Sprawozdanie: Tomograf

Eryk Szpotański nr indeksu 136811

1 Skład grupy

Eryk Szpotański nr indeksu 136811

2 Zastosowany model tomografu

stożkowy addytywny

3 Zastosowany język programowania

transformata + algorytm Bresenhama: C (podstawowe biblioteki: stdio.h, math.h, stdlib.h, string.h, omp.h)

GUI: Python + Streamlit (bilbioteki: streamlit, ctypes, math, numpy, skimage, os, copy)

4 Opis głównych funkcji programu

pozyskiwanie odczytów:

```
for (unsigned int i = 0; i < width; ++i) {
        alpha = i * step;
        E = point_on_circle(C, alpha);

        beta = alpha - spread / 2 + M_PI;
        for (int j = 0; j < decod; ++j) {
                beta += step_beta;
                D = point_on_circle(C, beta);
                if (E.x <= D.x)
                        result[i][j] = line_bres(E.x, E.y, D.x, D.y);
                else
                        result[i][j] = line_bres(D.x, D.y, E.x, E.y);
        }
}
```

Gdzie alpha i beta to kąty (w radianach) odpowiadające pozycji emitera i aktualnego dekodera na wcześniej obliczonym okręgu C (opisanego na obrazie). 'step' to podana zmiana kąta (również w radianach) odpowiadająca jednemu krokowi, spread to podana rozpiętość emiterów, a decod

to podana ilość dekoderów. Width to ilość kroków (inaczej szerokość sinogramu), obliczane tutaj E oraz D to pozycje odpowiednio emitera oraz aktualnego dekodera, a wywoływana funkcja line_bres zwraca sumę poszczególnych piskeli wybranych przez algorytm Bresenhama. Podany kod odpowiada fragmentowi z pliku C/transform.c linii 464.

filtrowanie:

```
const float rev_pi = -(float) (M_2_PI * M_2_PI);
float filtered(unsigned int row, int col) {
        float sum = 0.0f;
        for (int i = -11; i <= 11; i += 2) {
                if (i + col >= 0 && i + col < result_b)
                        sum += (rev_pi / (float) (i * i))
                                * result[row][col + i];
        }
        sum += result[row][col];
        return sum;
}
```

Zmienna globalna result zawiera sinogram. Podany kod odpowiada fragmentowi z pliku C/transform.c linia 602.
Rozmiar maski wynosi 23.

przetwarzanie końcowe:

```
res = gaussian(res, sigma=1)
if fil:
        ma = np.amin(res)
        print(ma)
        res = res - ma
        ma = np.amax(res)
        print(ma)
        res = res / ma
else:
        ma = np.amax(res)
        print(ma)
        res = res / ma
```

'gaussian' jest funkcją z skimage, a fil przybiera wartość True, jeśli obraz jest filtrowany, a res to sam otrzymany obraz. Kod ten odpowiada fragmentowi z pliku Python/main.py linii 86.

5 Informacje dodatkowe

GUI należy uruchamiać z folderu Python komendą:

```
streamlit run gui.py
```

W innym przypadku program nie będzie mógł wczytać biblioteki utworzonej z programu napisanego w C (transf.so).