

Wykrywanie minuncji w odciskach palców

Biometria

Natalia Choszczyk | Mateusz Deptuch

Politechnika Warszawska

Wydział Matematyki i Nauk Informacyjnych

29 maja 2025

Spis treści

1. Wprowadzenie	4
2. Metodologia i struktura algorytmu	5
2.1. Przegląd algorytmu	5
2.2. Wykorzystane biblioteki	5
3. Implementacja poszczególnych etapów	6
3.1. Normalizacja obrazu	6
3.2. Segmentacja i maska ROI	6
3.3. Estymacja orientacji	7
3.4. Filtrowanie Gabora	8
3.5. Szkieletyzacja	9
3.5.1. Szkieletyzacja morfologiczna	9
3.5.2. Algorytm K3M	10
3.6. Wykrywanie minuncji	11
4. Główne funkcje systemu	13
4.1. Pipeline przetwarzania	13
4.2. Wynik przetwarzania odcisku	14
5. Wyniki i analiza	15

6. Wnioski i podsumowanie	16
7. Bibliografia	17

1. Wprowadzenie

Głównym celem projektu było opracowanie kompletnego systemu do analizy i rozpoznawania minuncji w odciskach palców. Realizuje on pełną ścieżkę przetwarzania obrazu odcisku palca, począwszy od normalizacji obrazu, przez segmentację i estymację orientacji, aż po wykrywanie charakterystycznych punktów minuncji przy użyciu metod szkieletyzacji morfologicznej oraz algorytmu K3M.

Minuncje to charakterystyczne punkty w strukturze linii papillarnych odcisku palca, które stanowią podstawę identyfikacji biometrycznej. Wykrywanie tych punktów wymaga zastosowania technik przetwarzania obrazu takich jak filtrowanie Gabora, analiza orientacji lokalnej oraz algorytmy szkieletyzacji. W tym projekcie wykrywane są jedynie dwa rodzaje minuncji: zakończenia i bifurkacje.

2. Metodologia i struktura algorytmu

2.1. Przegląd algorytmu

Opracowany system składa się z następujących etapów przetwarzania:

1. Normalizacja obrazu w celu ujednolicenia wartości jasności i kontrastu
2. Maskowanie obrazu w celu wyodrębnienia obszaru zainteresowania
3. Estymacja mapy orientacji lokalnych linii papillarnych
4. Analiza częstotliwości grzbietów i filtrowanie Gabora
5. Szkieletyzacja obrazu metodą morfologiczną i algorytmem K3M
6. Wykrywanie i klasyfikacja minuncji

2.2. Wykorzystane biblioteki

- ▢ `numpy` - operacje macierzowe i numeryczne, podstawa wszystkich obliczeń
- ▢ `cv2` - biblioteka OpenCV do przetwarzania obrazów, operacje morfologiczne
- ▢ `matplotlib` - wizualizacja wyników przetwarzania
- ▢ `scipy` - zaawansowane operacje naukowe, w szczególności `scipy.ndimage` do rotacji i filtrowania
- ▢ `pandas` - zarządzanie danymi minuncji w formie tabelarycznej
- ▢ `math` - funkcje matematyczne do obliczeń kątów i orientacji

3. Implementacja poszczególnych etapów

3.1. Normalizacja obrazu

Funkcja `normalize_image` implementuje normalizację obrazu poprzez dopasowanie średniej i wariancji do zadanych wartości docelowych. Dla każdego piksela obliczana jest znormalizowana wartość według wzoru:

$$\text{pixel}_{norm} = \text{target_mean} \pm \sqrt{\frac{\text{target_var} \cdot (\text{pixel} - \text{img_mean})^2}{\text{img_var}}}$$

gdzie znak zależy od tego, czy piksel jest jaśniejszy czy ciemniejszy od średniej obrazu.



Rysunek 3.1: Obraz po normalizacji

3.2. Segmentacja i maska ROI

Funkcja `segment_and_normalize` realizuje segmentację obrazu poprzez analizę lokalnego odchylenia standardowego w blokach. Obszary o niskiej wariancji są maskowane jako tło. Dodatkowo nakładana

3.3. ESTYMACJA ORIENTACJI

jest maska eliptyczna koncentrująca się na centralnej części odcisku palca, co zapewnia, brak nałożenia maski w centralnej części odcisku.



Rysunek 3.2: Obraz z nałożoną maską

3.3. Estymacja orientacji

Funkcja `estimate_orientation_map` oblicza lokalną orientację linii papillarnych wykorzystując gradienty Sobela. Obraz zostaje podzielony na bloki i dla każdego z nich wyznaczana jest dominująca orientacja na podstawie wzoru:

$$\theta = \frac{1}{2} \arctan \left(\frac{\sum 2G_x G_y}{\sum (G_x^2 - G_y^2)} \right)$$

gdzie G_x i G_y to składowe gradientu w kierunkach poziomym i pionowym.



Rysunek 3.3: Mapa orientacji

3.4. Filtrowanie Gabora

Implementacja filtrów Gabora w funkcji `gabor_filter` wykorzystuje banki filtrów dostosowanych do lokalnej orientacji i częstotliwości grzbietów. Każdy filtr ma postać:

$$G(x, y) = \exp\left(-\frac{x'^2}{\sigma_x^2} - \frac{y'^2}{\sigma_y^2}\right) \cos(2\pi f x')$$

gdzie (x', y') to współrzędne obrócone zgodnie z lokalną orientacją.



Rysunek 3.4: Obraz po filtrze Gabora

3.5. Szkieletyzacja

System implementuje dwie metody szkieletyzacji:

3.5.1. Szkieletyzacja morfologiczna

Metoda `morphological_skeleton` wykorzystuje iteracyjne operacje morfologiczne erozji i otwarcia do uzyskania szkieletu odcisku palca.

Na początku, stosowane jest morfologiczne otwarcie z małym jądrem eliptycznym, w celu usunięcia drobnych szumów w obrazie.

Szkielet obrazu jest początkowo pusty. W każdej iteracji wykonywane są następujące kroki:

1. Obraz jest poddawany operacji otwarcia , co usuwa cienkie struktury.
2. Od oryginału odejmowany jest wynik otwarcia. Różnica ta dodawana jest do szkieletu.
3. Obraz poddawany jest erozji.
4. Wynik odejmowania jest dodawany do bieżącego szkieletu.
5. Proces jest powtarzany dopóki obraz po erozji nie zawiera już żadnych białych pikseli.

Po zakończeniu iteracji, otrzymany szkielet jest odwracany kolorystycznie.

Morphological Skeleton



Rysunek 3.5: Obraz po szkieletyzacji morfologicznej

3.5.2. Algorytm K3M

Funkcja `k3m_skeleton` implementuje iteracyjny algorytm szkieletowania binarnego obrazu. Działanie algorytmu polega na iteracyjnym usuwaniu pikseli, spełniających konkretne warunki topologiczne, z krawędzi obiektów.

W każdej iteracji analizowane są piksele spełniające poniższe warunki:

- Piksel jest aktywny (równy 1).
- Liczba sąsiadów (spośród 8 otaczających pikseli) zawiera się w przedziale 2 do 6.
- Liczba przejść między 0 i 1 w cyklicznym ciągu sąsiadów jest równa 1 (warunek zapewniający zachowanie spójności).
- Co najmniej jeden z sąsiadów należących do aktualnie analizowanego wzorca (**pattern**) ma wartość 0.

Lista wzorców analizowana jest w kolejnych etapach i odnosi się do pozycji sąsiadów wokół badanego piksela, zgodnie z numeracją:

[0]	[1]	[2]
[7]	X	[3]
[6]	[5]	[4]

- [0, 2, 4, 6] – sprawdza cztery główne kierunki (góra, prawo, dół, lewo); piksel zostaje oznaczony do usunięcia, jeśli przynajmniej jeden z tych sąsiadów jest pusty (0).
- [0], [2], [4], [6] – kolejne iteracje analizujące te kierunki pojedynczo, co pozwala na dokładniejsze usuwanie zbędnych pikseli i lepsze zachowanie topologii.

Proces powtarzany do momentu, gdy w żadnym etapie nie znajdą już zmiany.

Algorytm zapewnia szkielet jedno-pikselowej szerokości, co czyni go odpowiednim do dalszej analizy strukturalnej, w tym wykrywania minucji w obrazach linii papilarnych.

K3M Skeleton



Rysunek 3.6: Obraz po szkieletyzacji K3M

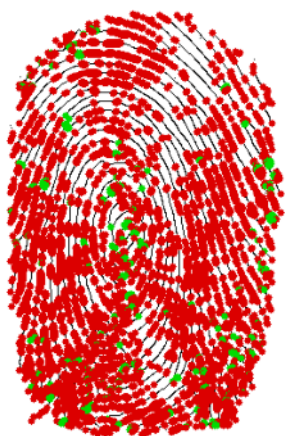
3.6. Wykrywanie minuncji

Funkcja `extract_minutiae` identyfikuje dwa typy minuncji:

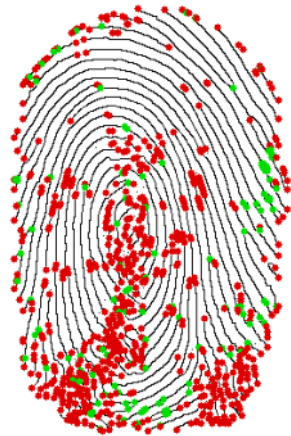
- **Zakończenia** (endings) - punkty, w których liczba przejść między pikselami białymi i czarnymi w 8-sąsiedztwie wynosi 1
- **Rozgałęzienia** (bifurcations) - punkty z 3 przejściami w sąsiedztwie

Dodatkowo system może obliczać orientację minuncji i wizualizować je za pomocą strzałek wskazujących kierunek lokalnej orientacji.

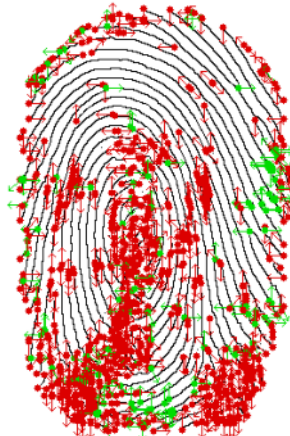
Morphological Minutiae



K3M Minutiae



K3M Minutiae with arrows



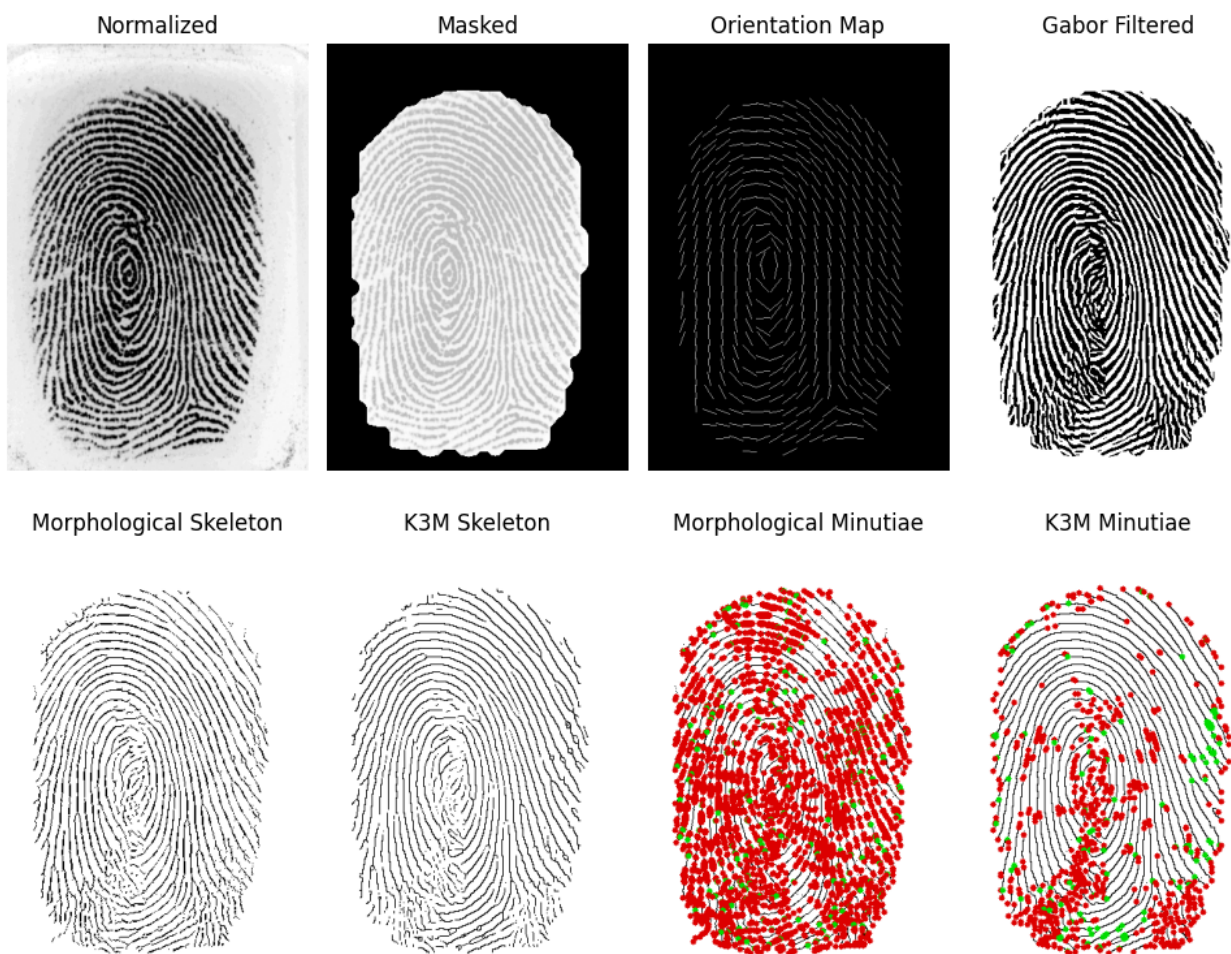
Rysunek 3.7: Obraz po wykryciu minuncji na pomocą dwóch metod

Jak widać, dużo lepiej minuncje wykrywane są w szkielecie otrzymanym przez algorytm K3M, co wynika z większej ciągłości oraz zapewnionej grubości jednego piksela szkieletu.

4. Główne funkcje systemu

4.1. Pipeline przetwarzania

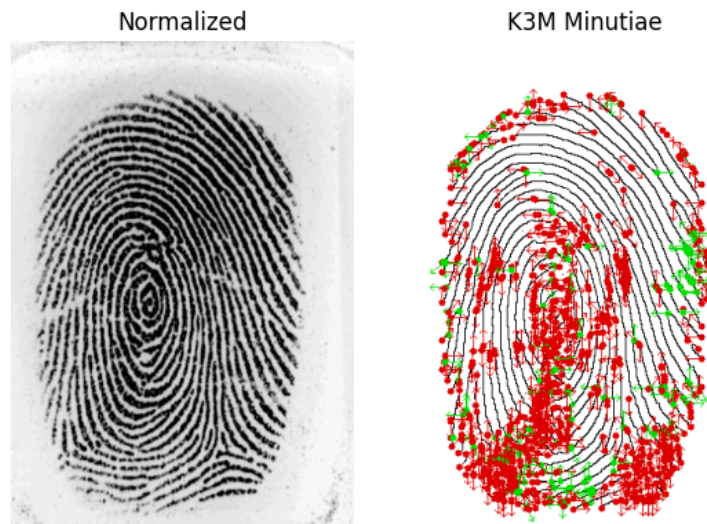
Funkcja `fingerprint_pipeline` realizuje kompletny proces analizy odcisku palca. Wyświetla wszystkie etapy przetwarzania w formie macierzy 2×4 obrazów, umożliwiając wizualną kontrolę każdego kroku algorytmu.



Rysunek 4.1: Wynik funkcji `fingerprint_pipeline`

4.2. Wynik przetwarzania odcisku

Funkcja `fingerprint_output` koncentruje się na końcowym wyniku - porównuje obraz znormalizowany z wykrytymi minuncjami K3M, dodając wizualizację orientacji minuncji w postaci strzałek.



Rysunek 4.2: Wynik funkcji `fingerprint_output`

5. Wyniki i analiza

System skutecznie przetwarza obrazy odcisków palców, wyodrębniając charakterystyczne minuncje niezbędne do identyfikacji biometrycznej. Porównanie metod szkieletyzacji pokazuje, że algorytm K3M generuje czystsze szkielety z mniejszą liczbą artefaktów w porównaniu do metody morfologicznej.

Wykryte minuncje są klasyfikowane jako zakończenia lub rozgałęzienia z zachowaniem informacji o ich położeniu i orientacji. Dane te mogą być następnie wykorzystane do budowy szablonów biometrycznych i porównywania odcisków palców.

6. Wnioski i podsumowanie

Kluczowe osiągnięcia projektu to:

- Implementacja pełnego pipeline'u przetwarzania odcisków palców
- Porównanie dwóch metod szkieletyzacji – morfologicznej i K3M
- Wykrywanie i klasyfikacja minuncji z informacją o orientacji

Na podstawie eksperymentów stwierdzono, że algorytm K3M lepiej nadaje się do analizy struktur linii papilarnych. Zapewnia on większą ciągłość oraz gwarantuje uzyskanie jedno-pikselowej szerokości linii, co jest kluczowe dla poprawnej detekcji i klasyfikacji minuncji. W porównaniu do podejścia morfologicznego, które może pozostawiać fragmenty o większej grubości lub niedoskonałości w strukturze, K3M dostarcza bardziej regularny i spójny szkielet.

System może być rozwijany o dodatkowe funkcjonalności, takie jak dopasowywanie szablonów minuncji, ocena jakości odcisków palców czy integracja z bazami danych biometrycznych. Jakość wykrywania minuncji zależy od parametrów segmentacji i filtrowania, które mogą być dostrajane dla różnych typów obrazów.

7. Bibliografia

- 📖 *K3M: A Universal Algorithm for Image Skeletonization and A Review of Thinning Techniques* - Khalid Saeed, Marek Tabędzki, Mariusz Rybniak, Marcin Adamski
- 📖 <https://pages.mini.pw.edu.pl/~rafalkoj/www/?Dydaktyka:2024>
- 📖 Dokumentacja OpenCV - <https://docs.opencv.org/>
- 📖 https://github.com/cuevas1208/fingerprint_recognition