

Analiza Obrazów - Projekt

Karolina Kędzior

11 czerwca 2016

1 Wstęp

W ramach projektu należało zastosować operacje morfologiczne oraz inne, tak aby policzyć kości na zdjęciu poniżej.



Rysunek 1: Analizowane zdjęcie rentgenowskie dłoni

Prace wykonano w języku JAVA przy pomocy biblioteki KiMage. Do wyodrębnienia kształtu kości zostały wykorzystane następujące operacje:

- Erozja i Dylacja dla obrazów w odcieniach szarości,
- Prunning,
- Thinning,
- Sobel,
- Binaryzacja oparta na progowaniu globalnym,
- Component Labeling udostępniony w bibliotece KiMage.

Ich implementacje można znaleźć [tutaj](#)

2 Opis metody

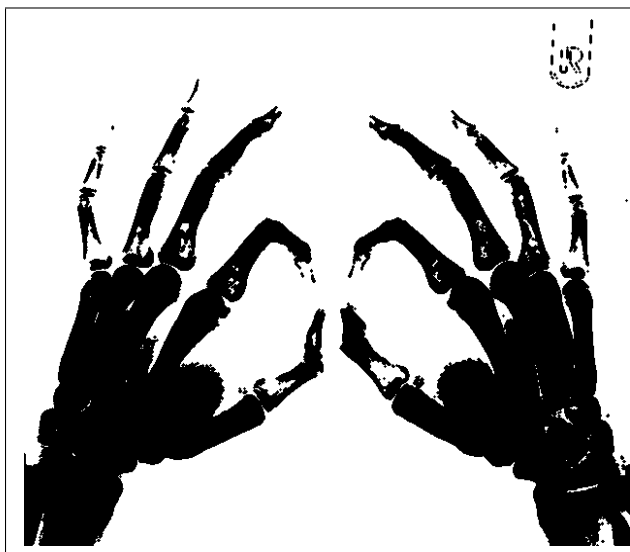
Cały proces można podzielić na 3 istotne etapy: wyodrębnieniu kości z całej dłoni, wydobycie krawędzi szkieletu oraz połączenie otrzymanych obrazów tak by otrzymać obraz końcowy z którego określamy spójne składowe.

a) Uzyskanie tła

Ogólny pomysł na rozwiązanie zadania opiera się na nałożeniu zarysu kości na czarne tło reprezentujące cień kości. Stąd konieczne jest wyodrębnienie tego cienia z dłoni ręki. Problem stanowi odpowiednie dobranie progu do binaryzacji.

Za pomocą Binaryzacji Otsu można uzyskać tylko całe dłonie, stąd własnoręcznie dobrałam parametry operacji. Po wielokrotnych próbach ustaliłam wartości progów: 125 i 220.

Ze względu ,iż niektóre fragmenty kości są w tych samych odcieniach szaro-



Rysunek 2: Końcowy wynik. Wyodrębnione kości z dłoni

ści co skóra, otrzymany wynik nie jest idealny (widoczne lekkie ubytki przy palcach). Aby zmniejszyć efekty operacji użyłam erozji.

```
exec2.add(new ImagePartByRange(), "min", 125, "max", 220);
exec2.add(new Erosion2(new CircularSE(3)));
```

Rysunek 3: Fragment kodu. Binarystacja oraz Erozyja z maską 3x3.

b) Wyodrębnienie krawędzi

Jest to zdecydowanie najtrudniejszy etap do zrealizowania. Mając już tło szkieletu potrzebujemy linii które podzieliłyby je na fragmenty odpowiadające poszczególnym kościom.

Zaczęłam od prostego wykrycia krawędzi metodą Sobel'a 5(a), poprzedzonego dwukrotną operacją Erozyji w celu zmniejszenia rozmiaru kości/zwiększając odległości między krawędziami. Następnie mamy binaryzację w celu pozbycia się zbędnych fragmentów 5(b). Mając ogólny zarys należy skupić się na domknięciu wszystkich linii, szczególnie przy małych kościach śródręcza. Aby połączyć cześć drobnych kresek naprzemiennie stosuje dylację, pocienianie i erozyję tak aby nie podzielić kości na zbyt małe fragmenty. Na końcu pozbywam się drobnych "wypustek" 5(d).

```
exec.add(new Erosion2(new CrossSE(3)));
exec.add(new Erosion2(new CrossSE(3)));
exec.add(new Sobel());

exec.add(new ImagePartByRange(), "min", 0, "max", 123);
exec.add(new Dilation2(new CrossSE(3)));
exec.add(new Dilation2(new CrossSE(3)));
exec.add(new Erosion2(new CrossSE(3)));

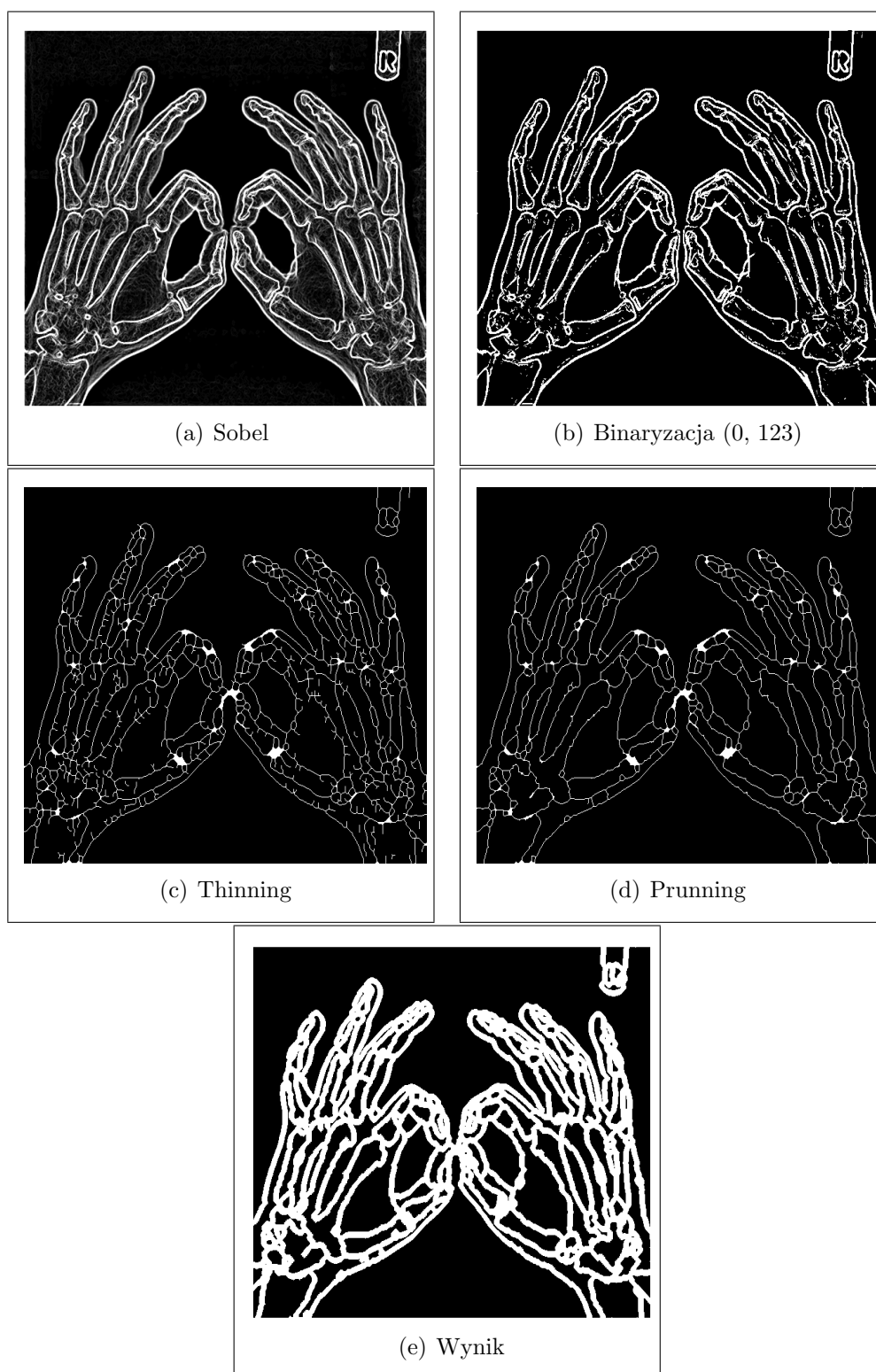
exec.add(new Thining(3));
exec.add(new Dilation2(new CrossSE(3)));
exec.add(new Dilation2(new CrossSE(3)));

exec.add(new Thining(3));
exec.add(new Dilation2(new CrossSE(3)));

exec.add(new Thining(1));
exec.add(new Dilation2(new CrossSE(3)));
exec.add(new Dilation2(new CrossSE(3)));
exec.add(new Erosion2(new CrossSE(3)));

exec.add(new Thining(7));
exec.add(new Prunning(10));
exec.add(new Dilation2(new CrossSE(5)));
exec.add(new Dilation2(new CrossSE(5)));
```

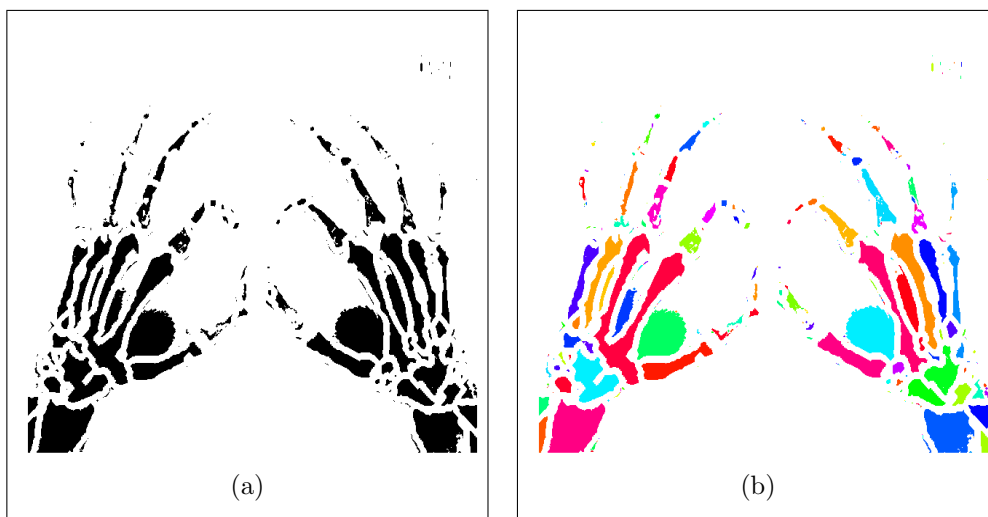
Rysunek 4: Fragment kodu. Uzyskiwanie zarysu krawędzi



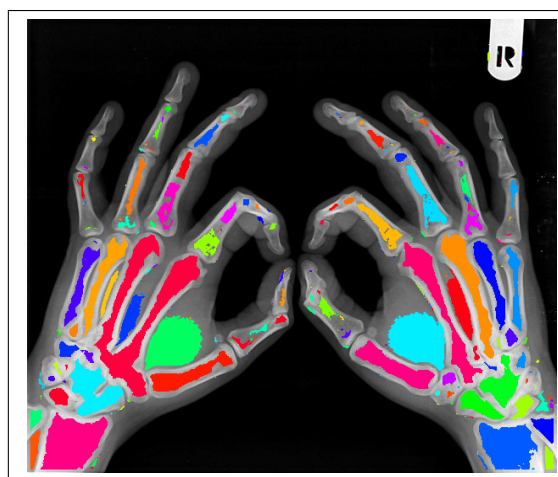
Rysunek 5: Uzyskiwanie zarysu kości.

c) Dodawanie obrazów

Mając wyniki z poprzednich sekcji nakładamy na rys. 2 wszystkie białe części z 5(e), gdyż nie chcemy zmieniać nic poza obszarem dłoni. Następnie używamy operacji ComponentLabeling 6(b).



Rysunek 6:



Rysunek 7: Wynik końcowy