif
```

```
if unterkorr eq 1 then begin
```

```
;Wieder Addition des Verschiebungsvektors, damit die Untergrungreflexe auf den
;Reflexen zum liegen kommen
```

```
vektorr1b = vektorr1b + verschieb
vektorr2b = vektorr2b + verschieb
vektorr3b = vektorr3b + verschieb
vektorr4b = vektorr4b + verschieb
vektorr5b = vektorr5b + verschieb
vektorr6b = vektorr6b + verschieb
vektorr7b = vektorr7b + verschieb
vektorr8b = vektorr8b + verschieb
vektorr9b = vektorr9b + verschieb
vektorr10b = vektorr10b + verschieb
vektorr11b = vektorr11b + verschieb
vektorr12b = vektorr12b + verschieb
vektorr13b = vektorr13b + verschieb
vektorr14b = vektorr14b + verschieb
vektorr15b = vektorr15b + verschieb
vektorr16b = vektorr16b + verschieb
vektorr17b = vektorr17b + verschieb
vektorr18b = vektorr18b + verschieb
```

```
vektorm2b = vektorm2b + verschieb
vektorm3b = vektorm3b + verschieb
vektorm4b = vektorm4b + verschieb
vektorm5b = vektorm5b + verschieb
vektorm6b = vektorm6b + verschieb
vektorm7b = vektorm7b + verschieb
vektorm8b = vektorm8b + verschieb
vektorm9b = vektorm9b + verschieb
vektorm10b = vektorm10b + verschieb
vektorm11b = vektorm11b + verschieb
vektorm12b = vektorm12b + verschieb
vektorm13b = vektorm13b + verschieb
vektorm14b = vektorm14b + verschieb
vektorm15b = vektorm15b + verschieb
vektorm16b = vektorm16b + verschieb
vektorm17b = vektorm17b + verschieb
vektorm18b = vektorm18b + verschieb
```

```
untergkorr = fltarr(dimccd,dimccd)
```

```
laufeins = 0
```

```

for lauf = 0, anzahl1-1, 2 do begin
print, lauf
  if vektorr1b(lauf) - (durchmesserunter1/2) lt 1 then goto, vierdreissig
  if vektorr1b(lauf+1) - (durchmesserunter1/2) lt 1 then goto, vierdreissig
  if vektorr1b(lauf) + (durchmesserunter1/2) gt dimccd then goto, vierdreissig
  if vektorr1b(lauf+1) + (durchmesserunter1/2) gt dimccd then goto, vierdreissig

  untergkorr((vektor1b(lauf)-(durchmesser/2)):$
(vektor1b(lauf)+(durchmesser/2)-1),(vektor1b(lauf+1)-$
(durchmesser/2)):(vektor1b(lauf+1)+(durchmesser/2)-1))$
=kreis*vektor1bunter(laufeins)

  vierdreissig:

  if total(vektor1b) eq -42 then goto, vierdreissiga

  if vektor1b(lauf) - (durchmesserunter1/2) lt 1 then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
  , 'x = ',vektor1b(lauf),' und y = ',vektor1b(lauf+1),$
  ' nicht moeglich!!'
  if vektor1b(lauf+1) - (durchmesserunter1/2) lt 1 then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
  , 'x = ',vektor1b(lauf),' und y = ',vektor1b(lauf+1),$
  ' nicht moeglich!!'
  if vektor1b(lauf) + (durchmesserunter1/2) gt dimccd then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
  , 'x = ',vektor1b(lauf),' und y = ',vektor1b(lauf+1),$
  ' nicht moeglich!!'
  if vektor1b(lauf+1) + (durchmesserunter1/2) gt dimccd then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
  , 'x = ',vektor1b(lauf),' und y = ',vektor1b(lauf+1),$
  ' nicht moeglich!!'

  vierdreissiga:
;tvsc, congrid( (alog(untergkorr-min(untergkorr)+1)),512,512)
laufeins = laufeins+1
endfor

tvsc, congrid( (alog(untergkorr-min(untergkorr)+1)),512,512)

laufeins = 0

```

```

for lauf = 0, anzahl2-1, 2 do begin
;print, lauf
  if vektorr2b(lauf) - (durchmesserunter1/2) lt 1 then goto, fuenfdreissig
  if vektorr2b(lauf+1) - (durchmesserunter1/2) lt 1 then goto, fuenfdreissig
  if vektorr2b(lauf) + (durchmesserunter1/2) gt dimccd then goto, fuenfdreissig
  if vektorr2b(lauf+1) + (durchmesserunter1/2) gt dimccd then goto, fuenfdreissig

  untergkorr((vektor2b(lauf)-(durchmesser/2)):$
(vektor2b(lauf)+(durchmesser/2)-1),(vektor2b(lauf+1)-$
(durchmesser/2)):(vektor2b(lauf+1)+(durchmesser/2)-1))$
=kreis*vektor2bunter(laufeins)

  fuenfdreissig:

  if total(vektor2b) eq -42 then goto, fuenfdreissiga

  if vektor2b(lauf) - (durchmesserunter1/2) lt 1 then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
  , 'x = ',vektor2b(lauf),' und y = ',vektor2b(lauf+1),$
  ' nicht moeglich!!'

```

```
if vektorr2b(lauf+1) - (durchmesserunter1/2) lt 1 then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
  ,x = ',vektorr2b(lauf),' und y = ',vektorr2b(lauf+1),$
  ' nicht moeglich!!'
if vektorr2b(lauf) + (durchmesserunter1/2) gt dimccd then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
  ,x = ',vektorr2b(lauf),' und y = ',vektorr2b(lauf+1),$
  ' nicht moeglich!!'
if vektorr2b(lauf+1) + (durchmesserunter1/2) gt dimccd then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
  ,x = ',vektorr2b(lauf),' und y = ',vektorr2b(lauf+1),$
  ' nicht moeglich!!'

  fuenfdreissiga:
;tvsc1, congrid( (alog(untergkorr-min(untergkorr)+1)),512,512)

laufeins = laufeins+1
endfor

tvsc1, congrid( (alog(untergkorr-min(untergkorr)+1)),512,512)

  laufeins = 0

for lauf = 0, anzahl3-1, 2 do begin
;print, lauf
  if vektorr3b(lauf) - (durchmesserunter1/2) lt 1 then goto, sechsdreissig
  if vektorr3b(lauf+1) - (durchmesserunter1/2) lt 1 then goto, sechsdreissig
  if vektorr3b(lauf) + (durchmesserunter1/2) gt dimccd then goto, sechsdreissig
  if vektorr3b(lauf+1) + (durchmesserunter1/2) gt dimccd then goto, sechsdreissig

  untergkorr((vektorr3b(lauf)-(durchmesser/2)):$
(vektorr3b(lauf)+(durchmesser/2)-1),(vektorr3b(lauf+1)-$
(durchmesser/2)):(vektorr3b(lauf+1)+(durchmesser/2)-1))$
=kreis*vektorr3bunter(laufeins)

  sechsdreissig:

  if total(vektorr3b) eq -42 then goto, sechsdreissiga

  if vektorr3b(lauf) - (durchmesserunter1/2) lt 1 then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
  ,x = ',vektorr3b(lauf),' und y = ',vektorr3b(lauf+1),$
  ' nicht moeglich!!'
  if vektorr3b(lauf+1) - (durchmesserunter1/2) lt 1 then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
  ,x = ',vektorr3b(lauf),' und y = ',vektorr3b(lauf+1),$
  ' nicht moeglich!!'
  if vektorr3b(lauf) + (durchmesserunter1/2) gt dimccd then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
  ,x = ',vektorr3b(lauf),' und y = ',vektorr3b(lauf+1),$
  ' nicht moeglich!!'
  if vektorr3b(lauf+1) + (durchmesserunter1/2) gt dimccd then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
  ,x = ',vektorr3b(lauf),' und y = ',vektorr3b(lauf+1),$
  ' nicht moeglich!!'

  sechsdreissiga:
;tvsc1, congrid( (alog(untergkorr-min(untergkorr)+1)),512,512)
laufeins = laufeins+1
endfor

tvsc1, congrid( (alog(untergkorr-min(untergkorr)+1)),512,512)
```

```

laufeins = 0

for lauf = 0, anzahl4-1, 2 do begin
;print, lauf
  if vektorr4b(lauf) - (durchmesserunter1/2) lt 1 then goto, siebendreissig
  if vektorr4b(lauf+1) - (durchmesserunter1/2) lt 1 then goto, siebendreissig
  if vektorr4b(lauf) + (durchmesserunter1/2) gt dimccd then goto, siebendreissig
  if vektorr4b(lauf+1) + (durchmesserunter1/2) gt dimccd then goto, siebendreissig

  untergkorr((vektor4b(lauf)-(durchmesser/2)):$
(vektor4b(lauf)+(durchmesser/2)-1),(vektor4b(lauf+1)-$
(durchmesser/2)):(vektor4b(lauf+1)+(durchmesser/2)-1))$
=kreis*vektor4bunter(laufeins)

  siebendreissig:

  if total(vektor4b) eq -42 then goto, siebendreissiga

  if vektorr4b(lauf) - (durchmesserunter1/2) lt 1 then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
  , 'x = ',vektor4b(lauf),' und y = ',vektor4b(lauf+1),$
  ' nicht moeglich!!'
  if vektorr4b(lauf+1) - (durchmesserunter1/2) lt 1 then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
  , 'x = ',vektor4b(lauf),' und y = ',vektor4b(lauf+1),$
  ' nicht moeglich!!'
  if vektorr4b(lauf) + (durchmesserunter1/2) gt dimccd then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
  , 'x = ',vektor4b(lauf),' und y = ',vektor4b(lauf+1),$
  ' nicht moeglich!!'
  if vektorr4b(lauf+1) + (durchmesserunter1/2) gt dimccd then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
  , 'x = ',vektor4b(lauf),' und y = ',vektor4b(lauf+1),$
  ' nicht moeglich!!'

  siebendreissiga:
;tvsc, congrid( (alog(untergkorr-min(untergkorr)+1)),512,512)
laufeins = laufeins+1
endfor

```

```

tvsc, congrid( (alog(untergkorr-min(untergkorr)+1)),512,512)

```

```

laufeins = 0

```

```

for lauf = 0, anzahl5-1, 2 do begin
;print, lauf
  if vektorr5b(lauf) - (durchmesserunter1/2) lt 1 then goto, achtdreissig
  if vektorr5b(lauf+1) - (durchmesserunter1/2) lt 1 then goto, achtdreissig
  if vektorr5b(lauf) + (durchmesserunter1/2) gt dimccd then goto, achtdreissig
  if vektorr5b(lauf+1) + (durchmesserunter1/2) gt dimccd then goto, achtdreissig

  untergkorr((vektor5b(lauf)-(durchmesser/2)):$
(vektor5b(lauf)+(durchmesser/2)-1),(vektor5b(lauf+1)-$
(durchmesser/2)):(vektor5b(lauf+1)+(durchmesser/2)-1))$
=kreis*vektor5bunter(laufeins)

  achtdreissig:

  if total(vektor5b) eq -42 then goto, achtdreissiga

  if vektorr5b(lauf) - (durchmesserunter1/2) lt 1 then print,$

```

```

'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
  ,x = ',vektorr5b(lauf),' und y = ',vektorr5b(lauf+1),$
  ' nicht moeglich!!'
if vektorr5b(lauf+1) - (durchmesserunter1/2) lt 1 then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
  ,x = ',vektorr5b(lauf),' und y = ',vektorr5b(lauf+1),$
  ' nicht moeglich!!'
if vektorr5b(lauf) + (durchmesserunter1/2) gt dimccd then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
  ,x = ',vektorr5b(lauf),' und y = ',vektorr5b(lauf+1),$
  ' nicht moeglich!!'
if vektorr5b(lauf+1) + (durchmesserunter1/2) gt dimccd then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
  ,x = ',vektorr5b(lauf),' und y = ',vektorr5b(lauf+1),$
  ' nicht moeglich!!'

      achtdreissiga:
;tvsc1, congrid( (alog(untergkorr-min(untergkorr)+1)),512,512)
laufeins = laufeins+1
endfor

tvsc1, congrid( (alog(untergkorr-min(untergkorr)+1)),512,512)

      laufeins = 0

for lauf = 0, anzahl6-1, 2 do begin
;print, lauf
  if vektorr6b(lauf) - (durchmesserunter1/2) lt 1 then goto, neundreissig
  if vektorr6b(lauf+1) - (durchmesserunter1/2) lt 1 then goto, neundreissig
  if vektorr6b(lauf) + (durchmesserunter1/2) gt dimccd then goto, neundreissig
  if vektorr6b(lauf+1) + (durchmesserunter1/2) gt dimccd then goto, neundreissig

  untergkorr((vektorr6b(lauf)-(durchmesser/2)):$
(vektorr6b(lauf)+(durchmesser/2)-1),(vektorr6b(lauf+1)-$
(durchmesser/2)):(vektorr6b(lauf+1)+(durchmesser/2)-1))$
=kreis*vektorr6bunter(laufeins)

  neundreissig:

  if total(vektorr6b) eq -42 then goto, neundreissiga

  if vektorr6b(lauf) - (durchmesserunter1/2) lt 1 then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
  ,x = ',vektorr6b(lauf),' und y = ',vektorr6b(lauf+1),$
  ' nicht moeglich!!'
  if vektorr6b(lauf+1) - (durchmesserunter1/2) lt 1 then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
  ,x = ',vektorr6b(lauf),' und y = ',vektorr6b(lauf+1),$
  ' nicht moeglich!!'
  if vektorr6b(lauf) + (durchmesserunter1/2) gt dimccd then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
  ,x = ',vektorr6b(lauf),' und y = ',vektorr6b(lauf+1),$
  ' nicht moeglich!!'
  if vektorr6b(lauf+1) + (durchmesserunter1/2) gt dimccd then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
  ,x = ',vektorr6b(lauf),' und y = ',vektorr6b(lauf+1),$
  ' nicht moeglich!!'

      neundreissiga:
;tvsc1, congrid( (alog(untergkorr-min(untergkorr)+1)),512,512)
laufeins = laufeins+1
endfor

```

```

tvsc1, congrid( (alog(untergkorr-min(untergkorr)+1)),512,512)

  laufeins = 0

for lauf = 0, anzahl7-1, 2 do begin
;print, lauf
  if vektorr7b(lauf) - (durchmesserunter1/2) lt 1 then goto, vierzig
  if vektorr7b(lauf+1) - (durchmesserunter1/2) lt 1 then goto, vierzig
  if vektorr7b(lauf) + (durchmesserunter1/2) gt dimccd then goto, vierzig
  if vektorr7b(lauf+1) + (durchmesserunter1/2) gt dimccd then goto, vierzig

  untergkorr((vektor7b(lauf)-(durchmesser/2)):$
(vektor7b(lauf)+(durchmesser/2)-1),(vektor7b(lauf+1)-$
(durchmesser/2)):(vektor7b(lauf+1)+(durchmesser/2)-1))$
=kreis*vektor7bunter(laufeins)

  vierzig:

  if total(vektor7b) eq -42 then goto, vierziga

  if vektor7b(lauf) - (durchmesserunter1/2) lt 1 then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
  , 'x = ',vektor7b(lauf),' und y = ',vektor7b(lauf+1),$
  ' nicht moeglich!!'
  if vektor7b(lauf+1) - (durchmesserunter1/2) lt 1 then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
  , 'x = ',vektor7b(lauf),' und y = ',vektor7b(lauf+1),$
  ' nicht moeglich!!'
  if vektor7b(lauf) + (durchmesserunter1/2) gt dimccd then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
  , 'x = ',vektor7b(lauf),' und y = ',vektor7b(lauf+1),$
  ' nicht moeglich!!'
  if vektor7b(lauf+1) + (durchmesserunter1/2) gt dimccd then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
  , 'x = ',vektor7b(lauf),' und y = ',vektor7b(lauf+1),$
  ' nicht moeglich!!'

  vierziga:
;tvsc1, congrid( (alog(untergkorr-min(untergkorr)+1)),512,512)
  laufeins = laufeins+1
endfor

```

```

tvsc1, congrid((alog(untergkorr-min(untergkorr)+1)),512,512)

laufeins=0

for lauf = 0, anzahl8-1, 2 do begin
;print, lauf
  if vektorr8b(lauf) - (durchmesserunter1/2) lt 1 then goto, vierzig1
  if vektorr8b(lauf+1) - (durchmesserunter1/2) lt 1 then goto, vierzig1
  if vektorr8b(lauf) + (durchmesserunter1/2) gt dimccd then goto, vierzig1
  if vektorr8b(lauf+1) + (durchmesserunter1/2) gt dimccd then goto, vierzig1

  untergkorr((vektor8b(lauf)-(durchmesser/2)):$
(vektor8b(lauf)+(durchmesser/2)-1),(vektor8b(lauf+1)-$
(durchmesser/2)):(vektor8b(lauf+1)+(durchmesser/2)-1))$
=kreis*vektor8bunter(laufeins)

  vierzig1:

  if total(vektor8b) eq -42 then goto, vierziga1

```

```

if vektorr8b(lauf) - (durchmesserunter1/2) lt 1 then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
  ,x = ',vektorr8b(lauf),' und y = ',vektorr8b(lauf+1),$
  ' nicht moeglich!!'
if vektorr8b(lauf+1) - (durchmesserunter1/2) lt 1 then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
  ,x = ',vektorr8b(lauf),' und y = ',vektorr8b(lauf+1),$
  ' nicht moeglich!!'
if vektorr8b(lauf) + (durchmesserunter1/2) gt dimccd then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
  ,x = ',vektorr8b(lauf),' und y = ',vektorr8b(lauf+1),$
  ' nicht moeglich!!'
if vektorr8b(lauf+1) + (durchmesserunter1/2) gt dimccd then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
  ,x = ',vektorr8b(lauf),' und y = ',vektorr8b(lauf+1),$
  ' nicht moeglich!!'

vierziga1:
;tvsc1, congrid( (alog(untergkorr-min(untergkorr)+1)),512,512)
laufeins = laufeins+1
endfor

tvsc1, congrid((alog(untergkorr-min(untergkorr)+1)),512,512)

laufeins=0

for lauf = 0, anzahl9-1, 2 do begin
;print, lauf
if vektorr9b(lauf) - (durchmesserunter1/2) lt 1 then goto, vierzig2
if vektorr9b(lauf+1) - (durchmesserunter1/2) lt 1 then goto, vierzig2
if vektorr9b(lauf) + (durchmesserunter1/2) gt dimccd then goto, vierzig2
if vektorr9b(lauf+1) + (durchmesserunter1/2) gt dimccd then goto, vierzig2

untergkorr((vektorr9b(lauf)-(durchmesser/2)):$
(vektorr9b(lauf)+(durchmesser/2)-1),(vektorr9b(lauf+1)-$
(durchmesser/2)):(vektorr9b(lauf+1)+(durchmesser/2)-1))$
=kreis*vektorr9bunter(laufeins)

vierzig2:

if total(vektorr9b) eq -42 then goto, vierziga2

if vektorr9b(lauf) - (durchmesserunter1/2) lt 1 then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
  ,x = ',vektorr9b(lauf),' und y = ',vektorr9b(lauf+1),$
  ' nicht moeglich!!'
if vektorr9b(lauf+1) - (durchmesserunter1/2) lt 1 then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
  ,x = ',vektorr9b(lauf),' und y = ',vektorr9b(lauf+1),$
  ' nicht moeglich!!'
if vektorr9b(lauf) + (durchmesserunter1/2) gt dimccd then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
  ,x = ',vektorr9b(lauf),' und y = ',vektorr9b(lauf+1),$
  ' nicht moeglich!!'
if vektorr9b(lauf+1) + (durchmesserunter1/2) gt dimccd then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
  ,x = ',vektorr9b(lauf),' und y = ',vektorr9b(lauf+1),$
  ' nicht moeglich!!'

vierziga2:
;tvsc1, congrid( (alog(untergkorr-min(untergkorr)+1)),512,512)

```

```

laufeins = laufeins+1
endfor

```

```

tvsc1, congrid((alog(untergkorr-min(untergkorr)+1)),512,512)

```

```

laufeins=0

```

```

for lauf = 0, anzahl10-1, 2 do begin

```

```

;print, lauf

```

```

if vektorr10b(lauf) - (durchmesserunter1/2) lt 1 then goto, vierzig3
if vektorr10b(lauf+1) - (durchmesserunter1/2) lt 1 then goto, vierzig3
if vektorr10b(lauf) + (durchmesserunter1/2) gt dimccd then goto, vierzig3
if vektorr10b(lauf+1) + (durchmesserunter1/2) gt dimccd then goto, vierzig3

```

```

untergkorr((vektorr10b(lauf)-(durchmesser/2)):$
(vektorr10b(lauf)+(durchmesser/2)-1),(vektorr10b(lauf+1)-$
(durchmesser/2)):(vektorr10b(lauf+1)+(durchmesser/2)-1))$
=kreis*vektorr10bunter(laufeins)

```

```

vierzig3:

```

```

if total(vektorr10b) eq -42 then goto, vierziga3

```

```

if vektorr10b(lauf) - (durchmesserunter1/2) lt 1 then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
,x = ',vektorr10b(lauf),' und y = ',vektorr10b(lauf+1),$
' nicht moeglich!!'

```

```

if vektorr10b(lauf+1) - (durchmesserunter1/2) lt 1 then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
,x = ',vektorr10b(lauf),' und y = ',vektorr10b(lauf+1),$
' nicht moeglich!!'

```

```

if vektorr10b(lauf) + (durchmesserunter1/2) gt dimccd then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
,x = ',vektorr10b(lauf),' und y = ',vektorr10b(lauf+1),$
' nicht moeglich!!'

```

```

if vektorr10b(lauf+1) + (durchmesserunter1/2) gt dimccd then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
,x = ',vektorr10b(lauf),' und y = ',vektorr10b(lauf+1),$
' nicht moeglich!!'

```

```

vierziga3:

```

```

;tvsc1, congrid( (alog(untergkorr-min(untergkorr)+1)),512,512)

```

```

laufeins = laufeins+1

```

```

endfor

```

```

tvsc1, congrid((alog(untergkorr-min(untergkorr)+1)),512,512)

```

```

laufeins=0

```

```

for lauf = 0, anzahl11-1, 2 do begin

```

```

;print, lauf

```

```

if vektorr11b(lauf) - (durchmesserunter1/2) lt 1 then goto, vierzig4
if vektorr11b(lauf+1) - (durchmesserunter1/2) lt 1 then goto, vierzig4
if vektorr11b(lauf) + (durchmesserunter1/2) gt dimccd then goto, vierzig4
if vektorr11b(lauf+1) + (durchmesserunter1/2) gt dimccd then goto, vierzig4

```

```

untergkorr((vektorr11b(lauf)-(durchmesser/2)):$
(vektorr11b(lauf)+(durchmesser/2)-1),(vektorr11b(lauf+1)-$
(durchmesser/2)):(vektorr11b(lauf+1)+(durchmesser/2)-1))$
=kreis*vektorr11bunter(laufeins)

```

```

vierzig4:

```

```

if total(vektorr11b) eq -42 then goto, vierziga4

if vektorr11b(lauf) - (durchmesserunter1/2) lt 1 then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
  ,x = ',vektorr11b(lauf),' und y = ',vektorr11b(lauf+1),$
  ' nicht moeglich!!'
if vektorr11b(lauf+1) - (durchmesserunter1/2) lt 1 then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
  ,x = ',vektorr11b(lauf),' und y = ',vektorr11b(lauf+1),$
  ' nicht moeglich!!'
if vektorr11b(lauf) + (durchmesserunter1/2) gt dimccd then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
  ,x = ',vektorr11b(lauf),' und y = ',vektorr11b(lauf+1),$
  ' nicht moeglich!!'
if vektorr11b(lauf+1) + (durchmesserunter1/2) gt dimccd then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
  ,x = ',vektorr11b(lauf),' und y = ',vektorr11b(lauf+1),$
  ' nicht moeglich!!'

vierziga4:
;tvsc1, congrid( (alog(untergkorr-min(untergkorr)+1)),512,512)
laufeins = laufeins+1
endfor

tvsc1, congrid((alog(untergkorr-min(untergkorr)+1)),512,512)

laufeins=0

for lauf = 0, anzahl12-1, 2 do begin
;print, lauf
if vektorr12b(lauf) - (durchmesserunter1/2) lt 1 then goto, vierzig5
if vektorr12b(lauf+1) - (durchmesserunter1/2) lt 1 then goto, vierzig5
if vektorr12b(lauf) + (durchmesserunter1/2) gt dimccd then goto, vierzig5
if vektorr12b(lauf+1) + (durchmesserunter1/2) gt dimccd then goto, vierzig5

untergkorr((vektorr12b(lauf)-(durchmesser/2)):$
(vektorr12b(lauf)+(durchmesser/2)-1),(vektorr12b(lauf+1)-$
(durchmesser/2)):(vektorr12b(lauf+1)+(durchmesser/2)-1))$
=kreis*vektorr12bunter(laufeins)

vierzig5:

if total(vektorr12b) eq -42 then goto, vierziga5

if vektorr12b(lauf) - (durchmesserunter1/2) lt 1 then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
  ,x = ',vektorr12b(lauf),' und y = ',vektorr12b(lauf+1),$
  ' nicht moeglich!!'
if vektorr12b(lauf+1) - (durchmesserunter1/2) lt 1 then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
  ,x = ',vektorr12b(lauf),' und y = ',vektorr12b(lauf+1),$
  ' nicht moeglich!!'
if vektorr12b(lauf) + (durchmesserunter1/2) gt dimccd then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
  ,x = ',vektorr12b(lauf),' und y = ',vektorr12b(lauf+1),$
  ' nicht moeglich!!'
if vektorr12b(lauf+1) + (durchmesserunter1/2) gt dimccd then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
  ,x = ',vektorr12b(lauf),' und y = ',vektorr12b(lauf+1),$
  ' nicht moeglich!!'

```

```

vierziga5:
;tvsc1, congrid( (alog(untergkorr-min(untergkorr)+1)),512,512)
laufeins = laufeins+1
endfor

```

```
tvsc1, congrid((alog(untergkorr-min(untergkorr)+1)),512,512)
```

```
laufeins=0
```

```

for lauf = 0, anzahl13-1, 2 do begin
;print, lauf
if vektorr13b(lauf) - (durchmesserunter1/2) lt 1 then goto, vierzig6
if vektorr13b(lauf+1) - (durchmesserunter1/2) lt 1 then goto, vierzig6
if vektorr13b(lauf) + (durchmesserunter1/2) gt dimccd then goto, vierzig6
if vektorr13b(lauf+1) + (durchmesserunter1/2) gt dimccd then goto, vierzig6

untergkorr((vektor13b(lauf)-(durchmesser/2)):$
(vektor13b(lauf)+(durchmesser/2)-1),(vektor13b(lauf+1)-$
(durchmesser/2)):(vektor13b(lauf+1)+(durchmesser/2)-1))$
=kreis*vektor13bunter(laufeins)

```

```
vierzig6:
```

```
if total(vektor13b) eq -42 then goto, vierziga6
```

```

if vektorr13b(lauf) - (durchmesserunter1/2) lt 1 then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
, 'x = ',vektor13b(lauf),' und y = ',vektor13b(lauf+1),$
' nicht moeglich!!'
if vektorr13b(lauf+1) - (durchmesserunter1/2) lt 1 then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
, 'x = ',vektor13b(lauf),' und y = ',vektor13b(lauf+1),$
' nicht moeglich!!'
if vektorr13b(lauf) + (durchmesserunter1/2) gt dimccd then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
, 'x = ',vektor13b(lauf),' und y = ',vektor13b(lauf+1),$
' nicht moeglich!!'
if vektorr13b(lauf+1) + (durchmesserunter1/2) gt dimccd then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
, 'x = ',vektor13b(lauf),' und y = ',vektor13b(lauf+1),$
' nicht moeglich!!'

```

```
vierziga6:
```

```

;tvsc1, congrid( (alog(untergkorr-min(untergkorr)+1)),512,512)
laufeins = laufeins+1
endfor

```

```
tvsc1, congrid((alog(untergkorr-min(untergkorr)+1)),512,512)
```

```
laufeins=0
```

```

for lauf = 0, anzahl14-1, 2 do begin
;print, lauf
if vektorr14b(lauf) - (durchmesserunter1/2) lt 1 then goto, vierzig7
if vektorr14b(lauf+1) - (durchmesserunter1/2) lt 1 then goto, vierzig7
if vektorr14b(lauf) + (durchmesserunter1/2) gt dimccd then goto, vierzig7
if vektorr14b(lauf+1) + (durchmesserunter1/2) gt dimccd then goto, vierzig7

untergkorr((vektor14b(lauf)-(durchmesser/2)):$
(vektor14b(lauf)+(durchmesser/2)-1),(vektor14b(lauf+1)-$
(durchmesser/2)):(vektor14b(lauf+1)+(durchmesser/2)-1))$

```

```
=kreis*vektorr14bunter(laufeins)
```

```
vierzig7:
```

```
if total(vektorr14b) eq -42 then goto, vierziga7
```

```
if vektorr14b(lauf) - (durchmesserunter1/2) lt 1 then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
  ,x = ',vektorr14b(lauf),' und y = ',vektorr14b(lauf+1),$
  ' nicht moeglich!!'
```

```
if vektorr14b(lauf+1) - (durchmesserunter1/2) lt 1 then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
  ,x = ',vektorr14b(lauf),' und y = ',vektorr14b(lauf+1),$
  ' nicht moeglich!!'
```

```
if vektorr14b(lauf) + (durchmesserunter1/2) gt dimccd then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
  ,x = ',vektorr14b(lauf),' und y = ',vektorr14b(lauf+1),$
  ' nicht moeglich!!'
```

```
if vektorr14b(lauf+1) + (durchmesserunter1/2) gt dimccd then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
  ,x = ',vektorr14b(lauf),' und y = ',vektorr14b(lauf+1),$
  ' nicht moeglich!!'
```

```
vierziga7:
```

```
;tvscf, congrid( (alog(untergkorr-min(untergkorr)+1)),512,512)
```

```
laufeins = laufeins+1
```

```
endfor
```

```
tvscf, congrid((alog(untergkorr-min(untergkorr)+1)),512,512)
```

```
laufeins=0
```

```
for lauf = 0, anzahl15-1, 2 do begin
```

```
;print, lauf
```

```
if vektorr15b(lauf) - (durchmesserunter1/2) lt 1 then goto, vierzig8
```

```
if vektorr15b(lauf+1) - (durchmesserunter1/2) lt 1 then goto, vierzig8
```

```
if vektorr15b(lauf) + (durchmesserunter1/2) gt dimccd then goto, vierzig8
```

```
if vektorr15b(lauf+1) + (durchmesserunter1/2) gt dimccd then goto, vierzig8
```

```
untergkorr((vektorr15b(lauf)-(durchmesser/2)):$
(vektorr15b(lauf)+(durchmesser/2)-1),(vektorr15b(lauf+1)-$
(durchmesser/2)):(vektorr15b(lauf+1)+(durchmesser/2)-1))$
=kreis*vektorr15bunter(laufeins)
```

```
vierzig8:
```

```
if total(vektorr15b) eq -42 then goto, vierziga8
```

```
if vektorr15b(lauf) - (durchmesserunter1/2) lt 1 then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
  ,x = ',vektorr15b(lauf),' und y = ',vektorr15b(lauf+1),$
  ' nicht moeglich!!'
```

```
if vektorr15b(lauf+1) - (durchmesserunter1/2) lt 1 then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
  ,x = ',vektorr15b(lauf),' und y = ',vektorr15b(lauf+1),$
  ' nicht moeglich!!'
```

```
if vektorr15b(lauf) + (durchmesserunter1/2) gt dimccd then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
  ,x = ',vektorr15b(lauf),' und y = ',vektorr15b(lauf+1),$
  ' nicht moeglich!!'
```

```
if vektorr15b(lauf+1) + (durchmesserunter1/2) gt dimccd then print,$
```

```

'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
  ,x = ',vektorr15b(lauf),' und y = ',vektorr15b(lauf+1),$
  ' nicht moeglich!!'

vierziga8:
;tvsc1, congrid( (alog(untergkorr-min(untergkorr)+1)),512,512)
laufeins = laufeins+1
endfor

tvsc1, congrid((alog(untergkorr-min(untergkorr)+1)),512,512)

laufeins=0

for lauf = 0, anzahl16-1, 2 do begin
;print, lauf
  if vektorr16b(lauf) - (durchmesserunter1/2) lt 1 then goto, vierzig9
  if vektorr16b(lauf+1) - (durchmesserunter1/2) lt 1 then goto, vierzig9
  if vektorr16b(lauf) + (durchmesserunter1/2) gt dimccd then goto, vierzig9
  if vektorr16b(lauf+1) + (durchmesserunter1/2) gt dimccd then goto, vierzig9

  untergkorr((vektorr16b(lauf)-(durchmesser/2)):$
(vektorr16b(lauf)+(durchmesser/2)-1),(vektorr16b(lauf+1)-$
(durchmesser/2)):(vektorr16b(lauf+1)+(durchmesser/2)-1))$
=kreis*vektorr16bunter(laufeins)

vierzig9:

  if total(vektorr16b) eq -42 then goto, vierziga9

  if vektorr16b(lauf) - (durchmesserunter1/2) lt 1 then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
  ,x = ',vektorr16b(lauf),' und y = ',vektorr16b(lauf+1),$
  ' nicht moeglich!!'
  if vektorr16b(lauf+1) - (durchmesserunter1/2) lt 1 then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
  ,x = ',vektorr16b(lauf),' und y = ',vektorr16b(lauf+1),$
  ' nicht moeglich!!'
  if vektorr16b(lauf) + (durchmesserunter1/2) gt dimccd then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
  ,x = ',vektorr16b(lauf),' und y = ',vektorr16b(lauf+1),$
  ' nicht moeglich!!'
  if vektorr16b(lauf+1) + (durchmesserunter1/2) gt dimccd then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
  ,x = ',vektorr16b(lauf),' und y = ',vektorr16b(lauf+1),$
  ' nicht moeglich!!'

vierziga9:
;tvsc1, congrid( (alog(untergkorr-min(untergkorr)+1)),512,512)
laufeins = laufeins+1
endfor

tvsc1, congrid((alog(untergkorr-min(untergkorr)+1)),512,512)

laufeins=0

for lauf = 0, anzahl17-1, 2 do begin
;print, lauf
  if vektorr17b(lauf) - (durchmesserunter1/2) lt 1 then goto, vierzig10
  if vektorr17b(lauf+1) - (durchmesserunter1/2) lt 1 then goto, vierzig10
  if vektorr17b(lauf) + (durchmesserunter1/2) gt dimccd then goto, vierzig10
  if vektorr17b(lauf+1) + (durchmesserunter1/2) gt dimccd then goto, vierzig10

```

```

untergkorr((vektorr17b(lauf)-(durchmesser/2)):$
(vektorr17b(lauf)+(durchmesser/2)-1),(vektorr17b(lauf+1)-$
(durchmesser/2)):(vektorr17b(lauf+1)+(durchmesser/2)-1))$
=kreis*vektorr17bunter(laufeins)

```

vierzig10:

```
if total(vektorr17b) eq -42 then goto, vierziga10
```

```
if vektorr17b(lauf) - (durchmesserunter1/2) lt 1 then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
, 'x = ',vektorr17b(lauf),' und y = ',vektorr17b(lauf+1),$
' nicht moeglich!!'

```

```
if vektorr17b(lauf+1) - (durchmesserunter1/2) lt 1 then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
, 'x = ',vektorr17b(lauf),' und y = ',vektorr17b(lauf+1),$
' nicht moeglich!!'

```

```
if vektorr17b(lauf) + (durchmesserunter1/2) gt dimccd then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
, 'x = ',vektorr17b(lauf),' und y = ',vektorr17b(lauf+1),$
' nicht moeglich!!'

```

```
if vektorr17b(lauf+1) + (durchmesserunter1/2) gt dimccd then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
, 'x = ',vektorr17b(lauf),' und y = ',vektorr17b(lauf+1),$
' nicht moeglich!!'

```

vierziga10:

```

;tvsc1, congrid( (alog(untergkorr-min(untergkorr)+1)),512,512)
laufeins = laufeins+1
endfor

```

```
tvsc1, congrid((alog(untergkorr-min(untergkorr)+1)),512,512)
```

laufeins=0

```
for lauf = 0, anzahl18-1, 2 do begin
```

```
 ;print, lauf
```

```

if vektorr18b(lauf) - (durchmesserunter1/2) lt 1 then goto, vierzig11
if vektorr18b(lauf+1) - (durchmesserunter1/2) lt 1 then goto, vierzig11
if vektorr18b(lauf) + (durchmesserunter1/2) gt dimccd then goto, vierzig11
if vektorr18b(lauf+1) + (durchmesserunter1/2) gt dimccd then goto, vierzig11

```

```

untergkorr((vektorr18b(lauf)-(durchmesser/2)):$
(vektorr18b(lauf)+(durchmesser/2)-1),(vektorr18b(lauf+1)-$
(durchmesser/2)):(vektorr18b(lauf+1)+(durchmesser/2)-1))$
=kreis*vektorr18bunter(laufeins)

```

vierzig11:

```
if total(vektorr18b) eq -42 then goto, vierziga11
```

```
if vektorr18b(lauf) - (durchmesserunter1/2) lt 1 then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
, 'x = ',vektorr18b(lauf),' und y = ',vektorr18b(lauf+1),$
' nicht moeglich!!'

```

```
if vektorr18b(lauf+1) - (durchmesserunter1/2) lt 1 then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
, 'x = ',vektorr18b(lauf),' und y = ',vektorr18b(lauf+1),$
' nicht moeglich!!'

```

```
if vektorr18b(lauf) + (durchmesserunter1/2) gt dimccd then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
, 'x = ',vektorr18b(lauf),' und y = ',vektorr18b(lauf+1),$

```

```

' nicht moeglich!!'
if vektorr18b(lauf+1) + (durchmesserunter1/2) gt dimccd then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
  ,x = ',vektor18b(lauf),' und y = ',vektor18b(lauf+1),$
' nicht moeglich!!'

vierziga11:
;tvsc1, congrid( (alog(untergkorr-min(untergkorr)+1)),512,512)
laufeins = laufeins+1
endfor

tvsc1, congrid((alog(untergkorr-min(untergkorr)+1)),512,512)

laufeins = 0

for lauf = 0, anzahlm2-1, 2 do begin
;print, lauf
if vektorrm2b(lauf) - (durchmesserunter1/2) lt 1 then goto, einvierzig
if vektorrm2b(lauf+1) - (durchmesserunter1/2) lt 1 then goto, einvierzig
if vektorrm2b(lauf) + (durchmesserunter1/2) gt dimccd then goto, einvierzig
if vektorrm2b(lauf+1) + (durchmesserunter1/2) gt dimccd then goto, einvierzig

untergkorr((vektorrm2b(lauf)-(durchmesser/2)):$
(vektorrm2b(lauf)+(durchmesser/2)-1),(vektorrm2b(lauf+1)-$
(durchmesser/2)):(vektorrm2b(lauf+1)+(durchmesser/2)-1))$
=kreis*vektorrm2bunter(laufeins)

einvierzig:

if total(vektorrm2b) eq -42 then goto, einvierziga

if vektorrm2b(lauf) - (durchmesserunter1/2) lt 1 then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
  ,x = ',vektorrm2b(lauf),' und y = ',vektorrm2b(lauf+1),$
' nicht moeglich!!'
if vektorrm2b(lauf+1) - (durchmesserunter1/2) lt 1 then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
  ,x = ',vektorrm2b(lauf),' und y = ',vektorrm2b(lauf+1),$
' nicht moeglich!!'
if vektorrm2b(lauf) + (durchmesserunter1/2) gt dimccd then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
  ,x = ',vektorrm2b(lauf),' und y = ',vektorrm2b(lauf+1),$
' nicht moeglich!!'

if vektorrm2b(lauf+1) + (durchmesserunter1/2) gt dimccd then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
  ,x = ',vektorrm2b(lauf),' und y = ',vektorrm2b(lauf+1),$
' nicht moeglich!!'

einvierziga:

;tvsc1, congrid( (alog(untergkorr-min(untergkorr)+1)),512,512)
laufeins = laufeins+1
endfor

tvsc1, congrid((alog(untergkorr-min(untergkorr)+1)),512,512)

laufeins = 0

for lauf = 0, anzahlm3-1, 2 do begin
;print, lauf, laufeins
if vektorrm3b(lauf) - (durchmesserunter1/2) lt 1 then goto, zweivierzig

```

```
if vektorrm3b(lauf+1) - (durchmesserunter1/2) lt 1 then goto, zweivierzig
if vektorrm3b(lauf) + (durchmesserunter1/2) gt dimccd then goto, zweivierzig
if vektorrm3b(lauf+1) + (durchmesserunter1/2) gt dimccd then goto, zweivierzig
```

```
untergkorr((vektorrm3b(lauf)-(durchmesser/2)):$
(vektorrm3b(lauf)+(durchmesser/2)-1),(vektorrm3b(lauf+1)-$
(durchmesser/2)):(vektorrm3b(lauf+1)+(durchmesser/2)-1))$
=kreis*vektorrm3bunter(laufeins)
```

zweivierzig:

```
if total(vektorrm3b) eq -42 then goto, zweivierziga
```

```
if vektorrm3b(lauf) - (durchmesserunter1/2) lt 1 then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
, 'x = ',vektorrm3b(lauf),' und y = ',vektorrm3b(lauf+1),$
' nicht moeglich!!'
```

```
if vektorrm3b(lauf+1) - (durchmesserunter1/2) lt 1 then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
, 'x = ',vektorrm3b(lauf),' und y = ',vektorrm3b(lauf+1),$
' nicht moeglich!!'
```

```
if vektorrm3b(lauf) + (durchmesserunter1/2) gt dimccd then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
, 'x = ',vektorrm3b(lauf),' und y = ',vektorrm3b(lauf+1),$
' nicht moeglich!!'
```

```
if vektorrm3b(lauf+1) + (durchmesserunter1/2) gt dimccd then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
, 'x = ',vektorrm3b(lauf),' und y = ',vektorrm3b(lauf+1),$
' nicht moeglich!!'
```

zweivierziga:

```
;tvsc1, congrid( (alog(untergkorr-min(untergkorr)+1)),512,512)
laufeins = laufeins+1
endfor
```

```
tvsc1, congrid((alog(untergkorr-min(untergkorr)+1)),512,512)
```

laufeins = 0

```
for lauf = 0, anzahlm4-1, 2 do begin
```

```
;print, lauf
```

```
if vektorrm4b(lauf) - (durchmesserunter1/2) lt 1 then goto, dreivierzig
if vektorrm4b(lauf+1) - (durchmesserunter1/2) lt 1 then goto, dreivierzig
if vektorrm4b(lauf) + (durchmesserunter1/2) gt dimccd then goto, dreivierzig
if vektorrm4b(lauf+1) + (durchmesserunter1/2) gt dimccd then goto, dreivierzig
```

```
untergkorr((vektorrm4b(lauf)-(durchmesser/2)):$
(vektorrm4b(lauf)+(durchmesser/2)-1),(vektorrm4b(lauf+1)-$
(durchmesser/2)):(vektorrm4b(lauf+1)+(durchmesser/2)-1))$
=kreis*vektorrm4bunter(laufeins)
```

dreivierzig:

```
if total(vektorrm4b) eq -42 then goto, dreivierziga
```

```
if vektorrm4b(lauf) - (durchmesserunter1/2) lt 1 then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
, 'x = ',vektorrm4b(lauf),' und y = ',vektorrm4b(lauf+1),$
' nicht moeglich!!'
```

```
if vektorrm4b(lauf+1) - (durchmesserunter1/2) lt 1 then print,$
```

```

'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
  ,x = ',vektorm4b(lauf),' und y = ',vektorm4b(lauf+1),$
  ' nicht moeglich!!'
if vektorm4b(lauf) + (durchmesserunter1/2) gt dimccd then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
  ,x = ',vektorm4b(lauf),' und y = ',vektorm4b(lauf+1),$
  ' nicht moeglich!!'

if vektorm4b(lauf+1) + (durchmesserunter1/2) gt dimccd then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
  ,x = ',vektorm4b(lauf),' und y = ',vektorm4b(lauf+1),$
  ' nicht moeglich!!'

dreivierziga:
;tvsc1, congrid( (alog(untergkorr-min(untergkorr)+1)),512,512)
laufeins = laufeins+1
endfor

tvsc1, congrid((alog(untergkorr-min(untergkorr)+1)),512,512)

laufeins = 0

for lauf = 0, anzahlm5-1, 2 do begin

if vektorm5b(lauf) - (durchmesserunter1/2) lt 1 then goto, viervierzig
if vektorm5b(lauf+1) - (durchmesserunter1/2) lt 1 then goto, viervierzig
if vektorm5b(lauf) + (durchmesserunter1/2) gt dimccd then goto, viervierzig
if vektorm5b(lauf+1) + (durchmesserunter1/2) gt dimccd then goto, viervierzig

untergkorr((vektorm5b(lauf)-(durchmesser/2)):$
(vektorm5b(lauf)+(durchmesser/2)-1),(vektorm5b(lauf+1)-$
(durchmesser/2)):(vektorm5b(lauf+1)+(durchmesser/2)-1))$
=kreis*vektorm5bunter(laufeins)

viervierzig:

if total(vektorm5b) eq -42 then goto, viervierziga

if vektorm5b(lauf) - (durchmesserunter1/2) lt 1 then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
  ,x = ',vektorm5b(lauf),' und y = ',vektorm5b(lauf+1),$
  ' nicht moeglich!!'
if vektorm5b(lauf+1) - (durchmesserunter1/2) lt 1 then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
  ,x = ',vektorm5b(lauf),' und y = ',vektorm5b(lauf+1),$
  ' nicht moeglich!!'
if vektorm5b(lauf) + (durchmesserunter1/2) gt dimccd then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
  ,x = ',vektorm5b(lauf),' und y = ',vektorm5b(lauf+1),$
  ' nicht moeglich!!'

if vektorm5b(lauf+1) + (durchmesserunter1/2) gt dimccd then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
  ,x = ',vektorm5b(lauf),' und y = ',vektorm5b(lauf+1),$
  ' nicht moeglich!!'

viervierziga:
;tvsc1, congrid( (alog(untergkorr-min(untergkorr)+1)),512,512)
laufeins = laufeins+1
endfor

tvsc1, congrid((alog(untergkorr-min(untergkorr)+1)),512,512)

```

```

laufeins = 0

for lauf = 0, anzahlm6-1, 2 do begin

    if vektorrm6b(lauf) - (durchmesserunter1/2) lt 1 then goto, viervierzig1
    if vektorrm6b(lauf+1) - (durchmesserunter1/2) lt 1 then goto, viervierzig1
    if vektorrm6b(lauf) + (durchmesserunter1/2) gt dimccd then goto, viervierzig1
    if vektorrm6b(lauf+1) + (durchmesserunter1/2) gt dimccd then goto, viervierzig1

    untergkorr((vektorrm6b(lauf)-(durchmesser/2)):$
    (vektorrm6b(lauf)+(durchmesser/2)-1),(vektorrm6b(lauf+1)-$
    (durchmesser/2)):(vektorrm6b(lauf+1)+(durchmesser/2)-1))$
    =kreis*vektorrm6bunter(laufeins)

    viervierzig1:

    if total(vektorrm6b) eq -42 then goto, viervierziga1

    if vektorrm6b(lauf) - (durchmesserunter1/2) lt 1 then print,$
    'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
    , 'x = ',vektorrm6b(lauf),' und y = ',vektorrm6b(lauf+1),$
    ' nicht moeglich!!'
    if vektorrm6b(lauf+1) - (durchmesserunter1/2) lt 1 then print,$
    'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
    , 'x = ',vektorrm6b(lauf),' und y = ',vektorrm6b(lauf+1),$
    ' nicht moeglich!!'
    if vektorrm6b(lauf) + (durchmesserunter1/2) gt dimccd then print,$
    'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
    , 'x = ',vektorrm6b(lauf),' und y = ',vektorrm6b(lauf+1),$
    ' nicht moeglich!!'

    if vektorrm6b(lauf+1) + (durchmesserunter1/2) gt dimccd then print,$
    'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
    , 'x = ',vektorrm6b(lauf),' und y = ',vektorrm6b(lauf+1),$
    ' nicht moeglich!!'

    viervierziga1:
    ;tvsc1, congrid( (alog(untergkorr-min(untergkorr)+1)),512,512)
    laufeins = laufeins+1
endfor

tvsc1, congrid((alog(untergkorr-min(untergkorr)+1)),512,512)

laufeins = 0

for lauf = 0, anzahlm7-1, 2 do begin

    if vektorrm7b(lauf) - (durchmesserunter1/2) lt 1 then goto, viervierzig2
    if vektorrm7b(lauf+1) - (durchmesserunter1/2) lt 1 then goto, viervierzig2
    if vektorrm7b(lauf) + (durchmesserunter1/2) gt dimccd then goto, viervierzig2
    if vektorrm7b(lauf+1) + (durchmesserunter1/2) gt dimccd then goto, viervierzig2

    untergkorr((vektorrm7b(lauf)-(durchmesser/2)):$
    (vektorrm7b(lauf)+(durchmesser/2)-1),(vektorrm7b(lauf+1)-$
    (durchmesser/2)):(vektorrm7b(lauf+1)+(durchmesser/2)-1))$
    =kreis*vektorrm7bunter(laufeins)

    viervierzig2:

    if total(vektorrm7b) eq -42 then goto, viervierziga2

```

```

if vektorrm7b(lauf) - (durchmesserunter1/2) lt 1 then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
, 'x = ',vektorrm7b(lauf),' und y = ',vektorrm7b(lauf+1),$
' nicht moeglich!!'
if vektorrm7b(lauf+1) - (durchmesserunter1/2) lt 1 then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
, 'x = ',vektorrm7b(lauf),' und y = ',vektorrm7b(lauf+1),$
' nicht moeglich!!'
if vektorrm7b(lauf) + (durchmesserunter1/2) gt dimccd then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
, 'x = ',vektorrm7b(lauf),' und y = ',vektorrm7b(lauf+1),$
' nicht moeglich!!'

if vektorrm7b(lauf+1) + (durchmesserunter1/2) gt dimccd then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
, 'x = ',vektorrm7b(lauf),' und y = ',vektorrm7b(lauf+1),$
' nicht moeglich!!'

viervierziga2:
;tvsc1, congrid( (alog(untergkorr-min(untergkorr)+1)),512,512)
laufeins = laufeins+1
endfor

tvsc1, congrid((alog(untergkorr-min(untergkorr)+1)),512,512)

laufeins = 0

for lauf = 0, anzahlm8-1, 2 do begin

if vektorrm8b(lauf) - (durchmesserunter1/2) lt 1 then goto, viervierzig3
if vektorrm8b(lauf+1) - (durchmesserunter1/2) lt 1 then goto, viervierzig3
if vektorrm8b(lauf) + (durchmesserunter1/2) gt dimccd then goto, viervierzig3
if vektorrm8b(lauf+1) + (durchmesserunter1/2) gt dimccd then goto, viervierzig3

untergkorr((vektorrm8b(lauf)-(durchmesser/2)):$
(vektorrm8b(lauf)+(durchmesser/2)-1),(vektorrm8b(lauf+1)-$
(durchmesser/2)):(vektorrm8b(lauf+1)+(durchmesser/2)-1))$
=kreis*vektorrm8bunter(laufeins)

viervierzig3:

if total(vektorrm8b) eq -42 then goto, viervierziga3

if vektorrm8b(lauf) - (durchmesserunter1/2) lt 1 then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
, 'x = ',vektorrm8b(lauf),' und y = ',vektorrm8b(lauf+1),$
' nicht moeglich!!'
if vektorrm8b(lauf+1) - (durchmesserunter1/2) lt 1 then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
, 'x = ',vektorrm8b(lauf),' und y = ',vektorrm8b(lauf+1),$
' nicht moeglich!!'
if vektorrm8b(lauf) + (durchmesserunter1/2) gt dimccd then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
, 'x = ',vektorrm8b(lauf),' und y = ',vektorrm8b(lauf+1),$
' nicht moeglich!!'

if vektorrm8b(lauf+1) + (durchmesserunter1/2) gt dimccd then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
, 'x = ',vektorrm8b(lauf),' und y = ',vektorrm8b(lauf+1),$
' nicht moeglich!!'

viervierziga3:

```

```
;tvsc1, congrid( (alog(untergkorr-min(untergkorr)+1)),512,512)
laufeins = laufeins+1
endfor
```

```
tvsc1, congrid((alog(untergkorr-min(untergkorr)+1)),512,512)
```

```
laufeins = 0
```

```
for lauf = 0, anzahlm9-1, 2 do begin
```

```
if vektorrm9b(lauf) - (durchmesserunter1/2) lt 1 then goto, viervierzig4
if vektorrm9b(lauf+1) - (durchmesserunter1/2) lt 1 then goto, viervierzig4
if vektorrm9b(lauf) + (durchmesserunter1/2) gt dimccd then goto, viervierzig4
if vektorrm9b(lauf+1) + (durchmesserunter1/2) gt dimccd then goto, viervierzig4
```

```
untergkorr((vektorrm9b(lauf)-(durchmesser/2)):$
(vektorrm9b(lauf)+(durchmesser/2)-1),(vektorrm9b(lauf+1)-$
(durchmesser/2)):(vektorrm9b(lauf+1)+(durchmesser/2)-1))$
=kreis*vektorrm9bunter(laufeins)
```

```
viervierzig4:
```

```
if total(vektorrm9b) eq -42 then goto, viervierziga4
```

```
if vektorrm9b(lauf) - (durchmesserunter1/2) lt 1 then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
,'x = ',vektorrm9b(lauf),' und y = ',vektorrm9b(lauf+1),$
' nicht moeglich!!'
```

```
if vektorrm9b(lauf+1) - (durchmesserunter1/2) lt 1 then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
,'x = ',vektorrm9b(lauf),' und y = ',vektorrm9b(lauf+1),$
' nicht moeglich!!'
```

```
if vektorrm9b(lauf) + (durchmesserunter1/2) gt dimccd then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
,'x = ',vektorrm9b(lauf),' und y = ',vektorrm9b(lauf+1),$
' nicht moeglich!!'
```

```
if vektorrm9b(lauf+1) + (durchmesserunter1/2) gt dimccd then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
,'x = ',vektorrm9b(lauf),' und y = ',vektorrm9b(lauf+1),$
' nicht moeglich!!'
```

```
viervierziga4:
```

```
;tvsc1, congrid( (alog(untergkorr-min(untergkorr)+1)),512,512)
laufeins = laufeins+1
endfor
```

```
tvsc1, congrid((alog(untergkorr-min(untergkorr)+1)),512,512)
```

```
laufeins = 0
```

```
for lauf = 0, anzahlm10-1, 2 do begin
```

```
if vektorrm10b(lauf) - (durchmesserunter1/2) lt 1 then goto, viervierzig5
if vektorrm10b(lauf+1) - (durchmesserunter1/2) lt 1 then goto, viervierzig5
if vektorrm10b(lauf) + (durchmesserunter1/2) gt dimccd then goto, viervierzig5
if vektorrm10b(lauf+1) + (durchmesserunter1/2) gt dimccd then goto, viervierzig5
```

```
untergkorr((vektorrm10b(lauf)-(durchmesser/2)):$
(vektorrm10b(lauf)+(durchmesser/2)-1),(vektorrm10b(lauf+1)-$
(durchmesser/2)):(vektorrm10b(lauf+1)+(durchmesser/2)-1))$
=kreis*vektorrm10bunter(laufeins)
```

viervierzig5:

```
if total(vektorm10b) eq -42 then goto, viervierziga5
```

```
if vektorm10b(lauf) - (durchmesserunter1/2) lt 1 then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
  ,x = ',vektorm10b(lauf),' und y = ',vektorm10b(lauf+1),$
  ' nicht moeglich!!'
```

```
if vektorm10b(lauf+1) - (durchmesserunter1/2) lt 1 then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
  ,x = ',vektorm10b(lauf),' und y = ',vektorm10b(lauf+1),$
  ' nicht moeglich!!'
```

```
if vektorm10b(lauf) + (durchmesserunter1/2) gt dimccd then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
  ,x = ',vektorm10b(lauf),' und y = ',vektorm10b(lauf+1),$
  ' nicht moeglich!!'
```

```
if vektorm10b(lauf+1) + (durchmesserunter1/2) gt dimccd then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
  ,x = ',vektorm10b(lauf),' und y = ',vektorm10b(lauf+1),$
  ' nicht moeglich!!'
```

viervierziga5:

```
;tvsc1, congrid( (alog(untergkorr-min(untergkorr)+1)),512,512)
```

```
laufeins = laufeins+1
```

```
endfor
```

```
tvsc1, congrid((alog(untergkorr-min(untergkorr)+1)),512,512)
```

```
laufeins = 0
```

```
for lauf = 0, anzahlm11-1, 2 do begin
```

```
if vektorm11b(lauf) - (durchmesserunter1/2) lt 1 then goto, viervierzig6
if vektorm11b(lauf+1) - (durchmesserunter1/2) lt 1 then goto, viervierzig6
if vektorm11b(lauf) + (durchmesserunter1/2) gt dimccd then goto, viervierzig6
if vektorm11b(lauf+1) + (durchmesserunter1/2) gt dimccd then goto, viervierzig6
```

```
untergkorr((vektorm11b(lauf)-(durchmesser/2)):$
(vektorm11b(lauf)+(durchmesser/2)-1),(vektorm11b(lauf+1)-$
(durchmesser/2)):(vektorm11b(lauf+1)+(durchmesser/2)-1))$
=kreis*vektorm11bunter(laufeins)
```

viervierzig6:

```
if total(vektorm11b) eq -42 then goto, viervierziga6
```

```
if vektorm11b(lauf) - (durchmesserunter1/2) lt 1 then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
  ,x = ',vektorm11b(lauf),' und y = ',vektorm11b(lauf+1),$
  ' nicht moeglich!!'
```

```
if vektorm11b(lauf+1) - (durchmesserunter1/2) lt 1 then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
  ,x = ',vektorm11b(lauf),' und y = ',vektorm11b(lauf+1),$
  ' nicht moeglich!!'
```

```
if vektorm11b(lauf) + (durchmesserunter1/2) gt dimccd then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
  ,x = ',vektorm11b(lauf),' und y = ',vektorm11b(lauf+1),$
  ' nicht moeglich!!'
```

```
if vektorm11b(lauf+1) + (durchmesserunter1/2) gt dimccd then print,$
```

```
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
  ,x = ',vektorm11b(lauf),' und y = ',vektorm11b(lauf+1),$
  ' nicht moeglich!!'
```

```
viervierziga6:
;tvsc1, congrid( (alog(untergkorr-min(untergkorr)+1)),512,512)
laufeins = laufeins+1
endfor
```

```
tvsc1, congrid((alog(untergkorr-min(untergkorr)+1)),512,512)
```

```
laufeins = 0
```

```
for lauf = 0, anzahlm12-1, 2 do begin
```

```
  if vektorm12b(lauf) - (durchmesserunter1/2) lt 1 then goto, viervierzig7
  if vektorm12b(lauf+1) - (durchmesserunter1/2) lt 1 then goto, viervierzig7
  if vektorm12b(lauf) + (durchmesserunter1/2) gt dimccd then goto, viervierzig7
  if vektorm12b(lauf+1) + (durchmesserunter1/2) gt dimccd then goto, viervierzig7
```

```
  untergkorr((vektorm12b(lauf)-(durchmesser/2)):$
(vektorm12b(lauf)+(durchmesser/2)-1),(vektorm12b(lauf+1)-$
(durchmesser/2)):(vektorm12b(lauf+1)+(durchmesser/2)-1))$
  =kreis*vektorm12bunter(laefeins)
```

```
viervierzig7:
```

```
if total(vektorm12b) eq -42 then goto, viervierziga7
```

```
if vektorm12b(lauf) - (durchmesserunter1/2) lt 1 then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
  ,x = ',vektorm12b(lauf),' und y = ',vektorm12b(lauf+1),$
  ' nicht moeglich!!'
```

```
if vektorm12b(lauf+1) - (durchmesserunter1/2) lt 1 then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
  ,x = ',vektorm12b(lauf),' und y = ',vektorm12b(lauf+1),$
  ' nicht moeglich!!'
```

```
if vektorm12b(lauf) + (durchmesserunter1/2) gt dimccd then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
  ,x = ',vektorm12b(lauf),' und y = ',vektorm12b(lauf+1),$
  ' nicht moeglich!!'
```

```
if vektorm12b(lauf+1) + (durchmesserunter1/2) gt dimccd then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
  ,x = ',vektorm12b(lauf),' und y = ',vektorm12b(lauf+1),$
  ' nicht moeglich!!'
```

```
viervierziga7:
;tvsc1, congrid( (alog(untergkorr-min(untergkorr)+1)),512,512)
laufeins = laufeins+1
endfor
```

```
tvsc1, congrid((alog(untergkorr-min(untergkorr)+1)),512,512)
```

```
laufeins = 0
```

```
for lauf = 0, anzahlm13-1, 2 do begin
```

```
  if vektorm13b(lauf) - (durchmesserunter1/2) lt 1 then goto, viervierzig8
  if vektorm13b(lauf+1) - (durchmesserunter1/2) lt 1 then goto, viervierzig8
  if vektorm13b(lauf) + (durchmesserunter1/2) gt dimccd then goto, viervierzig8
  if vektorm13b(lauf+1) + (durchmesserunter1/2) gt dimccd then goto, viervierzig8
```

```

untergkorr((vektorm13b(lauf)-(durchmesser/2)):$
(vektorm13b(lauf)+(durchmesser/2)-1),(vektorm13b(lauf+1)-$
(durchmesser/2)):(vektorm13b(lauf+1)+(durchmesser/2)-1))$
=kreis*vektorm13bunter(laufeins)

```

viervierzig8:

```
if total(vektorm13b) eq -42 then goto, viervierziga8
```

```
if vektorm13b(lauf) - (durchmesserunter1/2) lt 1 then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
, 'x = ',vektorm13b(lauf),' und y = ',vektorm13b(lauf+1),$
' nicht moeglich!!'

```

```
if vektorm13b(lauf+1) - (durchmesserunter1/2) lt 1 then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
, 'x = ',vektorm13b(lauf),' und y = ',vektorm13b(lauf+1),$
' nicht moeglich!!'

```

```
if vektorm13b(lauf) + (durchmesserunter1/2) gt dimccd then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
, 'x = ',vektorm13b(lauf),' und y = ',vektorm13b(lauf+1),$
' nicht moeglich!!'

```

```
if vektorm13b(lauf+1) + (durchmesserunter1/2) gt dimccd then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
, 'x = ',vektorm13b(lauf),' und y = ',vektorm13b(lauf+1),$
' nicht moeglich!!'

```

viervierziga8:

```

;tvsc1, congrid( (alog(untergkorr-min(untergkorr)+1)),512,512)
laufeins = laufeins+1
endfor

```

```
tvsc1, congrid((alog(untergkorr-min(untergkorr)+1)),512,512)
```

laufeins = 0

```
for lauf = 0, anzahlm14-1, 2 do begin
```

```

if vektorm14b(lauf) - (durchmesserunter1/2) lt 1 then goto, viervierzig9
if vektorm14b(lauf+1) - (durchmesserunter1/2) lt 1 then goto, viervierzig9
if vektorm14b(lauf) + (durchmesserunter1/2) gt dimccd then goto, viervierzig9
if vektorm14b(lauf+1) + (durchmesserunter1/2) gt dimccd then goto, viervierzig9

```

```

untergkorr((vektorm14b(lauf)-(durchmesser/2)):$
(vektorm14b(lauf)+(durchmesser/2)-1),(vektorm14b(lauf+1)-$
(durchmesser/2)):(vektorm14b(lauf+1)+(durchmesser/2)-1))$
=kreis*vektorm14bunter(laufeins)

```

viervierzig9:

```
if total(vektorm14b) eq -42 then goto, viervierziga9
```

```
if vektorm14b(lauf) - (durchmesserunter1/2) lt 1 then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
, 'x = ',vektorm14b(lauf),' und y = ',vektorm14b(lauf+1),$
' nicht moeglich!!'

```

```
if vektorm14b(lauf+1) - (durchmesserunter1/2) lt 1 then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
, 'x = ',vektorm14b(lauf),' und y = ',vektorm14b(lauf+1),$
' nicht moeglich!!'

```

```
if vektorm14b(lauf) + (durchmesserunter1/2) gt dimccd then print,$
```

```
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
  ,x = ',vektorm14b(lauf),' und y = ',vektorm14b(lauf+1),$
  ' nicht moeglich!!'

if vektorm14b(lauf+1) + (durchmesserunter1/2) gt dimccd then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
  ,x = ',vektorm14b(lauf),' und y = ',vektorm14b(lauf+1),$
  ' nicht moeglich!!'

viervierziga9:
;tvsc1, congrid( (alog(untergkorr-min(untergkorr)+1)),512,512)
laufeins = laufeins+1
endfor

tvsc1, congrid((alog(untergkorr-min(untergkorr)+1)),512,512)

laufeins = 0

for lauf = 0, anzahlm15-1, 2 do begin

if vektorm15b(lauf) - (durchmesserunter1/2) lt 1 then goto, viervierzig10
if vektorm15b(lauf+1) - (durchmesserunter1/2) lt 1 then goto, viervierzig10
if vektorm15b(lauf) + (durchmesserunter1/2) gt dimccd then goto, viervierzig10
if vektorm15b(lauf+1) + (durchmesserunter1/2) gt dimccd then goto, viervierzig10

untergkorr((vektorm15b(lauf)-(durchmesser/2)):$
(vektorm15b(lauf)+(durchmesser/2)-1),(vektorm15b(lauf+1)-$
(durchmesser/2)):(vektorm15b(lauf+1)+(durchmesser/2)-1))$
=kreis*vektorm15bunter(laefeins)

viervierzig10:

if total(vektorm15b) eq -42 then goto, viervierziga10

if vektorm15b(lauf) - (durchmesserunter1/2) lt 1 then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
  ,x = ',vektorm15b(lauf),' und y = ',vektorm15b(lauf+1),$
  ' nicht moeglich!!'
if vektorm15b(lauf+1) - (durchmesserunter1/2) lt 1 then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
  ,x = ',vektorm15b(lauf),' und y = ',vektorm15b(lauf+1),$
  ' nicht moeglich!!'
if vektorm15b(lauf) + (durchmesserunter1/2) gt dimccd then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
  ,x = ',vektorm15b(lauf),' und y = ',vektorm15b(lauf+1),$
  ' nicht moeglich!!'

if vektorm15b(lauf+1) + (durchmesserunter1/2) gt dimccd then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
  ,x = ',vektorm15b(lauf),' und y = ',vektorm15b(lauf+1),$
  ' nicht moeglich!!'

viervierziga10:
;tvsc1, congrid( (alog(untergkorr-min(untergkorr)+1)),512,512)
laufeins = laufeins+1
endfor

tvsc1, congrid((alog(untergkorr-min(untergkorr)+1)),512,512)

laufeins = 0

for lauf = 0, anzahlm16-1, 2 do begin
```

```

if vektorrm16b(lauf) - (durchmesserunter1/2) lt 1 then goto, viervierzig11
if vektorrm16b(lauf+1) - (durchmesserunter1/2) lt 1 then goto, viervierzig11
if vektorrm16b(lauf) + (durchmesserunter1/2) gt dimccd then goto, viervierzig11
if vektorrm16b(lauf+1) + (durchmesserunter1/2) gt dimccd then goto, viervierzig11

untergkorr((vektorrm16b(lauf)-(durchmesser/2)):$
(vektorrm16b(lauf)+(durchmesser/2)-1),(vektorrm16b(lauf+1)-$
(durchmesser/2)):(vektorrm16b(lauf+1)+(durchmesser/2)-1))$
=kreis*vektorrm16bunter(laufeins)

viervierzig11:

if total(vektorrm16b) eq -42 then goto, viervierziga11

if vektorrm16b(lauf) - (durchmesserunter1/2) lt 1 then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
',x = ',vektorrm16b(lauf),' und y = ',vektorrm16b(lauf+1),$
' nicht moeglich!!'
if vektorrm16b(lauf+1) - (durchmesserunter1/2) lt 1 then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
',x = ',vektorrm16b(lauf),' und y = ',vektorrm16b(lauf+1),$
' nicht moeglich!!'
if vektorrm16b(lauf) + (durchmesserunter1/2) gt dimccd then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
',x = ',vektorrm16b(lauf),' und y = ',vektorrm16b(lauf+1),$
' nicht moeglich!!'

if vektorrm16b(lauf+1) + (durchmesserunter1/2) gt dimccd then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
',x = ',vektorrm16b(lauf),' und y = ',vektorrm16b(lauf+1),$
' nicht moeglich!!'

viervierziga11:
;tvsc1, congrid( (alog(untergkorr-min(untergkorr)+1)),512,512)
laufeins = laufeins+1
endfor

tvsc1, congrid((alog(untergkorr-min(untergkorr)+1)),512,512)

laufeins = 0

for lauf = 0, anzahlm17-1, 2 do begin

if vektorrm17b(lauf) - (durchmesserunter1/2) lt 1 then goto, viervierzig12
if vektorrm17b(lauf+1) - (durchmesserunter1/2) lt 1 then goto, viervierzig12
if vektorrm17b(lauf) + (durchmesserunter1/2) gt dimccd then goto, viervierzig12
if vektorrm17b(lauf+1) + (durchmesserunter1/2) gt dimccd then goto, viervierzig12

untergkorr((vektorrm17b(lauf)-(durchmesser/2)):$
(vektorrm17b(lauf)+(durchmesser/2)-1),(vektorrm17b(lauf+1)-$
(durchmesser/2)):(vektorrm17b(lauf+1)+(durchmesser/2)-1))$
=kreis*vektorrm17bunter(laufeins)

viervierzig12:

if total(vektorrm17b) eq -42 then goto, viervierziga12

if vektorrm17b(lauf) - (durchmesserunter1/2) lt 1 then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
',x = ',vektorrm17b(lauf),' und y = ',vektorrm17b(lauf+1),$
' nicht moeglich!!'

```

```
if vektorm17b(lauf+1) - (durchmesserunter1/2) lt 1 then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
  ,x = ',vektorm17b(lauf),' und y = ',vektorm17b(lauf+1),$
  ' nicht moeglich!!'
if vektorm17b(lauf) + (durchmesserunter1/2) gt dimccd then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
  ,x = ',vektorm17b(lauf),' und y = ',vektorm17b(lauf+1),$
  ' nicht moeglich!!'

if vektorm17b(lauf+1) + (durchmesserunter1/2) gt dimccd then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
  ,x = ',vektorm17b(lauf),' und y = ',vektorm17b(lauf+1),$
  ' nicht moeglich!!'

viervierziga12:
;tvsc1, congrid( (alog(untergkorr-min(untergkorr)+1)),512,512)
laufeins = laufeins+1
endfor

tvsc1, congrid((alog(untergkorr-min(untergkorr)+1)),512,512)

laufeins = 0

for lauf = 0, anzahlm18-1, 2 do begin

if vektorm18b(lauf) - (durchmesserunter1/2) lt 1 then goto, viervierzig13
if vektorm18b(lauf+1) - (durchmesserunter1/2) lt 1 then goto, viervierzig13
if vektorm18b(lauf) + (durchmesserunter1/2) gt dimccd then goto, viervierzig13
if vektorm18b(lauf+1) + (durchmesserunter1/2) gt dimccd then goto, viervierzig13

untergkorr((vektorm18b(lauf)-(durchmesser/2)):$
(vektorm18b(lauf)+(durchmesser/2)-1),(vektorm18b(lauf+1)-$
(durchmesser/2)):(vektorm18b(lauf+1)+(durchmesser/2)-1))$
=kreis*vektorm18bunter(laefeins)

viervierzig13:

if total(vektorm18b) eq -42 then goto, viervierziga13

if vektorm18b(lauf) - (durchmesserunter1/2) lt 1 then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
  ,x = ',vektorm18b(lauf),' und y = ',vektorm18b(lauf+1),$
  ' nicht moeglich!!'
if vektorm18b(lauf+1) - (durchmesserunter1/2) lt 1 then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
  ,x = ',vektorm18b(lauf),' und y = ',vektorm18b(lauf+1),$
  ' nicht moeglich!!'
if vektorm18b(lauf) + (durchmesserunter1/2) gt dimccd then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
  ,x = ',vektorm18b(lauf),' und y = ',vektorm18b(lauf+1),$
  ' nicht moeglich!!'

if vektorm18b(lauf+1) + (durchmesserunter1/2) gt dimccd then print,$
'Eine konventionelle Untergrundkorrektur ist fuer den Reflex '$
  ,x = ',vektorm18b(lauf),' und y = ',vektorm18b(lauf+1),$
  ' nicht moeglich!!'

viervierziga13:
;tvsc1, congrid( (alog(untergkorr-min(untergkorr)+1)),512,512)
laufeins = laufeins+1
endfor
```

```
tvsc1, congrid((alog(untergkorr-min(untergkorr)+1)),512,512)

unterbildkonv = bildkonv - untergkorr

; Bild zum Untergrungkorregierten Bild wieder Zusammenfuegen
  get_lun, unit
  falsch = 1
  openw, unit, 'unterbildkonv'
  schwanz0 = assoc(unit, bytarr(schwanzlaenge), schwanzstart)
  schwanz0(0) = schwanz
  kopf0 = assoc(unit, bytarr(kopfccd) )
  kopf0(0) = kopf
  inhalt = assoc(unit, intarr(dimccd,dimccd),kopfccd)
  inhalt(0) = swap_endian(fix(unterbildkonv))
  close, unit
  free_lun, unit
  print, 'Geschrieben: unterbildkonv'
  tvsc1, congrid( (alog(unterbildkonv-min(unterbildkonv)+1)),512,512)

endif

; Bild wieder zum GATAN Format zusammensetzen

;bild = fix(bild)
  get_lun, unit
  falsch = 1
  openw, unit, 'bildkonv'
  schwanz0 = assoc(unit, bytarr(schwanzlaenge), schwanzstart)
  schwanz0(0) = schwanz
  kopf0 = assoc(unit, bytarr(kopfccd) )
  kopf0(0) = kopf
  inhalt = assoc(unit, intarr(dimccd,dimccd),kopfccd)
  inhalt(0) = swap_endian(fix(bildkonv))
  close, unit
  free_lun, unit
  print, 'Geschrieben: bildkonv'
  tvsc1, congrid( (alog(bildkonv-min(bildkonv)+1)),512,512)
  fehler: if (falsch eq 0) then print, 'Fehler bei I/O von ',name1ccd, '!'

end
```