



Arbeitspaket 3: Langzeitarchivierung von Forschungsdaten

Formatkonvertierung TIFF → NeXus¹

Autoren	Arbeitspaket 3: Langzeitarchivierung von Forschungsdaten
Editoren	Rathmann, Schlünzen
Datum	20. April 2012
Dokument Status	Entwurf
Dokument Version	0.2.2

Änderungen

Version	Datum	Name(n)	Kurzinfo
0.1.0	14.03.2011	Henne, Panzer, Samadi	Template
0.2.0	12.10.2011	Rathmann	Software-Dokumentation zu tiff2nexus.py
0.2.1	12.04.2012	Rathmann, Schlünzen	Ergänzungen
0.2.2	20.04.2012	Rathmann	kleine Änderungen

¹This work is created by the WissGrid project. The project is funded by the German Federal Ministry of Education and Research (BMBF).

Inhaltsverzeichnis

1	Einleitung und Anforderungen	3
2	Design	4
2.1	Das Konvertierungsskript tiff2nexus.py	4
3	Technische Umsetzung	6
4	Installation und Benutzung von tiff2nexus.py	8
4.1	Systemanforderungen	8
4.2	Installation	8
4.3	Getting Started	8
5	FAQ	10
6	Erste Erfahrungen und VRE-Integration	12
6.1	GISAXS-Datensatz	12
6.2	Visualisierung	12
6.3	Archivierung	12
6.4	VRE-Integration	13
7	Hintergrundinformationen	15
7.1	Das Bilddateiformat TIFF (Tagged Image File Format)	15
7.2	Das Datenformat NeXus	15
	Literaturverzeichnis	16

1 Einleitung und Anforderungen

Die Photonenphysik nutzt die Photonen (Lichtteilchen), die als Synchrotronstrahlung bei der Richtungsänderung eines schnellen Strahls geladener Teilchen entstehen. Synchrotronstrahlung ist von ihrer Helligkeit und spektralen Zusammensetzung her für die Kristallstrukturanalyse hervorragend geeignet. Gracing Incidence Small Angle X-ray Scattering (GISAXS) Experimente untersuchen zum Beispiel das Aufwuchsverhalten von Materialien auf wohl definierten Oberflächen in hoher zeitlicher wie räumlicher Auflösung. Bei diesem Verfahren werden Oberflächen mit Materialien bedampft und die Röntgen-Diffraktion an der Oberfläche mittels eines Detektors in kurzen zeitlichen Abständen gemessen. An der Beamline P03 am Synchrotron Petra-III des DESY (1) kommt dabei ein Pilatus-Detektor zum Einsatz, ein verbreiteter Detektor-Typ. Eine solche Zeitserie besteht dann typischerweise aus einigen zehntausend bis zu einigen hunderttausend Einzelaufnahmen. Gespeichert werden die Einzelaufnahmen entweder im Bildformat TIFF (2) mit nicht ganz TIFF-konformem Header oder als CBF-Image, einem Binärformat mit verlustfreier Komprimierung, das speziell für kristallographische Experimente entwickelt wurde.

Andere Experimente an Photon- oder Neutronen-Quellen sind ganz ähnlich strukturiert. Fast immer besteht ein Datensatz aus einer extrem großen Anzahl von Einzelaufnahmen von 2D-Detektoren, die eine drei- oder vier-dimensionale Struktur repräsentieren. Zwar ist mittelfristig geplant, alle Daten schon in der Anfangsphase in einem portablen Container-Format — NeXus/HDF5 — abzulegen, um die Daten dann in ein Archiv überführen zu können. Bis dahin werden allerdings PBytes von Daten zu den bereits unstrukturiert akkumulierten Daten hinzukommen. Auch für diese Daten wird eine Lösung gesucht, die eine effiziente Speicherung und Langzeit-Archivierung erlaubt.

Die Photonenphysik hat daher über DESY Bedarf an einem Werkzeug für die Formatkonvertierung TIFF → NeXus angemeldet. NeXus (3) ist ein Standardformat in der Röntgen-, Neutronen- und Myonenphysik. Die vom Detektor Pilatus 300K erzeugten TIFF-Dateien enthalten je ein Bild und haben eine einheitliche Headerlänge von 4096 Bytes. In den Header sind Metainformationen als Klartext (ASCII) integriert. Binäre Daten werden vom Detektor in der Byte-Reihenfolge little endian geschrieben (Erläuterungen siehe Abschnitt 7.1).

Im Rahmen von WissGrid wird hiermit ein entsprechendes Konvertierungswerkzeug zur Verfügung gestellt. Die Anforderungen wurden zuvor vom DESY entwickelt:

- Zusammenfassung einer Serie von TIFF-Dateien zu einer NeXus-Datei
- Konvertierung der Bilddaten
- Auslesen von Metadaten aus den TIFF-Headern und Zuordnung zu NeXus-Gruppen
- Einfügen von Default-Information in NeXus-Gruppen

2 Design

2.1 Das Konvertierungsskript tiff2nexus.py

tiff2nexus.py ist ein Werkzeug für die Konvertierung TIFF → NeXus. Es sucht im Input-Verzeichnis nach TIFF-Dateien, liest aus diesen Bild- und Metadaten aus und schreibt die Daten alle zusammen in eine NeXus-Datei. Die Daten werden nach dem in Abb. 1 gezeigten Schema in NeXus-Datenfelder eingeordnet. Kästen symbolisieren NeXus-Gruppen. NeXus-Attribute beginnen mit @ und sind dem Datenfeld darüber zugeordnet.

Nicht alle für die Data Curation wichtigen Daten befinden sich in TIFF-Dateien. In einer Reihe von NeXus-Datenfeldern werden deshalb Rechenwerte oder ein Default geschrieben. Solche Inhalte sind in Abb. 1 mit einem Gleichheitszeichen gekennzeichnet. Default-Werte sollten nach der Formatkonvertierung durch die tatsächlichen Werte ersetzt werden.

Die Bilddaten aller TIFF-Bilder befinden sich nach der Konvertierung im dreidimensionalen Datenfeld data. Der vordere Index ist die Nummer des früheren TIFF-Bildes, der mittlere Index die Nummer des x-Pixels und der hintere die Nummer des y-Pixels. Die Bilddaten können in entry/instrument/detector/data und in data/data gefunden werden, die miteinander verlinkt sind (gestrichelte Linie).

tiff2nexus.py ist als Python-Skript realisiert worden. Es verwendet die folgenden Python-Module:

- nxs (Bestandteil der NeXus-Software)
- Image (Bildbearbeitung)
- numpy (Numeric-Python-Erweiterung)
- sys, os (systemspezifische Funktionen, Schnittstellen zum Betriebssystem)

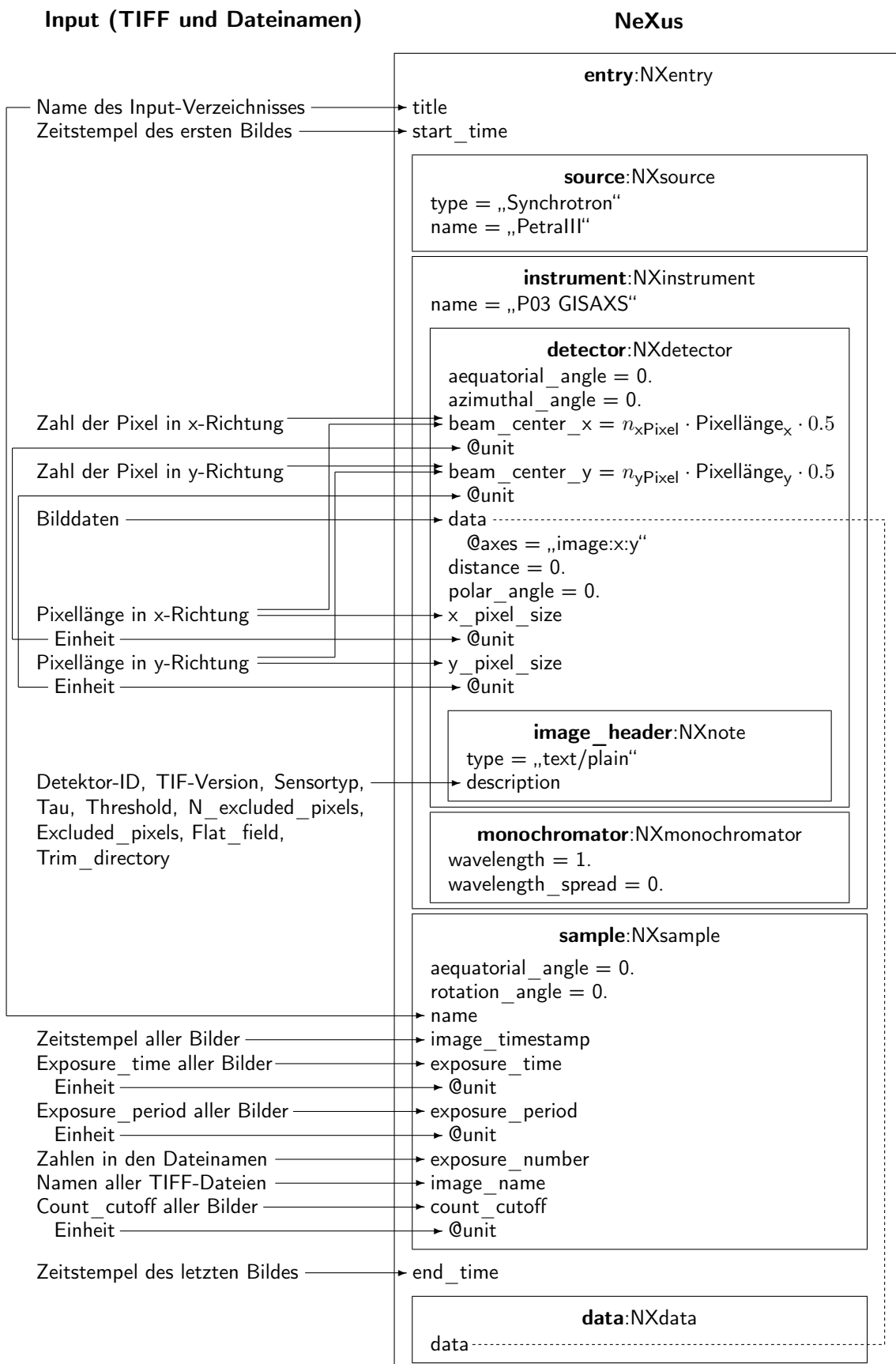


Abbildung 1: Zuordnung der TIFF-Daten zu NeXus-Datenfeldern

3 Technische Umsetzung

Typ: Unix-artiger Befehl

Programmiersprache: Python

Input:

Dem Programmaufruf wird der Name eines Input-Verzeichnisses als Argument mitgegeben. Von den im Input-Verzeichnis befindlichen Dateien versucht tiff2nexus.py alle diejenigen zu konvertieren, deren Name dem Schema *anyname_N.tif* entspricht. *anyname* ist dabei eine beliebige Zeichenkette und *N* eine natürliche Zahl bestehend aus einer oder mehreren Ziffern. Dazwischen muss sich der Unterstrich befinden.

In den TIFF-Dateien werden die folgenden Daten erwartet:

- In Byte 0 bis 1 „II“ für Byte-Reihenfolge little endian
- In Byte 2 bis 3 die Binärzahl 42
- In Byte 4 bis 7 als Binärzahl der Offset für das Image File Directory (IFD)
- In Byte 30 bis 49 der Zeitstempel im ASCII-Format
- In Byte 62 bis 92 den Detektortyp im ASCII-Format
- An beliebiger Stelle „# Pixel_size“, dahinter Pixelbreite und Einheit, Pixelhöhe und Einheit, alles ASCII und durch Leerzeichen voneinander getrennt
- Dahinter die folgenden Angaben im ASCII-Format in der angegebenen Reihenfolge, jeweils in einer eigenen Zeile beginnend mit „# “:
 1. Sensortyp (beliebiger Text)
 2. „Exposure_time“, Real-Zahl und Maßeinheit, durch Leerzeichen voneinander getrennt
 3. „Exposure_period“, Real-Zahl und Maßeinheit, durch Leerzeichen voneinander getrennt
 4. „Tau“, dahinter beliebiger Text
 5. „Count_cutoff“, Integer-Zahl und Maßeinheit, durch Leerzeichen voneinander getrennt
 6. „Threshold_setting“, dahinter beliebiger Text
 7. „N_excluded_pixels“, dahinter beliebiger Text
 8. „Excluded_pixels“, dahinter beliebiger Text
 9. „Flat_field“, dahinter beliebiger Text
 10. „Trim_directory“, dahinter beliebiger Text
- In Tag 256 die Bildbreite (Pixelzahl) als Binärzahl

- In Tag 257 die Bildhöhe (Pixelzahl) als Binärzahl
- In Tag 258 die Binärzahlen 16 oder 32 für den Integer-Typ in Bit/Pixel
- Die Bilddaten

Eine Reihe von Größen werden als in allen Bildern einer Serie als gleich vorausgesetzt und nur einmal in der NeXus-Datei abgelegt:

- Bildgröße
- Integer-Typ 16 oder 32 Bit/Pixel
- Pixelgröße
- Sensortyp
- Tau
- Threshold_setting
- N_excluded_pixels
- Excluded_pixels
- Flat_field
- Trim_directory

Wenn diese Größen nicht in allen eingelesenen TIFF-Bildern übereinstimmen, bricht die Konvertierung mit einer Fehlermeldung ab.

Output:

tiff2nexus.py erzeugt eine NeXus-Datei wahlweise im Format NeXus/HDF5 oder NeXus/XML. Default ist NeXus/HDF5.

Meldungen:

tiff2nexus.py erzeugt keine Log-Dateien. Fehlermeldungen werden auf den Standardfehlerkanal geschrieben. Die Namen der durch tiff2nexus.py ignorierten Dateien, die die obige Namenskonvention nicht erfüllen, werden auf den Standardausgabekanal geschrieben.

Tests:

Die Formatkonvertierung wurde mit mehreren Bilderserien des DESY getestet, die vom Detektor Pilatus 300K stammen.

4 Installation und Benutzung von tiff2nexus.py

4.1 Systemanforderungen

Sie brauchen ein unixartiges Betriebssystem. Getestet wurde tiff2nexus.py mit Linux-Kernel 2.6.37.6-0.7-desktop

4.2 Installation

Welche andere Software muss zuerst installiert werden?

Bevor Sie tiff2nexus.py nutzen können, müssen die folgenden Pakete installiert sein:

- NeXus (von <http://download.nexusformat.org/kits/>, getestet mit Version 4.2.1)
- HDF5 runtime library (getestet mit Version 1.8.8-13.1)
- Python (getestet mit Version 2.7)
- Die Python Imaging Library (getestet mit Version 1.1.7-5.1)

Python (4), die Python Imaging Library und die HDF5 runtime library sind normaler Bestandteil einer Linux-Distribution.

Wie installiere ich tiff2nexus.py?

1. Entpacken Sie das zip-komprimierte Paket:
 - `unzip tiff2nexus.0.1.zip`
2. Kopieren Sie tiff2nexus.py in Ihr Arbeitsverzeichnis.

4.3 Getting Started

Wie kann ich ein Testbeispiel laufen lassen?

1. Kopieren Sie tiff2nexus.py und das Verzeichnis test mit den TIFF-Testdateien in Ihr Arbeitsverzeichnis.
2. In einer Shell tippe
 - `python tiff2nexus.py test`Bitte haben Sie etwas Geduld. Der Befehl konvertiert die TIFF-Bilder aus test in die NeXus/HDF5-Datei test.h5
3. Die erzeugte Datei sollte mit der im Paket mitgelieferten bis auf einen Zeitstempel und evtl. Dateinamen und HDF5-Version übereinstimmen. test.h5 kann z.B. mit `nxbrowse` angeschaut werden, das Bestandteil des NeXus-Pakets ist. Auf bequemere Art können die Dateien mit `h5diff` verglichen werden, das Bestandteil des HDF5-Paketes ist.

Wie kann ich tiff2nexus.py starten?

Verwenden Sie anstelle von test Ihr Input-Verzeichnis und verfahren Sie wie beim Test. Aufruf allgemein:

```
python tiff2nexus.py <inputdir> [ <outputfile> [hdf5|xml] ]
```

Mit dem letzten optionalen Befehlsargument wird gesteuert, ob HDF5 (5) oder XML erzeugt wird. Default ist HDF5.

5 FAQ

Was ist tiff2nexus.py?

tiff2nexus.py ist ein Formatkonvertierer, der vom Detektor Pilatus 300K erzeugte TIFF-Dateien in einer NeXus-Datei zusammenfassen kann. Die Output-Datei ist entweder eine NeXus/HDF5- oder eine NeXus/XML-Datei.

Wo befindet sich ein Detektor Pilatus 300K?

Ein Pilatus 300K wird als Stahlgangdetektor am DESY in Hamburg an der Röntgenquelle PETRA III betrieben.

Für wen und was ist tiff2nexus.py gedacht?

tiff2nexus.py ist für die Photonophysik-Community gedacht und speziell an den Detektor Pilatus 300K und dessen TIFF-Output angepasst. Neben der bloßen Formatkonvertierung fasst tiff2nexus.py mehrere TIFF-Dateien zu einer NeXus-Datei zusammen und kann so die Archivierung von Messdaten unterstützen.

Kann tiff2nexus.py nur im Grid verwendet werden?

tiff2nexus.py kann auch stand-alone oder in einer anderen virtuellen Forschungs-umgebung (VRE) verwendet werden.

Wie kann ich tiff2nexus.py bekommen?

tiff2nexus.py ist im Paket tiff2nexus enthalten. Dieses können Sie per Download von <http://www.wissgrid.de/publikationen/deliverables/wp3.html> bekommen.

Kann tiff2nexus.py mehrere TIFF-Dateien in einem Befehl konvertieren?

Ja, es werden alle TIFF-Dateien konvertiert, die sich im Input-Verzeichnis befinden und die die in der nächsten Antwort beschriebene Namenskonvention erfüllen.

Welche Dateien werden konvertiert, welche ausgelassen?

Von den im Input-Verzeichnis befindlichen Dateien versucht tiff2nexus.py alle diejenigen zu konvertieren, deren Name dem Schema *anyname_#.tif* entspricht. *anyname* ist dabei eine beliebige Zeichenkette und # eine ganze Zahl bestehend aus einer oder mehreren Ziffern. Alle anderen Dateien werden übergangen, d.h. nicht konvertiert.

Welchen Namen bekommt die Output-Datei?

Wenn der Nutzer keinen Namen für die Output-Datei vorgegeben hat, wird der Name des Input-Verzeichnisses genommen und die Erweiterung *.h5* angehängt. *.h5* steht für HDF5, das Default-Outputformat.

Wie sind die Bilddaten in der NeXus-Datei angeordnet?

Die Bilddaten aller TIFF-Bilder befinden sich nach der Konvertierung im dreidimensionalen Datenfeld *data*. Der vordere Index ist die Nummer des TIFF-Bildes, der mittlere Index die Nummer des x-Pixels und der hintere die Nummer des y-Pixels.

Werden die TIFF-Dateien vor der Formatkonvertierung validiert?

Nein, durch tiff2nexus.py wird nur der Teil des TIFF-Standards abgeprüft, dessen Nichterfüllung tiff2nexus.py zum Absturz bringen würde. Überprüft wird auch, ob alle Metainformationen im TIFF-Header vorhanden sind, die der Detektor Pilatus 300K liefern sollte. Fehlt es an Metainformation oder wird ein Fehler entdeckt, wird die Formatkonvertierung beendet und eine Fehlermeldung auf den Standardfehlerkanal geschrieben.

Meine Konvertierung nach NeXus/XML kommt nicht zum Ende.

Scheinbare Endlosläufe und Abbrüche mit der Fehlermeldung „Speicherzugriffsfehler“ sind bei großen Datenmengen beobachtet worden. Die Ursache liegt nicht in tiff2nexus.py selbst, sondern wird in der verwendeten NeXus-Bibliothek vermutet. Konvertieren Sie stattdessen nach NeXus/HDF5.

Meine Frage ist nicht auf dieser Seite. Wo kann ich Antwort bekommen?

Wenden Sie sich an die Entwickler von tiff2nexus.py, Torsten Rathmann, rathmann(at)dkrz.de, und Frank Schlünzen, frank.schlunzen(at)desy.de

6 Erste Erfahrungen und VRE-Integration

6.1 GISAXS-Datensatz

Der GISAXS-Datensatz, der als typisches Beispiel verwendet wurde, bestand aus 100000 TIFF-Bildern mit einem Volumen von etwa 12 GByte. Durch Kompression lässt sich das Daten-Volumen auf etwa 900 MB reduzieren. Bei der Konvertierung in eine HDF5-Datei (5) werden die 100000 Aufnahmen in einer nicht-komprimierten 3D-Matrix abgelegt. Die Metadaten, die aus den modifizierten TIFF-Headern der Einzelaufnahmen extrahiert werden können, werden in die Metadaten-Struktur von NeXus überführt. Die Daten-Volumina der HDF5-Datei, komprimiert wie unkomprimiert, sind praktisch identisch mit denen der Original-Dateien. Der Transfer der HDF5-Datei im Vergleich zum TIFF ist allerdings bereits deutlich schneller (um einen Faktor 3.6 bei einer Replikation zwischen zwei Cluster-Filesystemen), da deutlich weniger Metadaten-Operationen nötig werden.

6.2 Visualisierung

Die Vorzüge der HDF5-Dateien liegen aber weniger auf der Daten-Transfer-Seite als in Bezug auf die simple Visualisierung, die rapide beschleunigte Analyse sowie der Archivierung. Es stehen zum Beispiel eine Vielzahl von Werkzeugen zur Verfügung, die es erlauben die Daten schnell und komfortabel zu visualisieren oder durch die Metadaten zu gehen. Dies schließt populäre kommerzielle Software-Pakete wie Matlab oder IDL als auch die HDF-eigenen Programme mit ein. Abbildung 2 zeigt ein Beispiel für den erwähnten Datensatz, wie er sich in HDFview darstellt. Der gesamte NeXus-Metadatenbaum wie eine Animation der Zeitserie lassen sich damit problemlos darstellen.

6.3 Archivierung

Die eigentliche Stärke der HDF5-Konvertierung liegt in der Archivierung. Die 100000 Einzelaufnahmen in dieser Form zu archivieren verbietet sich vornherein. Die Vielzahl kleiner Dateien lassen sich nur äußerst ineffizient auf Tapes abspeichern. Die Restaurierung eines solchen Datensatzes von Band nimmt sprichwörtlich Tage in Anspruch, da der zeitlimitierende Faktor die Positionierung der Leseköpfe auf dem Band ist, eine Operation, die mit der Anzahl der Dateien skaliert. Die HDF5-Datei kann hingegen in maximal wenigen Minuten restauriert werden.

Die gängige Praxis war daher bislang, die Bilder eines Datensatz in einem TARball zu speichern, und diese akkumulierten Daten ins Archiv zu stellen. Die Nachteile dieser Prozedur liegen auf der Hand: es gibt keine Möglichkeit direkt auf die Metadaten zuzugreifen; einzelne Aufnahmen können weder extrahiert noch visualisiert werden ohne die ganze TAR-Datei vollständig zu prozessieren. Das Format der Aufnahmen wird zudem wenig transparent. Auch wenn die TAR-Dateien sich langfristig prozessieren lassen sollten, ist dies für die darin enthaltenen TIFF-Dateien, die zudem einen modifizierten Header aufweisen, nicht so ohne weiteres gegeben, was eine Anpassung der Formatierung zu einem

späteren Zeitpunkt erschweren könnte. Die zurzeit am PetraIII-Synchrotron praktizierte Archivierungs-Prozedur sieht grob umrissen folgendermaßen aus:

- Paketierung der Einzelaufnahmen.
 - Für die GISAXS-Daten ist dies jetzt in Form von NeXus/HDF5 möglich.
 - Andere Experimente müssen sich fast ausschließlich des TARs bedienen.
- Extraktion der Metadaten und Ingest in einen Datenkatalog.
 - Für die NeXus-Daten werden die Metadaten automatisiert erfasst.
 - Für die Daten, die noch nicht im HDF5-Format vorliegen, muss eine manuell erstellte Datei als Metadaten-Quelle dienen, was der Qualität der Metadaten alles andere als zuträglich ist.
 - Der Ingest wird durch den Messzeit-Betreuer und nicht durch den Nutzer selbst erfolgen. Dies sichert minimale Interferenzen zwischen Messzeit-Betrieb und Archivierung.
 - Zur Zeit basiert der Daten-Katalog auf einer inhouse-Entwicklung. Eine Migration zu einer ICAT/TOPCAT (6), (7) basierten Lösung wird zur Zeit evaluiert. Dies hätte zudem den Vorteil, dass ICAT über eine XML-Schnittstelle verfügt, die es erlaubt die NeXus-Metadaten transparent und automatisch in den Datenkatalog einzuspeisen.

6.4 VRE-Integration

Entgegen der ursprünglichen Absicht ist zur Zeit keine Grid-Integration vorgesehen, da angesichts der einfachen und schnellen Konvertierung mit tiff2nexus kein Bedarf gesehen wird. Durch die Erfassung der Daten und Metadaten im ICAT-Datenkatalog und dem TOPCAT Web-Frontend können die Daten komfortabel dem Anwender (remote) zur Verfügung gestellt werden. Da am HASYLAB dCache als Storage und Archivierungssystem verwendet wird, stehen weitere Schnittstellen wie Webdav, nfs4.1, dccp oder gsi-/grid-ftp zur Nutzung der Daten auf lokalen und Grid-Infrastrukturen zur Verfügung.

Ein häufig geäußerter Wunsch der Anwender ist die Möglichkeit des Remote-Zugriffs auf einzelne oder ausgewählte Aufnahmen (slabs/slices) eines archivierten Datensatzes. Dies wäre mit TAR-Containern kaum vernünftig zu realisieren. HDF5 wiederum bietet exzellente Möglichkeiten dies mittels der H5WS-Schnittstelle zu realisieren. H5WS, HDF5 Web Services (8), bietet eine API die explizit für das Arbeiten mit remote verfügbaren Daten realisiert wurde. H5WS bildet zumindest teilweise die Funktionalitäten der HDF5 Lite API ab und ermöglicht es Nutzern direkt die (Meta-)Daten zu durchsuchen und remote zu analysieren. Die Implementierung basiert interessanterweise auf der Client-Server Infrastruktur von Globus, sowie gridFTP und HDF5, und wird zum Beispiel von Fusion Grid (9) als Grid-Service angeboten. Eine Möglichkeit dies zu nutzen wird sicher zukünftig untersucht werden.

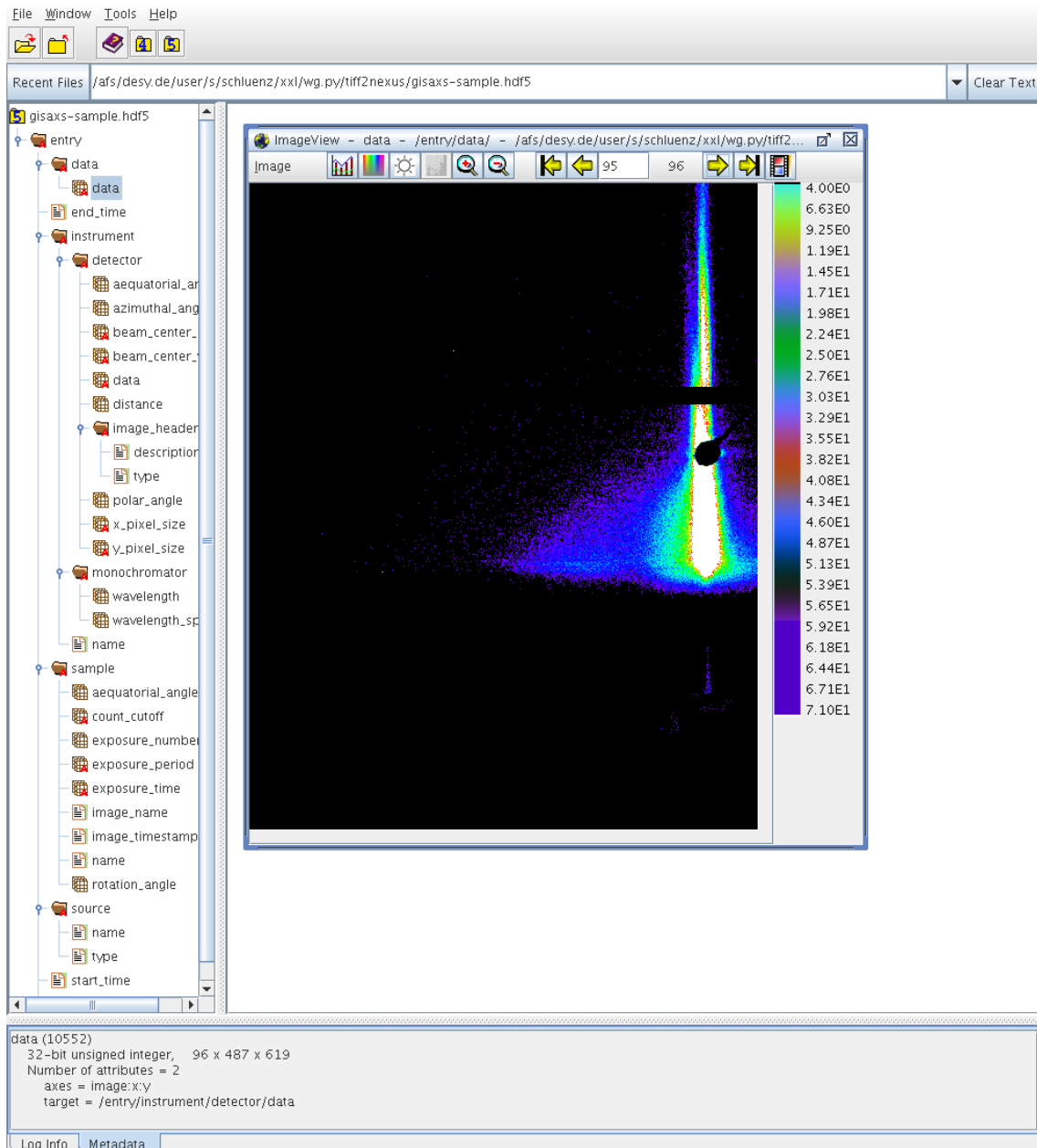


Abbildung 2: Visualisierung der Nexus-Datei mit hdfview

7 Hintergrundinformationen

7.1 Das Bilddateiformat TIFF (Tagged Image File Format)

TIFF ist ein Rasterbildformat und wurde als geräteunabhängiges Austauschformat von Aldus (jetzt Adobe) entwickelt. Die erste Version wurde 1986 publiziert, die aktuelle Version 6 (2) stammt aus dem Jahr 1992. Weite Verbreitung hat es als Ausgabeformat von Geräten wie Scannern und Fotoapparaten und als Ein- und Ausgabeformat von Programmen z.B. zur Fotobearbeitung gefunden. Da TIFF eines von wenigen Bilddateiformaten ist, die auch die unkomprimierte Speicherung von Bilddaten zulassen, wurde TIFF lange Zeit als Format der Wahl für die Speicherung der Archivversion von Bildern angesehen (10).

TIFF-Dateien sind gegliedert in einen Header und ein oder mehrere Image File Directories (IFDs). Die IFDs enthalten Verzeichniseinträge, die wiederum aus Tags bestehen, die die Daten enthalten. Die Möglichkeit, den Datentyp für jeden Tag individuell festzulegen, hat sicherlich mit zur weiten Verbreitung von TIFF beigetragen. Die hierarchische Struktur gestattet die Speicherung mehrerer Bilder in einer TIFF-Datei. Der Detektor Pilatus 300K schreibt nur ein Bild in jede TIFF-Datei.

Der TIFF-Standard (2) erlaubt die Byte-Reihenfolge little endian oder big endian. Beim Abspeichern einer Zahl in big endian wird das Byte mit den höchstwertigen Bits zuerst geschrieben. Bei little endian ist es genau umgekehrt: Das Byte mit den niedrigstwertigen Bits wird zuerst geschrieben. Die Ausgabedateien des Detektors Pilatus 300K besitzen die Byte-Reihenfolge little endian.

Eine TIFF-Datei beginnt mit einem 8-Byte-Header, in dem die Byte-Reihenfolge, die Zahl 42 als Kennung für TIFF und der Offset bis zum ersten IFD codiert sind. Der Header kann durch entsprechende Setzung des Offsets erweitert werden und Zusatzinformationen aufnehmen. Davon macht der Pilatus 300K Gebrauch. Dessen TIFF-Header sind auf eine Länge von 4096 Bytes eingestellt.

7.2 Das Datenformat NeXus

NeXus ist ein Standardformat für die Röntgen-, Neutronen- und Myonenphysik. Der Standard (3) sieht eine hierarchische Gliederung in *Datengruppen* vor. Auf der obersten Ebene befindet sich die Gruppe *NXentry*. Eine Gruppe kann Untergruppen oder Datenfelder enthalten. Datengruppen und -felder können mit Datenattributen versehen werden.

NeXus kann in Form zweier Dateiformate vorliegen, im Textformat als XML (eXtended Markup Language) und binär als HDF5 (Hierarchical Data Format 5) (5). HDF5 ist ein Datenformat zur effizienten Speicherung von Mess- und anderen wissenschaftlichen Daten. HDF5 ist auch eine Technologie für das Management sehr großer und komplexer Datenzusammenstellungen, und es wird entsprechend leistungsfähige Software dafür angeboten.

Literaturverzeichnis

- [1] Deutsches Elektronen-Synchrotron (DESY), <http://www.desy.de/>
- [2] TIFF 6.0 Spezifikation,
<http://partners.adobe.com/public/developer/en/tiff/TIFF6.pdf>
- [3] NeXus, <http://www.nexusformat.org/>
- [4] Python, <http://www.python.org/>
- [5] Hierarchical Data Format 5, <http://www.hdfgroup.org/HDF5/>
- [6] ICAT, <http://code.google.com/p/icatproject/wiki/IcatMain>
- [7] TOPCAT, <http://code.google.com/p/topcat/>
- [8] HDF5 Web Service, <http://www.hdfgroup.org/tools5desc.html>
- [9] fusiongrid, <http://www.fusiongrid.org>
- [10] M. Enders, Bilddokumente, in:
H. Neuroth, A. Oßwald, R. Scheffel, S. Strathmann, K. Huth,
nestor Handbuch, Eine kleine Enzyklopädie der digitalen Langzeitarchivierung,
Version 2.3
<http://nbn-resolving.de/urn:nbn:de:0008-2010071949>