Symulator tomografu komputerowego

Marek Subocz 136806

02.04.2020r.

1 Zastosowane narzędzia

- **rodzaj tomografu:** stożkowy
- **język programowania:** Python3

1.1 Użyte biblioteki Pythonowe

- matplotlib
- numpy
- tqdm
- ipywidgets

2 Opis głównych funkcji programu

2.1 pozyskiwanie odczytów dla poszczególnych detektorów

Za pomocą algorytmu Bresenhama wyznaczam piksele uczestniczące w obliczaniu sumy dla każdego detektora i zapisuję to na transformacie Radona

```
emiter = [int(center[0] + r * cos(alpha)), int(center[1] - r * sin(alpha))]
detectors = [
            (int(center[0] + r * cos(alpha + pi - ro / 2 + i * ro / (n_det - 1))),
             int(center[0] - r * sin(alpha + pi - ro / 2 + i * ro / (n_det - 1))))
            for i in range(n_det)]

# Radon
chosen_lines = [
np.clip(BresenhamLine(*emiter, *detector),0, img.shape[0] - 1)
for detector in detectors
]

lines = [
list(map(lambda p: img[p[0]][p[1]], line))
for line in chosen_lines
]
```

2.2 ustalanie jasności poszczególnych punktów obrazu wynikowego oraz jego przetwarzanie końcowe

Na koniec algorytmu odejmuje od wszystkich pikseli wartość najciemniejszego z nich, wartosci te są potem normalizowane

```
result_list = list(map(sum, lines))
radon[:, i_step] = result_list

# Inverse Radon
for i_line, line in enumerate(chosen_lines):
    for p in line:
        inv_radon[p[0]][p[1]] += result_list[i_line]
```

2.3 odczyt i zapis plików DICOM

Wynik obliczeń można zapisać do pliku DICOM, następnie program automatycznie odczytuje plik i pokazuje jego zawartość (wszystko jest już gotowe w notebooku).

```
def dicom_load(filename="data/test.dcm"):
    ds = pydicom.dcmread(filename)

    if "PatientName" in ds:
        print(f"PatientName: {ds.PatientName}")
    if "ImageComments" in ds:
        print(f"ImageComments: {ds.ImageComments}")
    if "StudyDate" in ds:
        print(f"StudyDate: {ds.StudyDate}")

    arr = ds.pixel_array
    img = Image.fromarray(np.uint8(arr * 255))
    return np.asarray(img, dtype="uint8")


def dicom_save(img, filename, patient, comments, date):
    ds = pydicom.dcmread(filename)
    ds.PatientName = patient
    ds.ImageComments = comments
    ds.StudyDate = date

    ds.PixelData = img.tobytes()
    ds.Rows, ds.Columns = img.shape
    ds.save_as(filename)
```