



Narodowe Centrum Badań Jądrowych









Formaty plików ECAT, InterFile i DICOM





Rafał Masełek

Plan prezentacji








ECAT

-  użycie
-  wersje
-  output z Gate
-  struktura
-  main header & sub-header
-  podsumowanie

InterFile

-  ogólnie
-  header
-  kodowanie obrazu
-  podsumowanie

DICOM

-  ogólnie
-  struktura
-  struktura Data Element
-  rodzaje Data Elements
-  zawiłości
-  przechowywanie obrazów
-  podsumowanie

ECAT – użycie

Format ECAT został stworzony na potrzeby skanerów PET produkowanych przez amerykańską firmę CTI we współpracy z Siemensem (ok. 10 lat temu doszło do wchłonięcia CTI przez Siemens). Skanery zyskały sobie dużą popularność w USA i UK. Ostatni skaner z serii został zaprezentowany w 2002 roku. Późniejsze skanery Siemens korzystały ze standardu DICOM.

- połowa lat 90 – ECAT ART i ECAT 951
- 1998 – ECAT Exact 47
- 2000 – ECAT Exact HR+
- 2002 – ECAT Accel
- ok 2013 – Scintron Upgrade: upgrade hardware obliczeniowego i oprogramowania; przejście na DICOM

ECAT – wersje

Starsze skanery używały ECAT 6, od HR+ domyślna jest wersja 7. ECAT 7 różni się od poprzedniej wersji niewielkimi zmianami w nagłówku (header) i tym, że w wersja 7 wspiera obrazowanie 3D (jedna macierz na objętość a nie na płaszczyznę).

	ECAT 6	ECAT 7
attenuation file	.atn	.a/.A
image/volume	.img	.v/.V
sinogram normalization	.nrm	.n/.N
sinogram	.scn	.s/.S

Rozszerzenia z małymi literami oznaczają dane, z wielkimi „database entry” (?)

Przykładowy plik .v przedstawiający gradient można pobrać [tutaj](#).

Gate generuje pliki .S
Nie udało mi się ich otworzyć, a jedynie podejrzeć header. Podobno STIR jest w stanie to zrobić ([link](#)).

Bibliotekę ECAT można pobrać [stąd](#). Kompilacja może sprawić problemy, ale jak się dostatecznie głęboko poszuka to można znaleźć sposoby na rozwiązanie problemów.
<http://gate-users.healthgrid.narkive.com/rrLH2Y3m/gate-ecat7-help>

Output z Gate

Możliwy output:

- ASCII
- binary
- ROOT
- Interfile
- Sinogram
- **ECAT 7**
- LMF
- Image CT

Żeby dostać output z Gate w postaci plików .S należy:

- ustawić *system* na *ecat* co pociąga za sobą trzypoziomową strukturę logiczną detektora
- włączyć output *sinogram*
- włączyć output *ecat7*

Jest jedno makro dostarczane wraz z Gate v 7.2, które ma zaimplementowany system *ecat* i generuje pliki *yourSinogram.S*
.../examples/example_PET/PET_Ecat_System.mac

ECAT – struktura

Plik obrazu ECAT składa się z:

- *main header*
- przynajmniej jednej *matrix list (mlist)*

ECAT traktuje pamięć blokowo po 512 bajtów, pierwszy blok zaczyna się dla bajtu 0, następny dla bajtu 512 itd. Z tego powodu nagłówek nie da się rozszerzyć.

Matrix list – tablica z jednym rzędem przypadającym na klatkę (? *frame*) danych.

Kolumny w matrix list:

- 0 – matrix identifier (frame number)
- 1 – matrix data start block number (subheader followed by image data)
- 2 – last block number of matrix (image) data
- 3 – matrix status:
 - 1 - exists - rw
 - 2 - exists - ro
 - 3 - matrix deleted

Na każdą klatkę (matrix) przypada jeden *sub-header* (zwany także image-header). Sub-header to pojedynczy blok 512 bajtów po którym następują dane obrazu.

ECAT – struktura

Obrazy w ECAT są zapisane jako dane RAW BINARY (macierze integerów).
Z metadanych możemy dowiedzieć się ile bajtów użyto do zapisania jednego piksela:

file_type = 7 (Volume-16) (możemy się domyslić, że 2 bajty, ale czasami jest tylko numer typu i wtedy nie wiemy, musimy próbować. Nie mamy też informacji o tym, czy są to signed czy unsigned integery. Czasem można zgadywać na podstawie:

image_min = 0

image_max = 255

W metadanych możemy znaleźć informacje o liczbie obrazów:

num_planes = 10

num_frames = 1

oraz o wymiarach obrazów:

num_dimensions = 3

x_dimension = 512

y_dimension = 512

z_dimension = 10

Z moich prób dekodowania wynika, że ECAT stosuje BIG ENDIAN.

<— Taki plik ma następującą strukturę:

- main header — 512B
- matrix list — 512B (wspólny dla wszystkich frame)
- subheader dla jedynego frame — 512B
- dane frame (10 planes po kolei)

ECAT7 – main header

--- Main Header ----

magic_number = MATRIX70v

original_file_name =

software_version = 70

scanner_model = 0

file_type = 7 (Volume-16)

serial_number =

scan_start_time = 0

isotope_name =

isotope_half_life = 0.0

radiopharmaceutical =

gantry_tilt = 0.0

gantry_rotation = 0.0

bed_elevation = 0.0

intrinsic_tilt = 0.0

wobble_speed = 0

tx_source_type = 0

distance_scanned = 0.0

transaxial_fov = 0.0

angular_compression = 0

coin_sample_mode = 0 (Net trues)

axial_sample_mode = 0

ecat_calib_factor = 1.0

calib_units = 2

calib_units_label = 0

compression_code = 0

study_type =

patient_id =

patient_name = MPI-Tool

patient_sex =

patient_dexterity =

dosage = 0.0

well_counter_corr_factor = 0.0

data_units = Total counts

septa_state = 0

reserved = [0, 0, 0, 0, 0, 0]

patient_age = 0.0

patient_height = 0.0

patient_weight = 0.0

patient_birth_date = 0

physician_name =

operator_name =

study_description =

acquisition_type = 0 (Undefined)

patient_orientation = 8

facility_name =

num_planes = 10

num_frames = 1

num_gates = 1

num_bed_positions = 0

init_bed_position = 0.0

bed_position_0 = 0.0

bed_position_1 = 0.0

bed_position_2 = 0.0

bed_position_3 = 0.0

bed_position_4 = 0.0

bed_position_5 = 0.0

bed_position_6 = 0.0

bed_position_7 = 0.0

bed_position_8 = 0.0

bed_position_9 = 0.0

bed_position_10 = 0.0

bed_position_11 = 0.0

bed_position_12 = 0.0

bed_position_13 = 0.0

bed_position_14 = 0.0

plane_separation = 0.10000000149

lower_scatter_threshold = 0

lower_true_threshold = 0

upper_true_threshold = 0

user_process_code =

acquisition_mode = 0

bin_size = 0.0

branching_fraction = 0.0

dose_start_time = 0

ECAT – podsumowanie

- Format związany z linią skanerów Siemens
- Przestarzały i nierozwijany
- Brak dostępnej specyfikacji (potrzeba instrukcji do urządzenia lub kontaktować się z producentem)
- Mała ilość oprogramowania wspierającego format; jeżeli już to w ograniczonym zakresie

ECAT – software (część, reszta w backendzie)

- (X)MedCon – małe ale potężne narzędzi do konwersji pomiędzy formatami: raw ASCII, raw binary, Acr/Nema, Analzye, Concorde/uPET, DICOM, ECAT6/7, gif, InterFile, INW, NIfTI, png
- narzędzia ze strony Turku PET Centre
- biblioteka w pythonie NiBabel do obsługi plików obrazowania medycznego
- program LONI Inspector do przeglądania metadanych

InterFile – ogólnie

Format InterFile powstał w 1980 roku jako próba ustandaryzowania formatów plików do obrazowania medycznego. Był popularny i używany przez wiele lat. Został wyparty przez inne formaty, głównie DICOM.

InterFile składa się z dwóch plików:

- pliku nagłówkowego ASCII z rozszerzeniem .h33/.hdr
- binarnego pliku z obrazem, z rozszerzeniem .i33/.img (lub .sin dla projekcji)

Wyróżnikiem formatu InterFile jest istnienie osobnego pliku z nagłówkiem kodowanego w ASCII. Wpisy w nagłówku mają postać par klucz-wartość, np:

!number of bytes per pixel := 3

Wykrzyknik oznacza, że token jest opcjonalny. Listę tokenów można znaleźć w oficjalnej specyfikacji wersji 3.3 dostępnej [tutaj](#).

Taka struktura pozwala łatwo edytować i rozszerzać header. Tak zmodyfikowany plik można następnie wyeksportować do innego, trudniejszego w edycji formatu.

Jest jedno makro dostarczane wraz z Gate v 7.2,
które ma zaimplementowany system SPECT (konieczny dla
InterFile) i generuje pliki InterFile z rozszerzeniami .hdr i .sin
.../examples/example_SPECT/SPECT.mac

InterFile – nagłówek

```

!INTERFILE :=
!imaging modality := nucmed
!originating system := ECAT0
!version of keys := 3.3
date of keys := 1996:09:24
conversion program := (X)MedCon
program author := Erik Nolf
program version := 0.14.1
program date := 2015:12:26
;
!GENERAL DATA :=
original institution :=
!data offset in bytes := 0
!name of data file := m000-gradient-512x512x10.i33
patient name := MPI-Tool
!patient ID :=
patient dob := 1970:01:01
patient sex := M
!study ID :=
exam type := Unknown
data compression := none
data encode := none
organ := Unknown
isotope := /
dose := 0
NUD/Patient Weight [kg] := 0.00
NUD/imaging modality := PT
NUD/activity := 0
NUD/activity start time := 00:00:00
NUD/isotope half life [hours] := 0.000000
;

```

```

!GENERAL IMAGE DATA :=
!type of data := Static
!total number of images := 10
study date := 1970:01:01
study time := 01:00:00
imagedata byte order := LITTLEENDIAN
process label :=
;
number of energy windows := 1
;
energy window [1] :=
energy window lower level [1] :=
energy window upper level [1] :=
flood corrected := N
decay corrected := N
;
!STATIC STUDY (General) :=
number of images/energy window := 10
;
!Static Study (each frame) :=
!image number := 1
!matrix size [1] := 512
!matrix size [2] := 512
!number format := signed integer
!number of bytes per pixel := 2
scaling factor (mm/pixel) [1] := +1.000000e+00
scaling factor (mm/pixel) [2] := +1.000000e+00
image duration (sec) := 0.000000e+00
image start time := 00:00:00
label := Unknown
!maximum pixel count := +2.550000e+02
!minimum pixel count := +0.000000e+00
total counts := 0
;
; TU SA POZOSTALE KLATKI
!END OF INTERFILE :=

```

InterFile – kodowanie obrazu

W InterFile obrazy/sinogramy są zapisywane w osobnym pliku jako RAW BINARY. Kolejne wartości pikseli są kodowane po sobie jako macierz. Jeżeli jest wiele obrazów/projekcji to kodowane są jeden po drugim. Parametry kodowania określone są w headerze:

!data offset in bytes := 0
data compression := none
!total number of images := 10
imagedata byte order := LITTLEENDIAN
!matrix size [1] := 512
!matrix size [2] := 512
!number format := signed integer
!number of bytes per pixel := 2

Powyższy przykład opisuje kodowanie 10 obrazków o wymiarach 512 na 512 pikseli. Każdy piksel jest kodowany za pomocą dwubajtowego integera ze znakiem (int16). Kolejność kodowania bitów to LITTLE ENDIAN (najmniej znaczący bit jako pierwszy). Nie ma żadnego offsetu i brak kompresji.

InterFile – podsumowanie

- Przystarzały i nierozwijany format.
- Dostępna szczegółowa specyfikacja.
- Rozdzielenie metadanych od obrazu.
- Prosty do edycji i odczytu header, można go bardzo łatwo rozbudować o nowe pola.
- Możliwy output z Gate*.
- Można go użyć do łatwej edycji metadanych i eksportu do DICOM.

DICOM – ogólnie

DICOM (ang. Digital Imaging and Communications in Medicine) – norma opracowana przez American College of Radiology (ACR) i National Electrical Manufacturers Association (NEMA) jako próba ujednolicenia i ustandaryzowania przechowywania i wymiany danych z obrazowań medycznych. DICOM to nie tylko format plików ale również protokół transmisji danych i cały abstrakcyjny model danych. DICOM odniósł bardzo duży sukces i stał się powszechnie obowiązującym standardem.

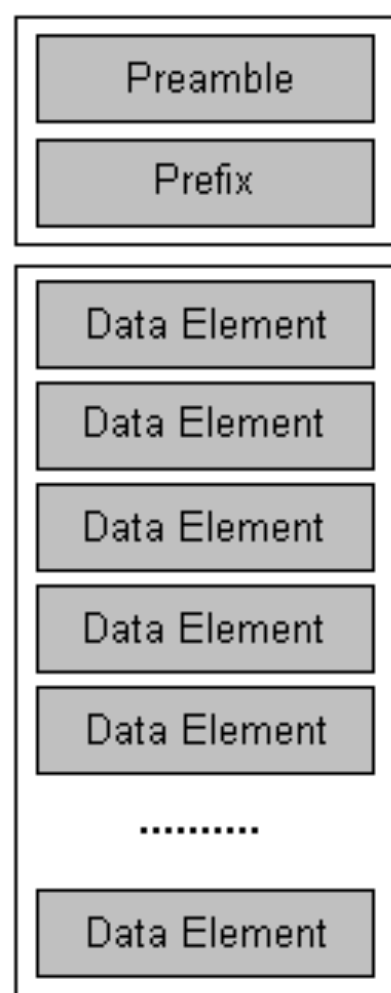
Charakterystyczne dla DICOM jest połączenie metadanych i obrazu w jeden plik (najczęściej z rozszerzeniem **.dcm** ale zdarzają się również inne lub brak rozszerzenia). Pliki DICOM posiadają niezwykle rozbudowane metadane, które mogą zawierać informacje o pacjencie, lekarzu, parametrach sprzętowych, warunkach pomiaru itd.

Standard DICOM (obejmujący również protokół transmisji) posiada bardzo rozbudowaną (setki stron!) dokumentację opisującą w bardzo dokładny i formalny sposób wszelkie aspekty związane z odwzorowywaniem rzeczywistości, przechowywaniem, wyświetlaniem i transmisją danych. Pełną dokumentację można znaleźć na stronie [standardu](#), a króciutki wstęp do najważniejszych zagadnień na stronie [LeadTools](#).

Przykładowe pliki do pobrania [tutaj](#) (na dole strony).

DICOM – struktura

Pliki DICOM składają się z dwóch części:



● header

Header

- **preamble** (128 bytes) – preambuła może zawierać specjalne dane dla aplikacji czytających pliki DICOM. Jej użycie jest opcjonalne. W przypadku nieużywania, wszystkie bajty należy ustawić na 00H.
- **prefix** (4 bytes) – składa się z czterech wielkich liter ,D' ,I' ,C' ,M' zakodowanych zgodnie z ISO 8859 G0.

Data Set

● Data Set – każdy plik zawiera jeden Data Set

- **Data Element** – Data Set zawiera wiele Data Elementów.
- **Data Element**
- ...

DICOM – struktura Data Element

Data Element (DE) ma 4 podstawowe części składowe:

- **DE Tag** – uporządkowana para **uint16_t**; numer grupy po którym następuje numer elementu. Jest to **unikalny** identyfikator DE. Tagi powinny być ułożone rosnąco względem grup i względem elementów w grupie.
- **Value Representation (VR)** – dwubitowy character string. Określa typ Data Element.
- **Value Length** – **uint16_t** lub **uint32_t** w zależności od VR. Określa długość (liczbę bajtów) wartości DE.
- **Value Field** – parzysta ilość bajtów zawierająca wartość Data Element.

DICOM – rodzaje Data Elements

- **Type 1 Required DE** – muszą być obecne w pliku, mieć poprawną wartość i niezerową długość.
- **Type 1C Conditional DE** – jeżeli pewne warunki są spełnione, to muszą znaleźć się w pliku. Muszą mieć wtedy poprawną wartość i niezerową długość.
- **Type 2 Required DE** – muszą być obecne w pliku, ale mogą nie mieć wartości i mieć długość zero.
- **Type 2C Conditional DE** – jeżeli pewne warunki są spełnione, to muszą znaleźć się w pliku. Mogą nie mieć wartości i mieć długość zero.
- **Type 3 Optional DE** – są opcjonalne, mogą nie mieć wartości i mieć długość zero.

DICOM – zawiłości

DICOM jest formatem bardzo rozbudowanym, posiadającym pewne zawiłości utrudniające jego obsługę.

- VR — Podstawowym problemem jest to, że VR może być podane jawnie (explicit) bądź niejawnie (implicit). W drugim przypadku Data Element nie zawiera VR. Co więcej, niektóre szczególne VR wymagają rezerwacji dodatkowego miejsca po dwubajtowym kodzie.
- Value Length — to, czy użyte jest 16 czy 32 bajty zależy od VR oraz od tego, czy jest ono jawnie użyte. Dla niektórych VR możliwe jest użycie nieokreślonej długości FFFFFFFFH (32 bity).
- Value — musi składać się z parzystej ilości bajtów. Czasami konieczne jest dopełnienie aby liczba bajtów była parzysta, np. w przypadku łańcucha znaków. W zależności od VR używa się spacji bądź NULL.
- Istnieje specjalne Value Representation **QS**, które rozpoczyna sekwencję elementów. Może to być użyte np. do tworzenia prostych struktur powtarzających się elementów.
- Konieczność użycia unikalnych UID.
- ...

DICOM – przechowywanie obrazów

DICOM posiada wprawdzie jeden Data Set ale można w nim przechowywać obrazy wieloklatkowe. Obrazy są przechowywane za pomocą Data Elements:

- **Pixel Data** (7FE0, 0010) – przechowuje wartości poszczególnych bajtów obrazu lub wskaźnik na sekwencję z danymi.
- **Rows** (0028, 0010) and **Columns** (0028, 0011) – definiuje rozmiary obrazu
- **Samples per Pixel** (0028, 0002) – liczba kanałów z kolorami (1 dla skali szarości, 3 dla RGB itd.)
- **Photometric Interpretation** (0028, 0004) – definiuje zastosowanie kanałów (np. ,RGB' albo ,MONOCHROME1')
- **Planar Configuration** (0028, 0006) – określa ułożenie danych z poszczególnych kanałów w buforze.
- **Bits Allocated** (0028, 0100) – określa rozmiar komórki pikselowej
- **Bits Stored** (0028, 0101) – określa ile bitów w komórce jest wykorzystanych na obraz
- **High Bit** (0028, 0102) – określa jak ułożone są bity wykorzystywane na obraz (bits stored) pośród wszystkich (bits allocated)

DICOM – podsumowanie

- Ogólnie przyjęty i rozwijany standard wspierany przez większość producentów sprzętu i oprogramowania
- Dostępna bardzo szczegółowa dokumentacja
- Bardzo rozbudowany format, przez co nieproste zapisywanie i trudny odczyt
- Nadaje się do przechowywania wielu informacji dodatkowych (metadanych)
- Bardzo dużo programów do czytania, edycji i konwersji; zarówno darmowych jak i komercyjnych. (ale najczęściej częściowe wsparcie).



Narodowe Centrum Badań Jądrowych



Dziękuję za uwagę!

źródła i programy dla ECAT

- <http://nipy.org/nibabel/reference/nibabel.ecat.html>
- http://wiki.opengatecollaboration.org/index.php/Users_Guide_V7.2:Data_output#Sinogram_output
- http://stir.sourceforge.net/documentation/doxy/html/namespacestir_1_1ecat.html
- http://www.turkupetcentre.net/petanalysis/format_image_ecat.html
- <http://www.nf.mpg.de/vinci3/doc/image-formats.html>
- <http://manpages.ubuntu.com/manpages/xenial/man4/m-ecat.4.html>
- http://www.turkupetcentre.net/analysis/doc/dicom2ecat7_usermanual.htm
- <http://www.dalcanton.it/tito/pyecat/>
- http://wiki.opengatecollaboration.org/index.php/Users_Guide_V7.2:Data_output
- <http://www.turkupetcentre.net/petanalysis/sinogram.html>
- http://hendrix.ei.dtu.dk/staff/fnielsen/bib/Nielsen2001BibNeuroinformatics/File_formats.html
- <http://rpm.pbone.net/index.php3/stat/45/idpl/17881406/numer/1/nazwa/medcon>
- <https://info.blockimaging.com/bid/100002/Siemens-ECAT-PET-Scanners-Compared>
- <https://imagej.nih.gov/ij/features.html>
- <https://www.openmicroscopy.org/site/support/bio-formats5.4/formats/ecat7.html>
- http://www.turkupetcentre.net/petanalysis/format_list.html
- <http://www.mieamerica.com/index.php?id=1250&L=4>

Dokumentacja InterFile

https://www.researchgate.net/publication/21707960_A_file_format_for_the_exchange_of_nuclear_medicine_image_data_A_specification_of_interfile_version_33

Źródła i programy dla DICOM

- <https://www.lifewire.com/dicom-file-2620657>
- <https://www.leadtools.com/sdk/medical/dicom-spec1>
- <http://www.microdicom.com/downloads.html>
- <http://www.dicomlibrary.com/>
- <http://www.aliza-dicom-viewer.com/download>
- dicom.nema.org
- <https://idoimaging.com/home>
- <http://support.dcmstk.org/docs/dump2dcm.html>
- <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3948928/>
- <http://dicomiseasy.blogspot.co.il>

Do biblioteki pydicom:

- https://github.com/darcymason/pydicom/blob/master/docs/getting_started.rst#id6
- <http://dicomiseasy.blogspot.com/2012/08/chapter-12-pixel-data.html>
- <http://stackoverflow.com/questions/14350675/create-pydicom-file-from-numpy-array>