

Wieloplatformowa przeglądarka obrazów DICOM w C++

Adam Jędrzejowski

24 czerwca 2019

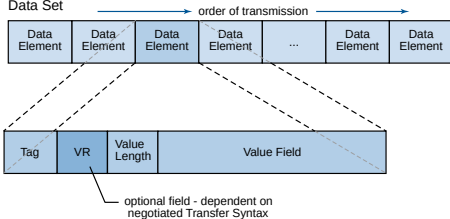
Obrazowe techniki medyczne

- Radiografia — RTG
- Tomografia komputerowa — CT
- Obrazowanie metodą rezonansu magnetycznego — MRI
- Ultrasonografia
- Scyntygrafia
- Tomografia SPECT
- Tomografia PET

Standard DICOM

Nazwa	Identyfikator	Typ danych	Opis
SpecificCharacterSet	(0008,0005)	CS	Używana specyfikacja
InstitutionName	(0008,0080)	LO	Miejsce wykonywania badania
Manufacturer	(0008,0070)	LO	Producent aplikacji
StationName	(0008,1010)	SH	Nazwa urządzenia
PatientID	(0010,0020)	LO	Identyfikator pacjenta
PatientsName	(0010,0010)	PN	Nazwisko pacjenta
PatientsBirthDate	(0010,0030)	DA	Data urodzin pacjenta
PatientsSex	(0010,0040)	CS	Płeć pacjenta
PatientsAge	(0010,1010)	AS	Wiek pacjenta
BodyPartExamined	(0018,0015)	CS	Badana część ciała
StudyDate	(0008,0020)	DA	Data badania
PhotometricInterpretation	(0028,0004)	CS	Format zapisu obrazu
Rows	(0028,0010)	US	Wysokość zdjęcia
Columns	(0028,0011)	CS	Szerokość zdjęcia

Data Set



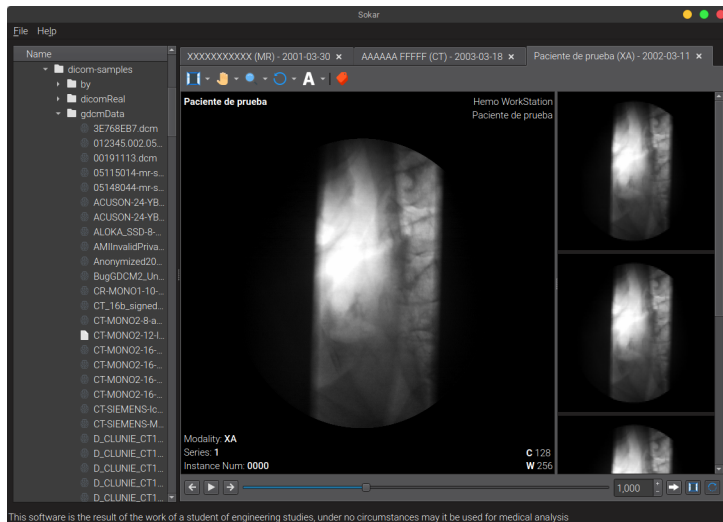
Cel pracy i założenia

CMake to wieloplatformowe narzędzie do automatycznego zarządzania procesem kompilacji programu. Jest to niezależne od kompilatora narzędzie pozwalające napisać jeden plik, z którego można wygenerować odpowiednie pliki budowania dla dowolnej platformy.

Qt jest zbiorem bibliotek i narzędzi programistycznych dedykowanych dla języków C++, QML i Java. Qt posiada bibliotekę do tworzenia interfejsu graficznego, oraz wiele innych rozwiązań ułatwiających programowanie obiektowe i zdarzeniowe. Posiadanych normy: IEC 62304:2015, IEC 61508:2010-3 7.4.4, ISO 9001:2015. Posiada systemy rodzicielstwa i sygnałów.

GDCM to biblioteka do obsługi standardu DICOM. Posiada możliwość wczytywania plików z dysku jak i z lokalizacji sieciowych oraz wczytywania plików DICOMDIR. Ma wbudowaną dekompresję obrazów i obsługi różnych kodowań tekstu.

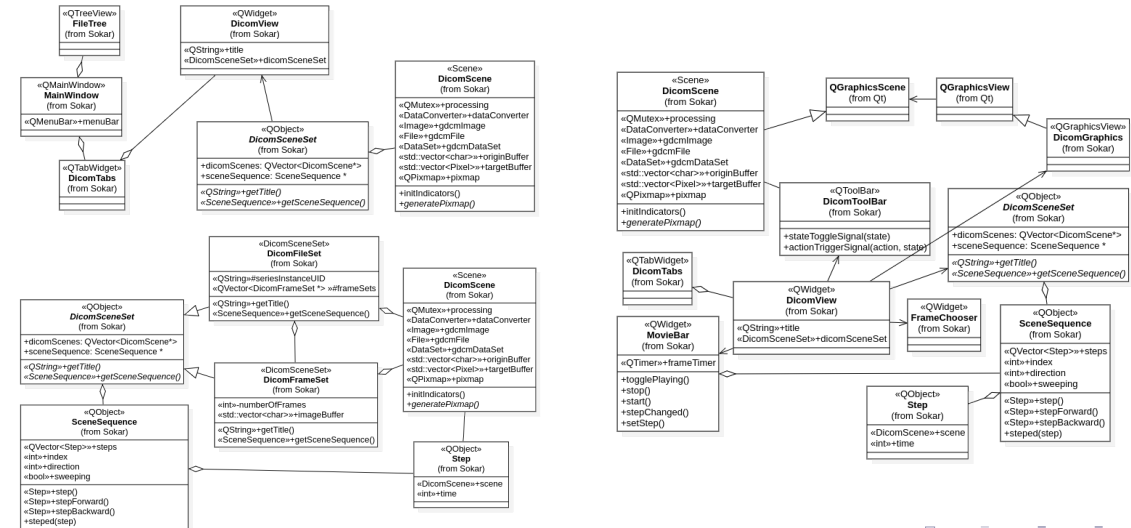
Projekt interfejsu graficznego



Projekt interfejsu graficznego - scena



Budowa obiektowa programu

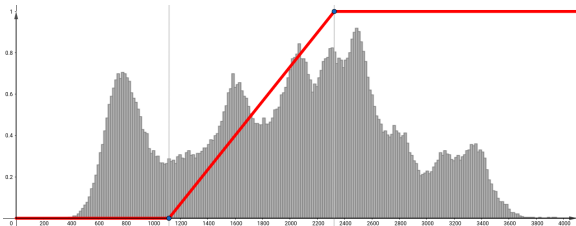


Dekoder DICOM

char*	8b ₁	8b ₂	8b ₃	8b ₄	8b ₅	8b ₆	8b ₇	8b ₈	8b ₉	8b ₁₀	8b ₁₁	8b ₁₂	8b ₁₃	8b ₁₄	8b ₁₅	8b ₁₆
int8*	8b ₁	8b ₂	8b ₃	8b ₄	8b ₅	8b ₆	8b ₇	8b ₈	8b ₉	8b ₁₀	8b ₁₁	8b ₁₂	8b ₁₃	8b ₁₄	8b ₁₅	8b ₁₆
int16*	16b ₁		16b ₂		16b ₃		16b ₄		16b ₅		16b ₆		16b ₇		16b ₈	
int32*	32b ₁				32b ₂				32b ₃				32b ₄			
int64*	64b ₁								64b ₂							

auto origin = (quint16 *) &originBuffer[0];

Okienkowanie



Standard DICOM przewiduje, że wszystkie dane powinny być wyskalowane za pomocą wzoru:

$$\text{OutputUnits} = m * SV + b$$

- m — wartość z RescaleSlope (0x0028, 0x1053),
- b — wartość z RescaleIntercept (0x0028, 0x1052),
- SV — stored values — wartość woksela z pliku,
- OutputUnits — wartość wynikowa.

Implementacja

$$x_0 = \text{center} - \text{width}/2 \quad y_0 = 1.0$$

$$x_1 = \text{center} + \text{width}/2 \quad y_1 = 0.0$$

$$(\text{OutputUnits} - b)/m = SV$$

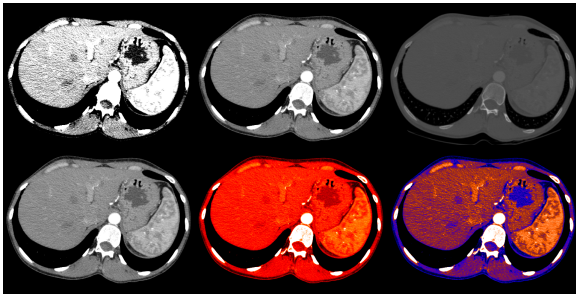
$$x_0 = \text{rescaleIntercept} \quad x_0/ = \text{rescaleSlope}$$

$$x_1 = \text{rescaleIntercept} \quad x_1/ = \text{rescaleSlope}$$

$$a = (y_1 - y_0)/(x_1 - x_0) \quad b = y_1 - a * x_1$$

Tablica LUT

8b	12b	16b	32b	64b
768B	196kB	196kB	12, 5GB	55 * 10 ⁶ TB



Orientacja pacjenta

Definicja standardu DICOM

$$\begin{bmatrix} P_x \\ P_y \\ P_z \\ 1 \end{bmatrix} = \begin{bmatrix} X_x \Delta_i & Y_x \Delta_j & 0 & S_x \\ X_y \Delta_i & Y_y \Delta_j & 0 & S_y \\ X_z \Delta_i & Y_z \Delta_j & 0 & S_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} i \\ j \\ 0 \\ 1 \end{bmatrix} = M \begin{bmatrix} i \\ j \\ 0 \\ 1 \end{bmatrix}$$

gdzie:

- P_{xyz} — koordynaty wksela (i,j) we współrzędnych obrazu wyrażone w milimetrach,
- S_{xyz} — trzy wartości z elementu ze znacznikiem Image Position (0x0020, 0x0032). Oznacza on punkt pozycji pacjenta wyrażony w milimetrach w stosunku do urządzenia wykonującego pomiar.
- X_{xyz} — trzy pierwsze wartości ze znacznika Image Orientation (0x0020, 0x0037),
- Y_{xyz} — trzy ostatnie wartości ze znacznika Image Orientation (0x0020, 0x0037),
- i i j — oznaczają współrzędne na macierzy obrazu, odpowiednio kolumnę i wiersz. Zero oznacza początek,
- Δ_i i Δ_j — rzeczywista wielkość piksela obrazu wyrażona w milimetrach. W algorytmie wyznaczania strony pacjenta ta wartość może wynosić 1, ponieważ odpowiada za skalę.

Implementacja

$$PatientPosition = imgMatrix * ScenePosition$$

$$imgMatrix^{-1} * PatientPosition = imgMatrix^{-1} * imgMatrix * ScenePosition$$

$$imgMatrix^{-1} * PatientPosition = ScenePosition$$

$$ScenePosition = imgMatrix^{-1} * PatientPosition$$

gdzie:

- $imgMatrix$ — macierz przekształcenia obrazu, o której będzie później opisana,
- $ScenePosition$ — pozycja na obrazie, która nas interesuje,
- $PatientPosition$ — jeden z punktów względem pacjenta.

$$rotateTransform * \begin{pmatrix} X_x & Y_x & 0 & 0 \\ X_y & Y_y & 0 & 0 \\ X_z & Y_z & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} * PatientPosition$$

Funckje przeglądarki

