

Politechnika Łódzka
Wydział Fizyki Technicznej, Informatyki i Matematyki
Stosowanej
Kierunek Informatyka

Jan Stolarek

Identyfikacja użytkownika na podstawie analizy linii papilarnych

Praca magisterska
napisana pod kierunkiem
prof. dr hab. Michała Jacymirskiego
oraz
mgr inż. Bartłomieja Stasiaka

Łódź 2008

Spis treści

Wstęp	1
1 Systemy biometryczne	3
1.1 Podział systemów biometrycznych	3
1.2 Cechy biometryczne	6
1.3 Błędy systemów biometrycznych	11
1.3.1 Błędy systemów weryfikacji	11
1.3.2 Błędy systemów identyfikacji	14
1.4 Podsumowanie	15
2 Rozpoznawanie odcisków palców	16
2.1 Pozyskiwanie odcisków palców	16
2.2 Cechy odcisków palców	24
2.3 Ekstrakcja cech	26
2.3.1 Metody oparte o operacje morfologiczne	26
2.3.2 Metody oparte o analizę linii	27
2.3.3 Zastosowanie sieci neuronowych	37
2.4 Dopasowywanie odcisków palców	38
2.4.1 Metody korelacyjne	39
2.4.2 Metody oparte o minucje	40
2.4.3 Metody oparte o krawędzie	44
2.5 Podsumowanie	45
3 Ulepszone algorytmy rozpoznawania odcisków palców	46
3.1 Przetwarzanie wstępne	46
3.2 Ekstrakcja cech	48

3.2.1	Segmentacja odcisku z tła	48
3.2.2	Wyznaczanie kierunku linii	50
3.2.3	Wyznaczanie nowych punktów początkowych	50
3.2.4	Warunki stopu	51
3.2.5	Wykrywanie fałszywych minucji	54
3.3	Dopasowywanie wzorców	56
3.3.1	Wyznaczanie przekształcenia geometrycznego	56
3.3.2	Wyznaczanie stopnia podobieństwa odcisków	58
4	Technologie i narzędzia	59
4.1	Język programowania	59
4.2	Środowisko programistyczne	60
4.3	Techniki programistyczne	60
4.4	Użyte biblioteki	62
4.5	Baza odcisków palców	64
4.6	Aplikacja	66
4.6.1	Struktura katalogów projektu	66
4.6.2	Struktura pakietów	67
5	Badania eksperymentalne	69
5.1	Ogólne wyniki	70
5.1.1	Skuteczność rozpoznawania	70
5.1.2	Szybkość działania	70
5.2	Ekstrakcja cech	71
5.2.1	Przetwarzanie wstępne	71
5.2.2	Mapa kierunków	72
5.2.3	Wyznaczanie nowych punktów początkowych	73
5.2.4	Kryteria stopu	74
5.2.5	Wykrywanie fałszywych minucji	75
5.2.6	Polepszanie jakości sekcji	77
5.3	Dopasowywanie wzorców	78
5.3.1	Wyznaczanie przekształcenia geometrycznego	78
5.3.2	Wyznaczanie stopnia podobieństwa odcisków	80
5.4	Optymalne ustawienia	81

5.5 Porównanie z istniejącymi algorytmami	83
Podsumowanie	84
Bibliografia	86
A Obsługa aplikacji	88
A.1 Uruchamianie	88
A.2 Konfiguracja parametrów	89
B Płyta CD	94

Wstęp

Każdy z nas posiada na czubkach swoich palców drobne bruzdy skórne, tzw. *linie papilarne*. Francis Galton, uważany za twórcę współczesnej daktyloskopii – nauki zajmującej się kryminalnym badaniem odcisków palców – w roku 1892 opublikował wyniki swoich badań w pracy zatytułowanej *Finger prints*. Uzasadnił w niej naukowo, że odciski palców dwojga ludzi nie mogą być identyczne, jak również nie mogą być identyczne linie papilarne tego samego człowieka znajdujące się na różnych palcach. Dowiódł także, że na przestrzeni całego ludzkiego życia linie papilarne pozostają niezmienione i nie da się ich usunąć. Galton nie był pierwszym, który badał ten element ludzkiego ciała. Przed Galtonem badaniami linii papilarnych zajmowali się H. Faulds (jako pierwszy na świecie zidentyfikował za pomocą odcisków palców sprawcę przestępstwa w roku 1879), W. Hershell (wyniki swoich badań nad liniami papilarnymi ogłosił w roku 1877 w czasopiśmie *The Nature*) oraz E. Purkynie (jego pionierska praca klasyfikująca typy odcisków palców datuje się na rok 1823) [9]. Jednak to Galton nadał tej dziedzinie wymiar naukowy i to dzięki jego pracom stała się ona nieodłącznym elementem kryminalistyki aż po dziś dzień.

Biometria to nauka zajmująca się badaniem fizycznych i behawioralnych cech istot żywych w celu odróżniania ich od siebie, a analiza odcisków palców jest dziś jedną z wielu metod biometrycznych. Znajduje ona coraz szersze zastosowanie nie tylko w kryminalistyce, ale w zagadnieniach związanych z szeroko pojętym bezpieczeństwem. Od czasu kiedy komputer stał się nieodłącznym elementem w każdej firmie, instytucji państwowej i niemal w każdym domu minęło zaledwie kilkanaście lat. Na twardej dyskach przechowywane są coraz większe ilości poufnych danych, a uniemożliwienie dostępu do nich osobom niepowołanym ma kluczowe znaczenie dla bezpieczeństwa firm i państwa. Dlatego też zagadnienia rozpoznawania użytkownika stają się w dzisiejszym świecie coraz ważniejsze. Przy rosnącej mocy obliczeniowej komputerów dotychczas czę-

sto stosowane zabezpieczenia za pomocą haseł – Coś Co Wiesz (ang. *Something You Know*) – przestają zapewniać wymagany poziom bezpieczeństwa. Hasło można złamać, można je zapomnieć lub – co gorsza – nierozważny użytkownik może zapisać je w miejscu dostępnym dla osób niepowołanych. Również klucz sprzętowy – Coś Co Masz (ang. *Something You Have*) – nie zapewnia najwyższego poziomu bezpieczeństwa, gdyż może on zostać z łatwością zgubiony lub skradziony. Rozwiązaniem niedogodności haseł i kluczy sprzętowych mogą być właśnie metody biometryczne dokonujące weryfikacji użytkownika na podstawie Czegoś Czym Jest (ang. *Something You Are*). Dlatego też w ostatnich latach coraz bardziej rośnie zainteresowanie systemami biometrycznymi, a wpływy branży mają podwoić się w ciągu najbliższych czterech lat [10].

Ilości danych przechowywanych obecnie w bazach danych biometrycznych są tak wielkie, że ludzie nie są już w stanie analizować i przeszukiwać ich samodzielnie¹. Rosnące zapotrzebowanie doprowadziło do rozwoju licznych metod automatycznego rozpoznawania odcisków palców. Może się wydawać, że jest to zagadnienie w pełni zbadane, ponieważ było to jedno z pierwszych zagadnień automatycznego rozpoznawania wzorców już kilkadziesiąt lat temu. Pogląd ten jest jednak błędny, gdyż automatyczne rozpoznawanie odcisków palców wciąż jest dużym wyzwaniem.

Z tego powodu w poniższej pracy postawiono sobie za cel ulepszenie istniejących algorytmów rozpoznawania odcisków palców, służących do weryfikacji użytkownika. W pierwszym rozdziale zaprezentowano przegląd ogólnych zagadnień związanych z systemami biometrycznymi. Rozdział 2 poświęcony został metodom i algorytmom stosowanym w analizie linii papilarnych, jako że jest to metoda wiodąca. Zaprezentowano również ulepszone wersje algorytmów rozpoznawania odcisków palców (Rozdział 3) oraz opisano stworzoną na potrzeby tej pracy implementację i związane z nią technologie (Rozdział 4). Nowe rozwiązania poddano badaniom eksperymentalnym i porównaniu z istniejącymi już algorytmami w Rozdziale 5. W Dodatku A zamieszczono informacje dotyczące obsługi napisanej aplikacji. W Dodatku B dołączona została płyta z aplikacją oraz elektroniczną wersją tej pracy.

¹Jako przykład może posłużyć baza odcisków palców należąca do FBI. Zawiera ona dane ponad 47 milionów osób [28].

Rozdział 1

Systemy biometryczne

Systemy biometryczne to systemy, które rozpoznają osobę na podstawie jej cech fizycznych (np. linie papilarne, siatkówka oka) bądź behawioralnych (np. brzmienie głosu). Proces powstawania takiego systemu jest złożony. Nim więc rozpocznie się tworzenie nowego systemu konieczne jest zrozumienie wielu kluczowych zagadnień, które zostaną przybliżone w tym rozdziale. Przede wszystkim należy określić wymagania stawiane budowanemu systemowi. Czy wystarczy sprawdzić, że użytkownik jest tym za kogo się podaje? A może będzie trzeba ustalać tożsamość osoby na podstawie jej cech biometrycznych? Rozdział 1.1 zawiera szczegółowe omówienie tych aspektów. Istotny jest także dobór cech biometrycznych rozpoznawanych przez tworzony system. Pewne cechy mogą bowiem zapewniać wyższą skuteczność, ale być trudne do przekształcenia na postać cyfrową za pomocą czytnika. Rozdział 1.2 prezentuje stosowane obecnie metody biometryczne.

1.1 Podział systemów biometrycznych

Podziału tych systemów na różne rodzaje można dokonać ze względu na wiele kryteriów. W zależności od zadania, jakie wykonują, dzieli się je na dwa rodzaje: systemy *weryfikacji* i systemy *identyfikacji*.

System weryfikacji ma za zadanie sprawdzić czy użytkownik rzeczywiście jest tym, za kogo się podaje. W tym celu użytkownik poddawany weryfikacji podaje swój

numer PIN¹ i pozwala na pobranie, za pomocą czytnika, wybranych cech biometrycznych. Następnie dokonywane jest porównanie cech danej osoby, odczytanych w trakcie weryfikacji, z wzorcem danej osoby zapisanym w bazie danych. Przeprowadzane jest porównanie jeden-do-jednego w celu sprawdzenia czy osoba jest tą, za którą się podaje. System potwierdza podawaną tożsamość bądź ją odrzuca.

System identyfikacji rozpoznaje kim jest dany użytkownik. Aby tego dokonać odczytuje, za pomocą czytnika, cechy biometryczne danej osoby (numer PIN nie jest w tym trybie wykorzystywany), a następnie porównuje je z wzorcami zapisanymi w bazie danych. System przeprowadza porównanie jeden-do-wielu w celu określenia tożsamości osoby. Odpowiedzią systemu jest zidentyfikowana tożsamość bądź odrzucenie danej osoby jako niezarejestrowanej w bazie.

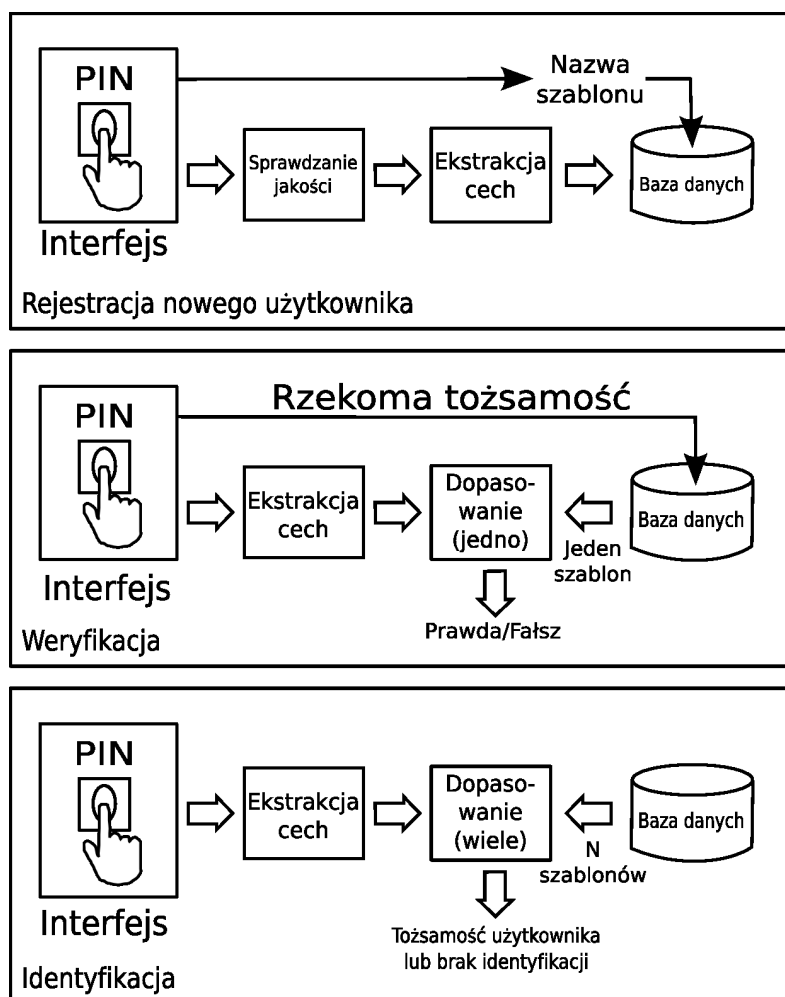
W niniejszej pracy terminy *rozpoznawanie*, *identyfikacja* i *autentykacja* będą często używane zamiennie do określenia zdolności systemu do rozpoznawania użytkowników, bez względu na to czy chodzi o weryfikację czy identyfikację w sensie trybu pracy systemu. Przebieg weryfikacji i identyfikacji został zilustrowany na Rysunku 1.1.

Istotnym elementem systemu biometrycznego jest faza rejestracji użytkownika (ang. *user enrollment*) w systemie (Rysunek 1.1). Polega ona na odczytaniu, za pomocą czytnika, danych biometrycznych i przekształceniu ich do postaci cyfrowej. Przeprowadzana jest kontrola jakości uzyskanych danych w celu zapewnienia, że jakość ta jest wystarczająca do skutecznego rozpoznawania osoby. Następnie uzyskany zapis poddawany jest procesowi ekstrakcji cech w celu stworzenia tzw. szablonu (ang. *template*). Szablon jest zapisywany w bazie danych lub na karcie magnetycznej wydawanej użytkownikowi. W tym drugim przypadku otrzymujemy połączenie metody Czegoś Co Masz z metodą Czegoś Czym Jesteś, a w przypadku przypisania użytkownikowi numeru PIN (bądź innego identyfikatora, który musi wprowadzić przy próbie autentykacji) także z metodą Czegoś Co Wiesz.

Systemy biometryczne można podzielić, ze względu na wymaganą od nich szybkość odpowiedzi, na dwa rodzaje: pracujące w trybie *on-line* i w trybie *off-line*.

System pracujący w trybie on-line to system, od którego wymaga się odpowiedzi na zachodzące zdarzenia w czasie rzeczywistym (przykładowo użytkownik

¹ang. *Personal Identification Number*, czyli numer identyfikacyjny użytkownika. Zamiast niego można wymagać od użytkownika podania imienia i nazwiska lub przypisanego mu loginu.



Rysunek 1.1: Schemat rejestracji, weryfikacji i identyfikacji użytkownika

chce zalogować się do systemu). Takie systemy są z reguły w pełni zautomatyzowane. Rejestracja w systemie jest w pełni automatyczna i w jej trakcie nie jest przeprowadzana ręczna kontrola jakości pobranego wzorca. Również proces podejmowania decyzji rozpoznawania odbywa się bez udziału człowieka, a skanowanie cech przeprowadza się za pomocą automatycznego skanera, który dokonuje natychmiastowej konwersji do postaci cyfrowej.

System pracujący w trybie off-line to system, któremu pozwala się na odpowiadanie na kierowane do niego zapytania z pewnym opóźnieniem. Przykładem może być tu policyjna baza odcisków palców, gdzie zgłasza się odciski znalezione na

miejscu popełnienia przestępstwa i oczekuje na ich identyfikację. Takie systemy są półautomatyczne, a człowiek bierze udział zarówno w procesie rejestrowania nowych użytkowników (ekspert może czuwać, aby odciski pobierane od przestępców miały zadowalającą jakość) oraz identyfikacji (system może wytypować kilka najbardziej prawdopodobnych trafień zostawiając ostateczną decyzję ekspertowi).

Z powyższego opisu widać, że najczęściej systemy dokonujące identyfikacji pracują w trybie off-line, a systemy weryfikacyjne w trybie on-line (choć oczywiście nie jest to regułą).

Można także mówić o pracy systemu biometrycznego w trybie *pozytywnym* oraz *negatywnym*. Praca w trybie pozytywnym polega na stwierdzaniu czy osoba jest tą osobą, za którą się podaje. Taka sytuacja ma przykładowo miejsce przy próbie dostępu użytkownika do kontrolowanych przez system zasobów. System musi wtedy zweryfikować czy użytkownik naprawdę jest tą osobą, którą twierdzi że jest. W przypadku potwierdzenia tożsamości dokonywane jest sprawdzenie czy użytkownik ma prawo korzystać z zasobów, których żąda. Jeśli tak, to system zezwala na dostęp. Praca w trybie negatywnym polega na stwierdzaniu czy osoba jest tym, kim twierdzi, że nie jest. Przykładem może być sytuacja, w której użytkownik próbuje się po raz kolejny zarejestrować w systemie pod inną tożsamością. System pracujący w trybie negatywnym dokona porównania takiej osoby z danymi zapisanymi w bazie, po czym stwierdzi, że jest ona już w niej zarejestrowana i odrzuci żądanie tej osoby.

1.2 Cechy biometryczne

Tworząc system biometryczny należy zdecydować jaka cecha będzie używana jako identyfikator biometryczny². Aby danej cechy można było użyć w tej roli musi ona spełniać kilka podstawowych warunków:

Uniwersalność : posiada ją każdy człowiek

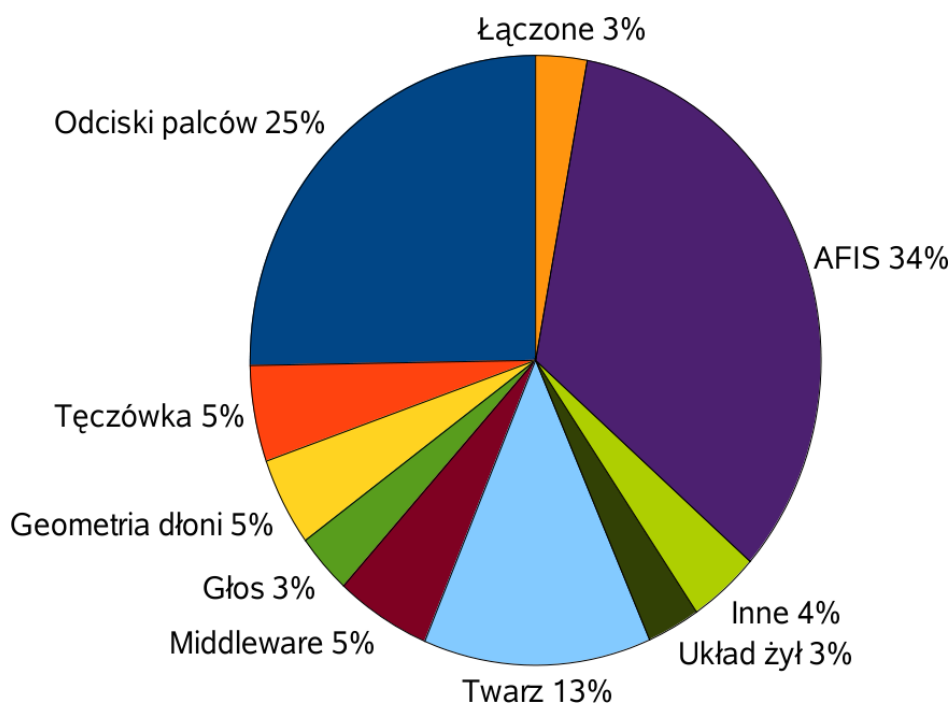
Niepowtarzalność : dowolnych dwoje ludzi daje się rozróżnić za pomocą tej cechy

Niezmiennność : nie zmienia się wraz z upływem czasu

²Można zbudować system łączony wykorzystujący kilka cech biometrycznych.

Łatwość pozyskiwania : daje się łatwo zmierzyć i przekształcić na postać cyfrową

W praktyce należy uwzględnić dodatkowe czynniki. W działającym systemie duże znaczenie ma bez wątpienia jego wydajność. Trzeba więc rozważyć czy użycie danej cechy umożliwia uzyskanie wymaganej skuteczności rozpoznawania przy określonych ograniczeniach czasowych. Uwzględnić należy także skłonność ludzi do zaakceptowania danej cechy biometrycznej w ich życiu codziennym. Może się zdarzyć, że wymagane będzie działanie systemu bez współpracy ze strony osoby rozpoznawanej (np. w przypadku identyfikacji osób poszukiwanych na obrazach z kamer). Istotne znaczenie ma także prawdopodobieństwo oszukania systemu poprzez podrobienie danej cechy lub oszukanie czytnika.



Rysunek 1.2: Udział poszczególnych metod w rynku systemów biometrycznych na podstawie [10]

Obecnie istnieje wiele aplikacji wykorzystujących różne cechy biometryczne człowieka. Jak widać na Rysunku 1.2, dominują systemy AFIS³ oraz systemy weryfikacji oparte o odciski palców. Oczywiście nie istnieje jedna cecha, która nadawałaby się

³ang. *Automated Fingerprint Identification System*, czyli w pełni automatyczne systemy służące do identyfikacji na podstawie odcisków palców.

do zastosowania w każdej sytuacji i w każdym systemie. Zanim podejmie się decyzję o wykorzystaniu konkretnego identyfikatora biometrycznego należy rozważyć wszystkie jego za i przeciw. Poniżej zaprezentowano omówienie najczęściej stosowanych cech biometrycznych.

DNA : Kod DNA jest unikalny (z wyjątkiem bliźniąt jednojajowych, które mają identyczny wzorec). Metoda wykorzystywana głównie w kryminalistyce. W przypadku innych zastosowań jej użycie jest ograniczone z kilku powodów. Po pierwsze łatwo jest ukraść próbkę DNA (np. w postaci włosa) w celu wykorzystania jej do uzyskania dostępu. Po drugie obecne techniki dopasowywania DNA wymagają skomplikowanych operacji chemicznych i udziału eksperta, tak więc niemożliwe staje się wykorzystanie tej metody w systemach on-line. Kolejną z przeszkód są kwestie związane z prywatnością. Na podstawie DNA można uzyskać informacje dotyczące zdrowia danej osoby (np. podatności na pewne choroby) i istnieją poważne obawy, że takie informacje mogłyby zostać wykradzione i wykorzystane w niepożądany sposób.

Ucho : Metoda opiera się na mierzeniu odległości między określonymi punktami ucha. Kształt ucha pozwala na rozróżnianie ludzi, jednak nie sądzi się, aby ta cecha była unikalna. Można więc zastosować ją do systemów, od których nie jest wymagany najwyższy stopień bezpieczeństwa, przeznaczonych dla niewielkiej ilości użytkowników.

Twarz : Rozpoznawanie w oparciu o kształt twarzy jest najbardziej akceptowalną cechą biometryczną, gdyż ten sposób używany jest przez ludzi do wzajemnego rozpoznawania się. Dużym wyzwaniem jest jednak zaprojektowanie systemu, który uwzględniałby proces starzenia, zmiany wyrazu twarzy oraz był niezależny od kąta obserwacji i oświetlenia. W przypadku systemów w pełni automatycznych problemem byłaby także możliwość charakteryzacji.

Obrazy termiczne : Wzorec ciepła emitowanego przez każdego z nas jest unikalny (także dla bliźniąt). Można zarejestrować go za pomocą kamery pracującej w podczerwieni. Metoda rejestrowania jest bezkontaktowa, jednak problemem staje się rejestracja cechy w obecności emiterów ciepła (np. klimatyzacja w samochodzie).

Czynnikiem, który istotnie hamuje wzrost popularności tej technologii, jest wysoka cena czytników.

Układ żył w dłoni : Metoda wykorzystuje czytnik, pracujący w paśmie zbliżonym do podczerwieni, do badania układu żył w dłoni. Wiarygodność tej cechy jest bardzo wysoka ze względu na jej unikalność oraz nikłe prawdopodobieństwo na wykradzenie czyjegoś wzorca i oszukanie czytnika. Jednak podobnie jak w poprzedniej metodzie wysoka cena czytników powoduje ograniczenie popularności tej techniki.

Chód : Chód jest cechą behawioralną. Nie jest unikalny, ale dla aplikacji o niskim poziomie zabezpieczeń, jest wystarczający. Technika polega na rejestracji, za pomocą kamery, chodu danej osoby i mierzeniu określonych ruchów stawów. Ta cecha jest również powszechnie akceptowana. Problemem jest zmienność tej cechy w czasie (dużym problemem mogą być urazy) oraz wysoka złożoność obliczeniowa związana z koniecznością przetwarzania sekwencji klatek wideo.

Geometria dłoni : Technika polega na mierzeniu długości palców oraz innych elementów dłoni, np. jej szerokości bądź odległości między palcami. Cecha nadaje się do wykorzystania w systemach weryfikacji o małej liczbie użytkowników, jednak do celów identyfikacji nie jest wystarczająco unikalna. Zaletą jest bardzo mały rozmiar szablonu.

Tęczówka : Polega na rejestracji obrazu tęczówki oka. Dokonuje się tego w sposób bezkontaktowy. Uważa się, że tęczówka jest w pełni unikalna. Technologie rozpoznawania tęczówek uznawane są za bardzo dokładne i szybkie. Wadą jest wymagana współpraca osoby rozpoznawanej, zarówno w procesie rejestracji jak i późniejszego rozpoznawania.

Zapach : Wiadomo, że każda rzecz wydziela swój zapach zależny od jej składu chemicznego. Również ludzie mają swój charakterystyczny zapach, który zależy od białek wydzielanych z potem. Identyfikacji można dokonywać za pomocą zespołu czujników, z których każdy jest wyczulony na inne składniki chemiczne. Nie jest jeszcze do końca jasne czy da się rozpoznać zapach ciała pomimo używania dezodorantów i innych środków kosmetycznych. Tą metodą nie da się również rozróżnić bliźniąt jednojajowych.

Siatkówka : Istotą tej cechy jest unikalny układ naczyń krwionośnych w siatkówce oka. W trakcie rejestracji wymaga się od rozpoznawanej osoby skupienia wzroku na określonym punkcie w celu pobrania obrazu ustalonego fragmentu siatkówki. Uważa się, że ta cecha biometryczna jest najbezpieczniejsza, ponieważ nie są znane sposoby na zmianę lub podrobienie układu naczyń w oku. Metoda wymaga świadomego udziału ze strony użytkownika i kontaktu oka z czujnikiem, co powoduje zmniejszenie jej akceptowalności wśród ludzi.

Podpis : Jest to kolejna cecha behawioralna. Pomimo, że wymaga świadomego udziału ze strony użytkownika, jest w pełni akceptowana, ponieważ własnoręczne podpisywanie dokumentów, celem poświadczenia ich autentyczności jest powszechne w wielu kulturach. Problemem jest duża zmienność tej cechy, zarówno w czasie jak i pod wpływem stanów emocjonalnych. Istnieje również możliwość sfałszowania podpisu.

Głos : Kolejna z cech behawioralnych, która jest również powszechnie akceptowalna. Głos jest jedyną cechą biometryczną, która może zostać wykorzystana do identyfikacji przez telefon. Unikalność głosu jest niewystarczająca do identyfikacji, tak więc nadaje się on tylko do weryfikacji użytkownika. Problemem są zniekształcenia głosu wywołane zarówno przez jego konwersję do postaci cyfrowej jak i stany chorobowe.

Linie papilarne : Szczegółowo analizowana w tej pracy metoda rozpoznawania użytkowników za pomocą odcisków palców oferuje doskonale zbalansowane właściwości. Wiemy o całkowitej unikalności odcisków palców oraz ich niezmienności w trakcie ludzkiego życia. Posiada je praktycznie każdy człowiek⁴. Dostępne są czytniki, które w trakcie skanowania na żywo potrafią dostarczyć wysokiej jakości obraz. Coraz większy wzrost popularności tej technologii spowodował znaczny spadek cen czytników linii papilarnych. Również wydajność systemów opartych o linie papilarne jest wysoka. Dzięki wysokiej popularności technologia jest co-

⁴Obecnie odciski palców można usunąć za pomocą głębokiej dermabrazji. Jest to zabieg chirurgii plastycznej polegający na ścieraniu wierzchnich warstw naskórka. Istnieją także rzadkie schorzenia genetyczne prowadzące w swych skutkach do braku linii papilarnych. Również w przypadku ran skóry na głębokość powyżej 1mm oraz odmrożeń i oparzeń III i IV stopnia linie papilarne ulegają zniszczeniu, jednak w przypadku delikatniejszych zranień zawsze odrastają w pierwotnej formie.

raz bardziej dopracowywana co sprawia, że oszukiwanie czytników staje się coraz trudniejsze. Najistotniejszą wadą związaną z tą cechą biometryczną jest brak możliwości wykorzystania jej do identyfikacji osób w czasie rzeczywistym bez ich wiedzy.

1.3 Błędy systemów biometrycznych

System biometryczny rozpoznający odciski palców⁵, jako wynik porównania dwóch odcisków, najczęściej zwraca stopień ich wzajemnego podobieństwa ss (z reguły liczba z przedziału od 0 do 1). Liczba ta określa podobieństwo między szablonem zapisanym w bazie, a weryfikowanym odciskiem. Im stopień podobieństwa bliższy jedności, tym większe prawdopodobieństwo, że oba odciski zostały pobrane z tego samego palca. Wynik bliżej zera oznacza duże prawdopodobieństwo tego, że są to dwa różne odciski. Decyzja o zaakceptowaniu bądź odrzuceniu odcisku podejmowana jest przez system na podstawie ustalonego progu granicznego t .

System biometryczny popełnia dwa rodzaje błędów. Pierwszy polega na uznaniu za identyczne dwóch odcisków, które w rzeczywistości pochodzą z różnych palców. Jest to tzw. fałszywe dopasowanie⁶. Drugi rodzaj błędu to uznanie dwóch identycznych odcisków palców za różne – tzw. fałszywe niedopasowanie⁷.

1.3.1 Błędy systemów weryfikacji

Formalnie błędy systemów weryfikacji można opisać w następujący sposób. Przez T oznaczmy szablon zapisany w bazie danych, natomiast przez I oznaczmy odcisk porównywany w procesie weryfikacji. Istnieją dwie możliwe hipotezy:

$H_0 : I \neq T$, odcisk wejściowy nie pochodzi od tej samej osoby co szablon.

$H_1 : I = T$, odcisk wejściowy pochodzi od tej samej osoby co szablon.

Związane z nimi są dwie decyzje:

⁵Informacje w tym rozdziale mogą dotyczyć każdego systemu opartego o inne cechy biometryczne. Jednak na potrzeby tej pracy ograniczono rozważania do linii papilarnych.

⁶ang. *false match*

⁷ang. *false non-match*

D_0 : Osoba nie jest tym, za kogo się podaje.

D_1 : Osoba jest tym, za kogo się podaje.

Weryfikacja oparta jest o stopień podobieństwa $s(T,I)$. Jeśli jest on większy od ustalonego progu t , to należy podjąć decyzję D_1 . W przeciwnym wypadku należy podjąć decyzję D_0 . Możemy więc określić następujące dwa rodzaje błędów:

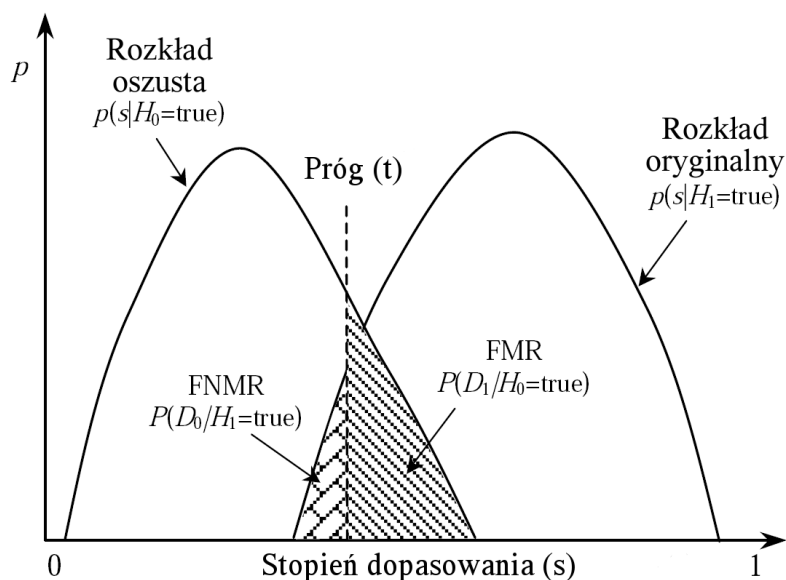
Typ I : Błędne dopasowanie (zadecydowano D_1 , gdy prawdą jest H_0).

Typ II : Błędne niedopasowanie (zadecydowano D_0 , gdy prawdą jest H_1).

Prawdopodobieństwo wystąpienia błędu typu pierwszego oznacza się jako FMR⁸. Prawdopodobieństwo wystąpienia błędu drugiego rodzaju oznaczane jest jako FNMR⁹:

$$FMR = P(D_1 | H_0 = \text{prawda})$$

$$FNMR = P(D_0 | H_1 = \text{prawda})$$



Rysunek 1.3: FMR i FNMR w zależności od wartości progu t [1]

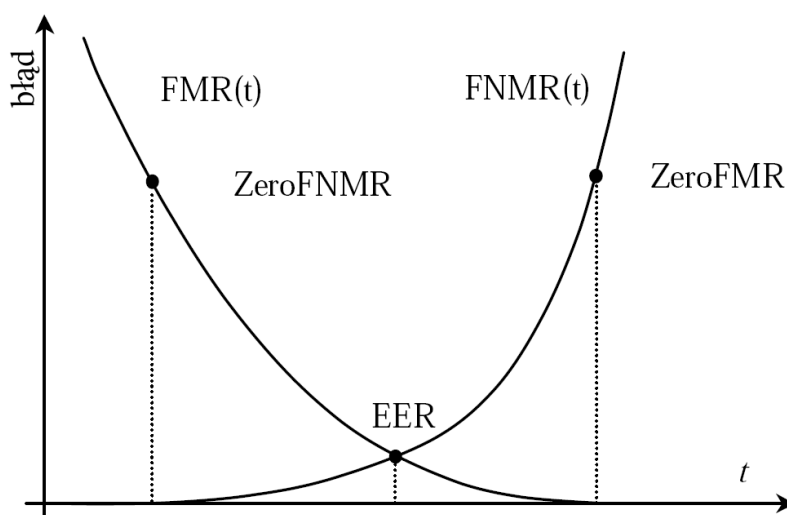
⁸ang. *false match rate*. W tej pracy zastosowano angielskie oznaczenia rodzajów błędów, ze względu na ich powszechne stosowanie w literaturze.

⁹ang. *false non-match rate*

$$FNMR = \int_0^t p(s|H_1 = \text{prawda})ds$$

$$FMR = \int_t^1 p(s|H_0 = \text{prawda})ds$$

W celu oszacowania skuteczności działania systemu biometrycznego należy zebrać wyniki generowane przez system dla par identycznych odcisków palców¹⁰ oraz wyniki generowane przez system dla par różniących się odcisków palców¹¹. FMR to procent par różnych odcisków, których stopień dopasowania jest większy od progu t . FNMR to procent par takich samych odcisków, których stopień dopasowania jest mniejszy od progu t . Widać, że FMR i FNMR są funkcjami t . Interpretację graficzną FMR i FNMR pokazano na Rysunku 1.3.



Rysunek 1.4: Przykład krzywych FMR(t) i FNMR(t) z zaznaczonymi EER, ZeroFNMR i ZeroFMR [1]

W praktyce często wprowadza się dodatkowe wskaźniki opisujące krzywe rozkładu prawdopodobieństw FMR i FNMR (Rysunek 1.4):

Equal-Error Rate (EER) : współczynnik określa wielkość błędu dla progu t , takiego, że $FMR(t) = FNMR(t)$. W praktyce funkcje FMR i FNMR nie są ciągłe, dlatego EER należy określać jako przedział.

¹⁰Rozkład prawdopodobieństwa $P(s | H_1 = \text{prawda})$ nazywany jest *rozkładem oryginalnym*.

¹¹Rozkład prawdopodobieństwa $P(s | H_0 = \text{prawda})$ nazywany jest *rozkładem oszusta*.

ZeroFNMR : najniższa wartość FMR, przy której nie zachodzą fałszywe niedopasowania.

ZeroFMR : najniższa wartość FNMR, przy której nie zachodzą fałszywe dopasowania.

1.3.2 Błędy systemów identyfikacji

Przyjmując pewne upraszczające założenia można opisać błędy systemu identyfikacji za pomocą definicji sformułowanych w poprzednim podrozdziale. Załóżmy, że w systemie identyfikacji nie stosuje się mechanizmu indeksowania, a więc każdy identyfikowany wzorzec musi zostać porównany z każdym spośród N szablonów zapisanych w bazie. Ponadto przyjmijmy, że dla każdego użytkownika w bazie danych zapisany jest dokładnie jeden szablon. Niech $FNMR_N$ i FMR_N oznaczają odpowiednio *false non-match rate* i *false match rate* dla systemu identyfikacji. Możemy je zdefiniować następująco:

$FNMR_N = FNMR$. Prawdopodobieństwo niedopasowania odcisku palca do szablonu jest takie same jak w przypadku systemu weryfikacji.

$FMR_N = 1 - (1 - FMR)^N$. Fałszywe dopasowanie zachodzi wtedy, kiedy odcisk zostanie błędnie dopasowany do jednego lub więcej szablonów zapisanych w bazie. FMR_N jest więc definiowane jako 1 zmniejszone o prawdopodobieństwo, że odcisk nie zostanie błędnie dopasowany do żadnego z odcisków znajdujących się w bazie. $(1 - FMR)$ oznacza prawdopodobieństwo, że odcisk nie zostanie błędnie dopasowany do jednego szablonu. $(1 - FMR)^N$ jest więc prawdopodobieństwem, że błędne dopasowanie nie zajdzie w stosunku do żadnego z N szablonów w bazie. Jeżeli FMR jest bardzo małe, to można zastosować przybliżenie: $FMR_N \approx N \cdot FMR$, co oznacza, że prawdopodobieństwo fałszywego dopasowania rośnie liniowo wraz ze wzrostem rozmiaru bazy danych.

Ostatni wniosek ma kolosalne znaczenie dla projektowania systemów identyfikacji. Oznacza on bowiem, że szybki wzrost FMR_N wraz ze wzrostem rozmiaru bazy danych jest znacznie większym problemem niż wzrost czasu obliczeń. Załóżmy przykładowo, że istnieje system identyfikacji posiadający 100000 użytkowników oraz, że zastosowano

w nim algorytm weryfikacji o $FMR = 10^{-5}$, a więc jedno błędne rozpoznanie na 10000 prób. Oznacza to, że prawdopodobieństwo błędnego zaakceptowania osoby podczas identyfikacji wynosi $FMR \approx 10\%$. W takim przypadku każdy miałby duże szanse na zidentyfikowanie jako użytkownik systemu jedynie poprzez próbę zidentyfikowania po kolei wszystkich swoich dziesięciu palców.

1.4 Podsumowanie

Jak zostało pokazane w tym rozdziale, wśród systemów biometrycznych dominują systemy oparte o linie papilarne. Dlatego to właśnie im należy poświęcić najwięcej uwagi. Poziom błędów metod przetwarzania oraz rozpoznawania danych ma decydujący wpływ na skuteczną pracę systemu biometrycznego. Oznacza to, że należy dążyć do zwiększania poprawności działania tych metod i algorytmów.

Rozdział 2

Metody i algorytmy rozpoznawania odcisków palców

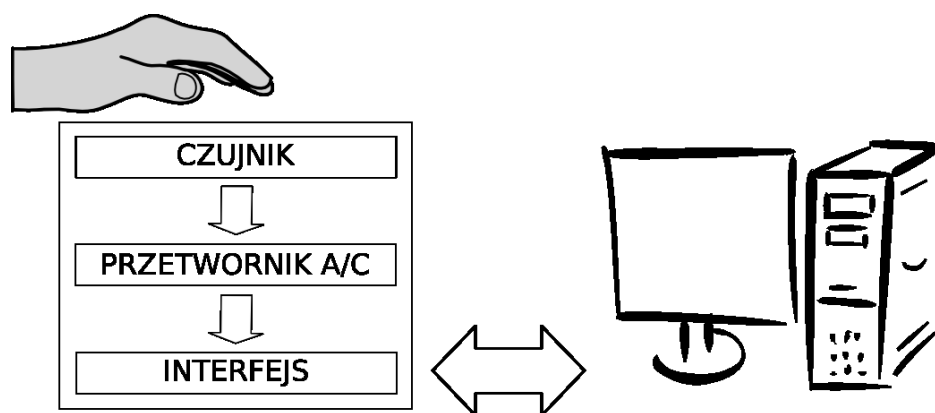
2.1 Pozyskiwanie odcisków palców

W tym rozdziale omówione zostaną wszystkie istotne aspekty związane z procesem pozyskiwania odcisków palców na potrzeby ich późniejszego rozpoznawania. Pierwszą historyczną metodą pobierania odcisków palców była metoda tuszu i kartki papieru. Osoba, której odciski pobierano, musiała umoczyć palec w tuszu, a następnie odcisnąć go na karcie¹. Odciski palców były później wprowadzane do komputerów za pomocą zwykłych skanerów optycznych, którymi skanowano karty z odcisniętymi liniami papilarnymi. Takie pozyskiwanie odcisków palców nazywane jest skanowaniem w trybie off-line. W dzisiejszych czasach powszechne stało się tzw. skanowanie na żywo (ang. *live scan*), polegające na pozyskiwaniu odcisków palców za pomocą specjalnych czytników. Czytnik taki nie wymaga stosowania tuszu i od razu przekształca zeskanowany obraz do postaci cyfrowej.

Pojawienie się tanich czytników cyfrowych spowodowało duży wzrost zainteresowania systemami opartymi o rozpoznawanie odcisków palców. Umożliwiło to wyjście technologii identyfikacji linii papilarnych poza laboratoria policyjne i wkroczenie do codziennego życia.

Ogólny schemat pozyskiwania odcisku palca za pomocą czytnika przedstawiony został na Rysunku 2.1. Użytkownik przykładą palec do czujnika, który rejestruje struktu-

¹Pierwszy wzór karty daktyloskopijnej opracował Francis Galton [9].



Rysunek 2.1: Schemat czytnika odcisków palców [1]

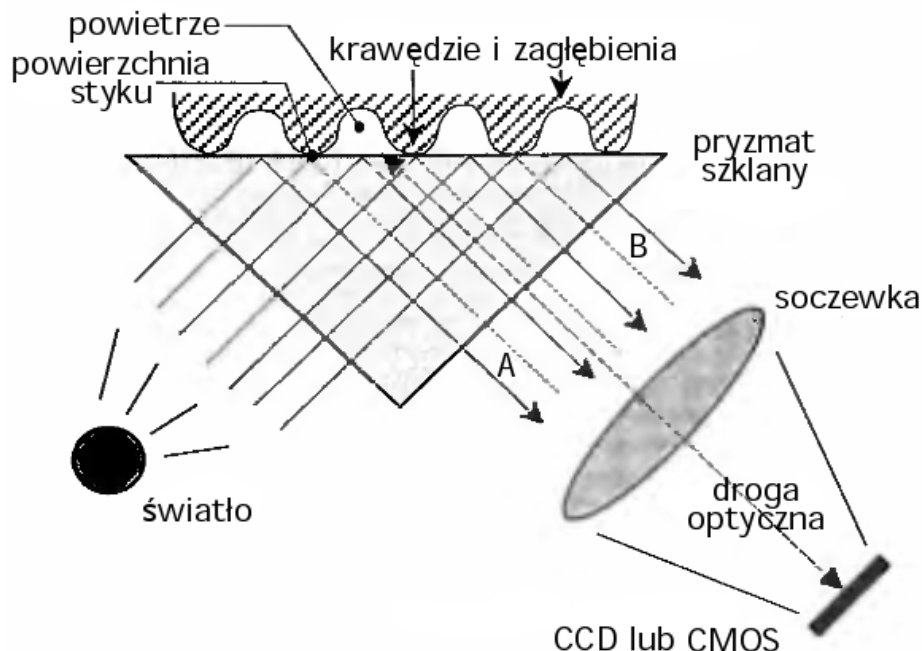
rę powierzchni palca. Przetwornik A/C² konwertuje dane z czujnika na postać cyfrową. Interfejs odpowiedzialny jest za komunikację czytnika z urządzeniami zewnętrznymi. Urządzeniem takim jest w szczególności komputer, służący zarówno do przechowywania pobranych odcisków jak i do ich przetwarzania. W przypadku odcisków palców postacią cyfrową danych odczytanych przez czytnik jest plik graficzny. Dlatego komputerowa analiza linii papilarnych jest zagadnieniem ściśle związanym z dziedziną przetwarzania oraz rozpoznawania obrazów.

Niektóre czytniki mogą odbiegać od pokazanego schematu, np. przetwornik A/C może być urządzeniem zewnętrznym. Należy też w tym miejscu rozróżnić stosowane w poniższej pracy pojęcia czytnika i czujnika. Przez *czujnik* określany jest jedynie element odpowiedzialny za rejestrację struktury powierzchni palca. Termin *czytnik* odnosi się zaś do całego urządzenia.

Czujnik jest najistotniejszym elementem systemu rozpoznającego odciski palców, gdyż to on tworzy graficzny obraz odcisku palca. Od jakości tego obrazu zależy później skuteczność rozpoznawania. Obecnie dostępne czytniki można podzielić na trzy kategorie w zależności od rodzaju używanego czujnika: optycznego, krzemowego lub ultradźwiękowego. W ramach tych kategorii można wyróżnić jeszcze bardziej zróżnicowane rodzaje czujników. Ich omówienie zaprezentowano poniżej.

Czujniki optyczne są najbardziej rozpowszechnionym typem czujników. Pierwotnie były bardzo drogie, jednak wraz z rozpowszechnianiem się technologii ich cena zmalała i jest porównywalna z cenami czytników innych typów.

²analogowo-cyfrowy



Rysunek 2.2: Schemat czujnika FTIR [1]

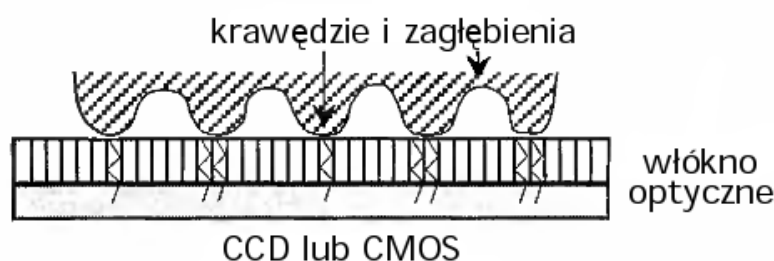
- *Całkowite rozproszone odbicie wewnętrzne*³: Jest to najstarsza i najpopularniejsza obecnie technika. Głównym elementem czujnika jest przyzmat. Z jednej strony jest on oświetlany, z drugiej zaś znajduje się światłoczuła matryca. Światło odbija się od wewnętrznej powierzchni przyzmatu, przechodzi przez soczewki ogniskujące i jest rejestrowane przez matrycę CCD⁴ lub CMOS⁵. Gdy użytkownik dotyka palcem do powierzchni przyzmatu, proces odbijania światła ulega zakłóceniu. Listewki skórne będące w bezpośrednim kontakcie z przyzmatem powodują zakłócenia w odbijaniu światła i są widoczne na zeskanowanym obrazie w kolorze ciemnym. Zagłębienia między listewkami skórnymi nie dotykają powierzchni, co powoduje normalne odbijanie się światła i w rezultacie jasny kolor na obrazie. Problemem tych czujników jest trapezoidalne zniekształcenie kształtu odcisku. Spowodowane jest to różną długością drogi przebywanej przez promienie w zależności od

³ang. *Frustrated Total Internal Reflection*, w literaturze często używa się skrótu FTIR.

⁴ang. *Charge-Coupled Device*

⁵ang. *Complementary Metal Oxide Semiconductor*, określenie pochodzi od techniki używanej do wytwarzania matrycy

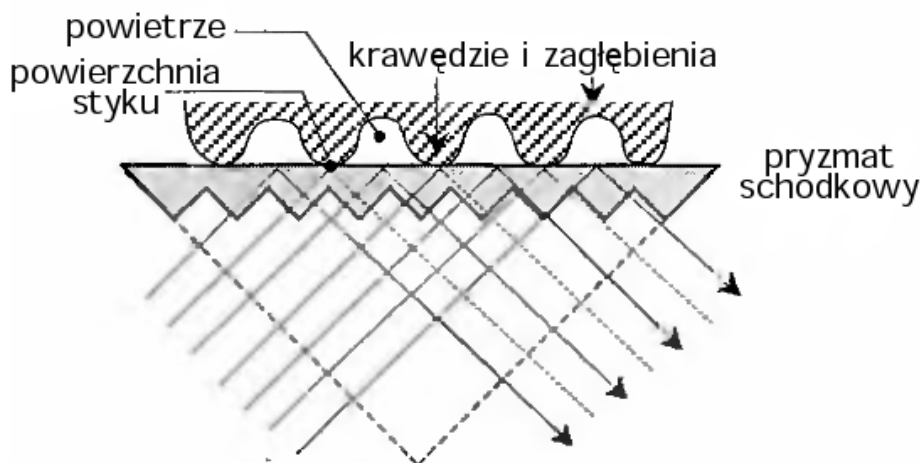
ich miejsca odbicia (patrz Rysunek 2.2, promienie A i B). Istnieją optyczne i programowe metody korekcji tego zniekształcenia. Czytników tego typu nie da się również znacząco zminiaturyzować, ponieważ zmniejszenie drogi pomiędzy powierzchnią pryzmatu, a matrycą wprowadza zniekształcenia na krawędziach obrazu. Niewątpliwą zaletą jest jednak to, że nie da się ich oszukać poprzez przyłożenie wydrukowanego odcisku palca, ponieważ tworzą one obraz palca na podstawie jego trójwymiarowej powierzchni.



Rysunek 2.3: Schemat czujnika z włóknami optycznymi [1]

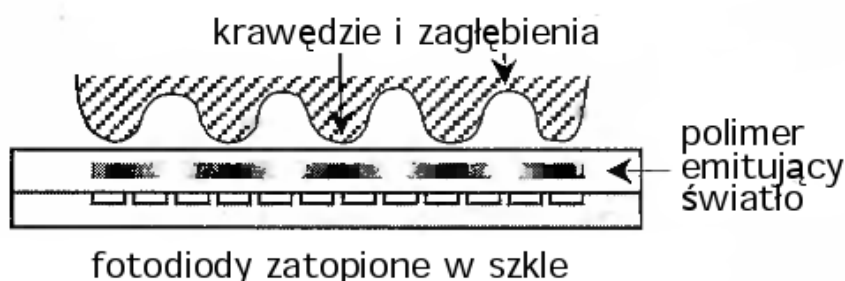
- *Włókna optyczne* (Rysunek 2.3): Zamiast pryzmatu stosuje się specjalną powierzchnię złożoną z włókien optycznych. Przewodzą one światło zmienne przez strukturę odcisku palca. Światłowoda matryca CCD lub CMOS przylega bezpośrednio po drugiej stronie tej powierzchni. Czujniki tego typu mogą być mniejsze od wymienionych wcześniej czujników FTIR. Ponieważ pomiędzy matrycą a włóknami optycznymi nie ma żadnych soczewek ogniskujących, matryca musi pokrywać cały obszar powierzchni skanowania. Z tego powodu, w przypadku produkcji czujników o dużej powierzchni, ich koszt może być wyższy niż czujników FTIR.
- *Całkowite rozproszone odbicie wewnętrzne przy użyciu pryzmatu schodkowego* (Rysunek 2.4): Ten typ czujnika różni się od zwykłego czujnika FTIR rodzajem stosowanego pryzmatu. Jest to pryzmat schodkowy, w rzeczywistości składający się z dużej liczby małych pryzmacików przylegających do siebie⁶. Pozwala to na redukcję rozmiarów urządzenia, jednak jakość obrazu uzyskana z takich czujników również jest gorsza.

⁶Pryzmat schodkowy jest skonstruowany na tej samej zasadzie co soczewka Fresnela [14].



Rysunek 2.4: Schemat czujnika FTIR z pryzmatem schodkowym [1]

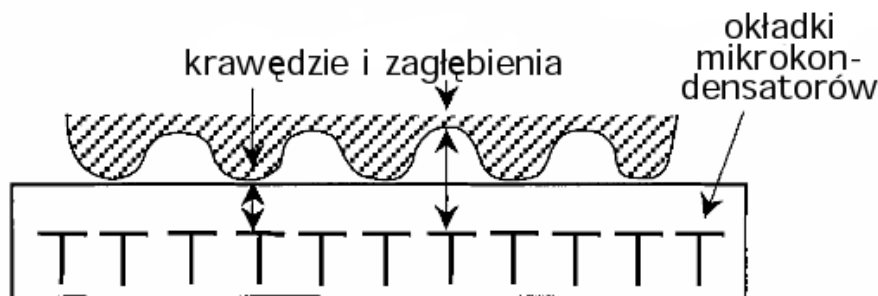
- *Czujniki elektrooptyczne* (Rysunek 2.5): Czujnik składa się z dwóch warstw. Pierwsza to polimer, który emituje światło w wyniku przyłożenia do niego napięcia. Dotknięcie palca powoduje zakłócenia w potencjale napięcia polimeru, przy czym inne zakłócenia powodują stykające się z jego powierzchnią listewki, a inne zagłębienia między nimi. Prowadzi to do różnic w natężeniu emitowanego światła. Druga warstwa czujnika to matryca złożona z fotodiod odpowiedzialnych za wytwarzanie cyfrowej reprezentacji otrzymanych sygnałów świetlnych. Niektóre produkowane czytniki wykorzystują tylko pierwszą warstwę, a zamiast drugiej stosują soczewki i wspomniane wcześniej matryce CMOS. Jakość odcisków pozyskiwanych tą techniką wciąż nie dorównuje jakością czytnikom FTIR.



Rysunek 2.5: Schemat czujnika elektrooptycznego [1]

- *Odczyt bezpośredni*: W tym typie czujnika stosuje się wysokiej jakości kamerę, która rejestruje z bliska powierzchnię palca. Problemem jest uzyskanie za pomocą tej techniki obrazów dobrej jakości.

Czujniki krzemowe zostały teoretycznie wymyślone w latach 80tych, jednak rozwiązania praktyczne pojawiły się dopiero w połowie lat 90tych. Miały być alternatywą dla drogich czujników optycznych, jednak obecnie różnica cen między tymi typami czujników zatarła się. Każdy czujnik krzemowy składa się z macierzy pikseli, z których każdy jest sam w sobie miniaturowym czujnikiem. Nie jest wymagane wykorzystanie żadnych elementów optycznych. Obecnie istnieją cztery metody przekształcania informacji fizycznej na impulsy elektryczne.

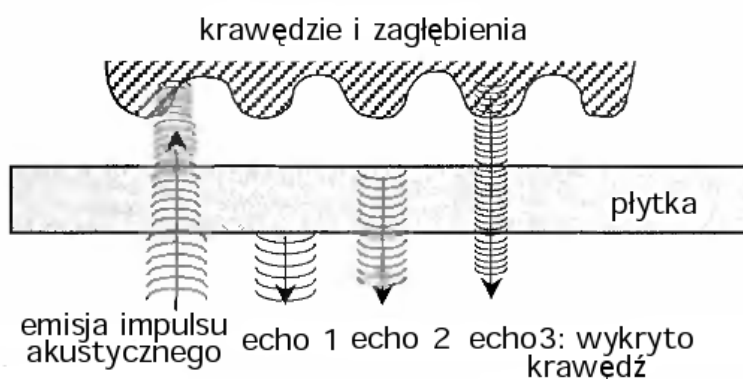


Rysunek 2.6: Schemat czujnika pojemnościowego [1]

- *Pojemnościowe* (Rysunek 2.6): Jest to najpopularniejszy typ czujnika wśród czujników krzemowych. W płytce krzemowej zatapia się okładki kondensatorów (tylko jeden biegun). Drugą okładką staje się palec przyłożony do czujnika. Pomędzy okładkami zatopionymi w płytce a palcem tworzy się potencjał elektryczny. Jego wielkość zależy od odległości od skóry, a więc jest inny dla listewek, a inny dla zagłębień. Na tej podstawie tworzony jest obraz odcisku palca. Podobnie jak czujniki FTIR, czujniki pojemnościowe także tworzą obraz palca na podstawie jego trójwymiarowej struktury i nie da się ich oszukać za pomocą wydrukowanych obrazów.
- *Termiczne*: Zbudowane są z materiału, który wytwarza prąd w wyniku różnic temperatury. Stykające się z powierzchnią czujnika listewki skórne wy-

tworzącą inną różnicę temperatur niż zagłębienia, co jest podstawą do wytworzenia graficznej reprezentacji linii papilarnych. Powierzchnia czujnika jest z reguły podgrzewana za pomocą prądu w celu zwiększenia różnicy temperatur między powierzchnią rejestrującą a palcem.

- *Pole elektryczne*: Czujnik składa się z generatora sygnału sinusoidalnego i małych anten rejestrujących ten sygnał zmodulowany przez nierówną powierzchnię skóry. Na podstawie sygnału zarejestrowanego przez każdą antenę wyliczany jest obraz odcisku palca.
- *Piezoelektryczne*: Wykorzystuje się czujniki czułe na nacisk, które w wyniku ściśnięcia wytwarzają prąd elektryczny (tzw. efekt piezoelektryczny). Powierzchnia czujnika pokryta jest materiałem nieprzewodzącym. Po dotknięciu palca do czujnika listewki i zagłębienia znajdują się w różnej odległości od czujników, a tym samym czujniki te generują prąd o różnym natężeniu w zależności od stopnia nacisku. Niestety w praktyce często nie da się wykryć różnicy pomiędzy wgłębieniami i wypukłościami linii papilarnych, a powłoka ochronna czytnika dodatkowo zniekształca obraz.



Rysunek 2.7: Schemat czujnika ultradźwiękowego [1]

Czujniki ultradźwiękowe wysyłają w stronę palca falę akustyczną, a następnie rejestrują jej echo (Rysunek 2.7). Na podstawie echa obliczane są odległości pomiędzy powierzchnią czujnika a poszczególnymi rejonami palca. Pozwala to na

wytworzenie obrazu linii papilarnych. Fale dźwiękowe wnikają na pewną głębokość w skórę, są nawet w stanie przeniknąć cienkie rękawiczki. Dlatego metoda ta odporna jest na zakłócenia spowodowane przez mechaniczne zabrudzenia palców i pozwala na uzyskanie obrazów bardzo dobrej jakości. Problemem jest duży rozmiar czytników zawierających elementy mechaniczne oraz ich wysoki koszt.

Wybór konkretnego rodzaju czytnika dla systemu biometrycznego jest wypadkową wielu czynników. Najważniejszym z nich jest z reguły jakość obrazu odcisku palca oferowanego przez czytnik. Jakość tą można opisać za pomocą kilku składowych:

- *Rozdzielczość*: Decyduje o ilości szczegółów widocznych na obrazie. FBI uznaje 500 dpi za minimalną dopuszczalną rozdzielczość (większość dostępnych na rynku czytników jest w stanie ją zapewnić). Rozdzielczość w granicach 250-300 dpi uznaje się za minimalną, przy której możliwe jest wykrywanie minucji⁷. Obrazy uzyskane przy zastosowaniu mniejszej rozdzielczości wykorzystuje się z reguły przy stosowaniu technik korelacyjnych.
- *Obszar*: Jest to kluczowy parametr czujnika, który określa jaki obszar odcisku palca zostanie zeskanowany. FBI w swoich specyfikacjach określa wymagany obszar skanowania jako 1 na 1 cal. Im obszar większy tym kompletniejszy obraz linii papilarnych i tym łatwiej jest go dopasować do wzorca. W czytnikach przeznaczonych na rynek systemów weryfikacyjnych obszar skanowania jest z reguły zmniejszany w celu obniżenia ich kosztów. Powoduje to problemy z późniejszym zeskanowaniem przez użytkownika tego samego fragmentu palca, a tym samym z jego prawidłową weryfikacją.
- *Głębina kolorów*: Określa ilość bitów przeznaczonych do zakodowania każdego piksela. Przy rozpoznawaniu odcisków palców kolor nie ma znaczenia i dlatego większość czytników pozyskuje obraz w skali szarości. Specyfikacja FBI sugeruje stosowanie głębokości 8 bitów na piksel, co odpowiada 256 odcieniom szarości. W praktyce wiele czujników pozyskuje dane o głębokości 2 lub 3 bity i przekształca je programowo na 8 bitów.

⁷Większość algorytmów rozpoznawania odcisków palców bazuje na korelacji w położeniu minucji. Zostanie to szczegółowo omówione w Rozdziale 2.2.

- *Dokładność geometryczna*: Istotne jest by pozyskany odcisk nie był zniekształcony. Większość czytników wprowadza pewien stopień zniekształcenia, który powinien zostać skorygowany.

W praktyce pojawiają się różne problemy związane z pozyskiwaniem odcisków palców. Są nimi zanieczyszczenia, zarówno powierzchni czytników jak i skanowanych palców. W przypadku osób pracujących fizycznie oraz osób starszych linie papilarne są dużo mniej wyraźne i czytnik może mieć problemy z prawidłowym rozróżnieniem listewek skórnych od zagłębień. Analogiczny problem występuje, gdy palce są wilgotne (np. od potu), zbyt suche albo zostaną dociśnięte do czujnika zbyt słabo lub zbyt mocno. W takich przypadkach uzyskane obrazy są często bardzo niskiej jakości. Z drugiej strony, wiele czytników optycznych dostarcza obrazy bardzo wysokiej jakości na których widoczne są pory w skórze znajdujące się na liniach papilarnych. Widoczność porów również może prowadzić do zakłóceń w trakcie rozpoznawania.

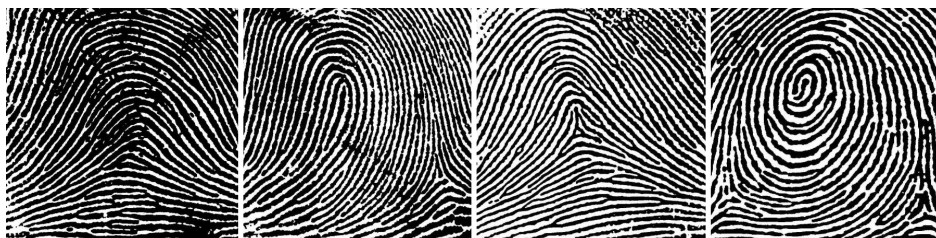
2.2 Cechy odcisków palców

W poprzednim rozdziale zostało powiedziane, że odciski palców dwóch różnych osób nie mogą być identyczne. Stwierdzenie o braku identyczności rozumiane jest intuicyjnie. Kiedy jednak zachodzi potrzeba stworzenia usystematyzowanej metody rozpoznawania, musi pojawić się pytanie o istnienie charakterystycznych cech pozwalających na rozróżnianie odcisków między sobą. Przed zagadnieniem tym stanęła daktyloskopia ponad 100 lat temu, a jej dokonania – choć nie wszystkie – zostały później przeniesione do automatycznych systemów rozpoznawania linii papilarnych. W przypadku odcisków palców można wyróżnić kilka rodzajów cech. W daktyloskopii w pierwszej kolejności analizuje się pewne cechy globalne, takie jak ogólny układ linii i stopień ich wygięcia. Wśród globalnych wzorów tworzonych przez linie wyróżnia się m.in. łuki (pierwszy i trzeci obrazek na Rysunku 2.8) czy pętlice (obrazek drugi na Rysunku 2.8). Ponadto wyróżnia się charakterystyczne punkty⁸ w odcisku palca, takie jak delta (punkt, w którym zbiegają się linie z trzech stron) czy rdzeń⁹ (punkt otoczony naokoło przez linie). Te cechy nie są jednak wystarczające do jednoznacznego opisanie i rozpoznania odcisku. Dlatego analizuje się także minucje, czyli charakterystyczne zmiany w przebiegu

⁸ang. *singularities*

⁹ang. *core*

linii papilarnych, m.in. rozwidlenia¹⁰, zakończenia¹¹, oczka i haczyki (Rysunek 2.9). Dopiero wzajemny układ minucji pozwala na jednoznaczne rozpoznanie odcisku.



Rysunek 2.8: Przykłady niektórych struktur linii papilarnych wyróżnianych w daktyloskopii [16]



Rysunek 2.9: Przykłady niektórych minucji wyróżnianych w daktyloskopii. Linie papilarne widoczne w kolorze czarnym, zagłębienia w kolorze białym [15]. Górny rząd od lewej: zakończenie, rozgałęzienie, haczyk. Dolny rząd od lewej: fragment linii, wysepka, oczko

W automatycznym rozpoznawaniu odcisków palców stosuje się nieco uproszczone metody. W przypadku prostych aplikacji dokonujących jedynie weryfikacji użytkownika można pominąć analizę ogólnej struktury (a więc znajdowanie delt itd.) i oprzeć się jedynie na samych minucjach. Spowodowane jest to tym, że w takim systemie dokonuje się jedynie porównań jeden-do-jednego. W przypadku systemów identyfikujących, a więc dokonujących porównań wiele-do-jednego, sam układ minucji również

¹⁰ang. *bifurcation*

¹¹ang. *termination*

wystarczyłyby do jednoznacznej identyfikacji palca. Jednak, biorąc pod uwagę ilość porównań jakiej taki system musiałby dokonywać, często analizuje się ogólną strukturę odcisku i wykorzystuje ją jako funkcję szufladkującą. Pozwala to na znaczny wzrost wydajności systemu poprzez ograniczenie ilości porównań. Proces znajdowania minucji w automatycznych systemach rozpoznawania odcisków palców również jest upraszczany, ponieważ ogranicza się jedynie do szukania dwóch rodzajów minucji: rozgałęzień i zakończeń linii. Jest to w zupełności wystarczające ponieważ wszystkie inne minucje są kombinacją tych dwóch.

Mówiąc o zakończeniach i rozwidleniach linii należy zwrócić uwagę na pewien rodzaj dualizmu. Zakończenie linii papilarnej jest jednocześnie rozwidleniem zagłębienia między liniami i odwrotnie – rozwidlenie linii oznacza zakończenie zagłębienia między tymi liniami. Wykorzystuje się to przy wyznaczaniu kierunku minucji. W przypadku zakończenia linii kierunek określony jest kierunkiem linii w punkcie jej zakończenia. Dla rozgałęzień kierunek określa się jako kierunek kończącego się zagłębienia.

2.3 Ekstrakcja cech

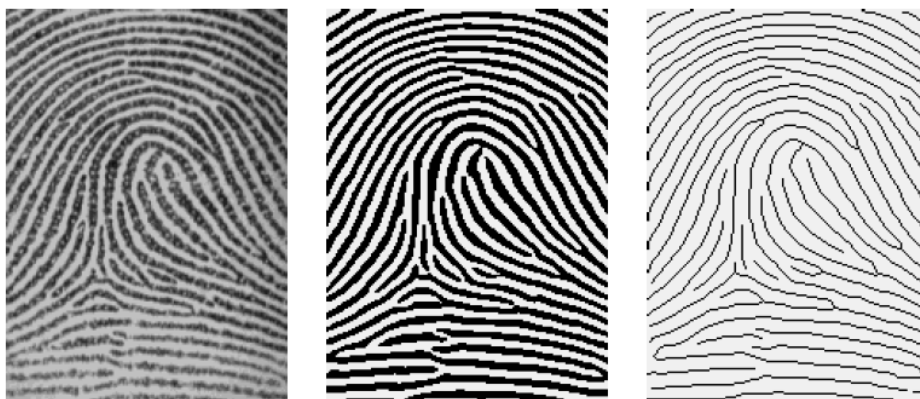
W literaturze dotyczącej automatycznego rozpoznawania odcisków palców zaproponowano wiele metod i algorytmów znajdowania minucji¹². Ich przegląd zostanie zaprezentowany w tym rozdziale. Algorytmy te mają kluczowe znaczenie, ponieważ od poprawnego zidentyfikowania minucji zależy poprawność rozpoznawania odcisku palca.

2.3.1 Metody oparte o operacje morfologiczne

Wiele początkowo proponowanych metod opierało się na dosyć złożonym przetwarzaniu wstępnym (Rysunek 2.10). Wymagało ono w pierwszym kroku przekształcenia obrazu ze skali szarości na postać binarną. Drugi krok polegał na ścienianiu linii papilarnych do szerokości jednego piksela poprzez stosowanie na obrazie operacji morfologicznych. W tak uzyskanym obrazie analizowano przebieg linii, by wykryć minucje. O ile samo wykrywanie minucji w takim obrazie jest zagadnieniem mało skomplikowanym, o tyle stosowanie operacji morfologicznych przysparza większych trudności.

¹²Istnieją algorytmy, które porównują odciski palców na podstawie korelacji. Nie wymagają one tym samym etapu ekstrakcji cech.

Największym problemem jest zapewnienie prawidłowego działania procesu ścieniania. Błędy mogą bowiem prowadzić do utraty części danych o minucjach, bądź do powstania nowych, fałszywych minucji. Metoda ta była więc obciążona dużym kosztem obliczeniowym na etapie przetwarzania wstępnego oraz dużym ryzykiem błędu.



Rysunek 2.10: Kolejne etapy przetwarzania wstępnego w algorytmie opartym o operacje morfologiczne

2.3.2 Metody oparte o analizę linii

Niedogodności metod wykorzystujących binaryzację i operacje morfologiczne spowodowały potrzebę zaprojektowania metod działających z większą skutecznością i bez potrzeby stosowania złożonego przetwarzania wstępnego mogącego prowadzić do utraty części danych. Doprowadziło to do powstania algorytmów, które wykrywają minucje poprzez śledzenie przebiegu linii papilarnych na obrazie, często bez potrzeby jakiegokolwiek przetwarzania wstępnego¹³. W tej pracy wybrano jako punkt wyjścia algorytm należący do tej właśnie grupy. Został on omówiony w [1] oraz [2]. Poszczególne etapy działania algorytmu zaprezentowano poniżej.

Segmentacja odcisku z tła

Nim przystąpi się do analizy układu linii papilarnych należy zlokalizować obszar zeskanowanego odcisku. Wybór konkretnej metody segmentacji może zależeć od postaci

¹³Oczywiście potrzeba wstępnego przetwarzania zależy mocno od jakości obrazu dostarczonego przez czytnik.

obrazu uzyskanego z czytnika. W tej pracy zaproponowano rozwiązanie dostosowane do czytnika omówionego w Rozdziale 4.5. Zaproponowany algorytm segmentacji omówiono w Rozdziale 3.2.1.

Obliczanie mapy kierunków

Kolejnym z etapów jest obliczenie mapy kierunków¹⁴. Określa ona lokalny zwrot linii papilarnych. Informacja ta jest wymagana na późniejszym etapie działania algorytmu. Większość algorytmów wyznaczania kierunku krawędzi opartych jest o metody gradientowe. W tej pracy wykorzystano jedną z takich metod przedstawioną szczegółowo w [12].

Obraz traktowany jest jako powierzchnia trójwymiarowa opisana jakąś funkcją. Wartość każdego piksela interpretowana jest jako spróbkowana i skwantyzowana wartość w danym punkcie. Przez *okno kierunkowe* określimy kwadratowy obszar o boku długości α . Niech centralnym punktem okna będzie piksel o współrzędnych (x_0, y_0) . Dla każdego punktu należącego do okna kierunkowego wyznaczymy wektor n_{ij} prostopadły do powierzchni funkcji w tym miejscu. Wektor kierunkowy dla każdego piksela (x_i, y_j) należącego do okna leży w płaszczyźnie xy i jest prostopadły do wektora n_{ij} . Wektor kierunkowy dla danego okna kierunkowego również leży w płaszczyźnie xy i jest to wektor "najbardziej prostopadły" do wszystkich wyliczonych wektorów n_{ij} w sensie minimalizacji średniokwadratowej.

Chcąc wyliczyć wektor n_{ij} dla danego piksela (x_i, y_j) , weźmy piksele sąsiadujące z danym punktem po przekątnej: (x_{i+1}, y_{j+1}) , (x_{i-1}, y_{j+1}) , (x_{i-1}, y_{j-1}) i (x_{i+1}, y_{j-1}) . Ich poziomy jasności oznaczmy odpowiednio jako a_1 , a_2 , a_3 i a_4 . Każdy wektor normalny zdefiniowany jest jako $n_{ij} = [a_{ij}, b_{ij}, 1]$ gdzie:

$$a_{ij} = \frac{-a_1 + a_2 + a_3 - a_4}{4} \quad , \quad b_{ij} = \frac{-a_1 - a_2 + a_3 + a_4}{4} \quad .$$

Oznaczmy jako $v_{ij} = (a_{ij}, b_{ij})$ wektory powstałe przez usunięcie z wektorów n_{ij} składowej z . Niech $t = (t_1, t_2)$ będzie poszukiwanym wektorem kierunkowym dla danego okna kierunkowego. Wyznaczamy t metodą najmniejszych kwadratów, co można zapisać formalnie jako:

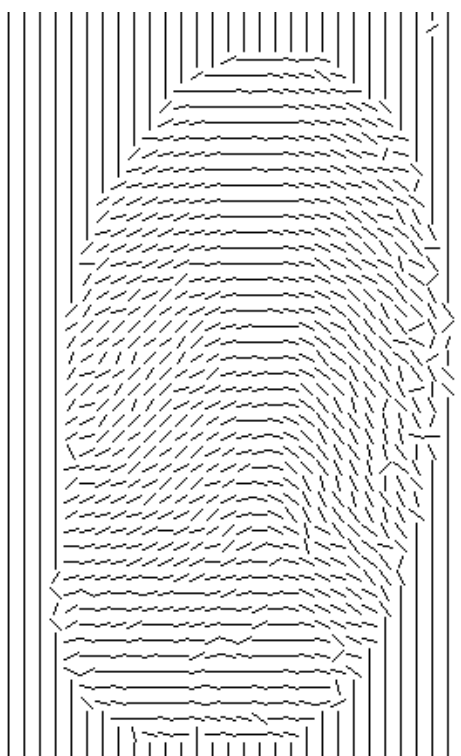
¹⁴ang. *directions map, orientation image*

$$\min \sum_{\substack{i=1..n \\ j=1..n}} |\langle v_{ij}, t \rangle|^2, \quad \text{przy założeniu, że } \|t\| = 1.$$

Pomijając szczegółowe wyprowadzenie, zapisujemy rozwiązanie jako:

$$A = \sum_{\substack{i=1..n \\ j=1..n}} (a_{ij})^2 \quad B = \sum_{\substack{i=1..n \\ j=1..n}} (b_{ij})^2 \quad C = \sum_{\substack{i=1..n \\ j=1..n}} a_{ij}b_{ij}$$

$$t = \begin{cases} \left[1, \frac{B-A}{2C} - \operatorname{sgn}(C) \sqrt{\left(\frac{B-A}{2C}\right)^2 + 1} \right] & \text{jeżeli } C \neq 0 \\ [1, 0] & \text{jeżeli } C = 0, A \leq B \\ [0, 1] & \text{jeżeli } C = 0, A > B \end{cases}.$$



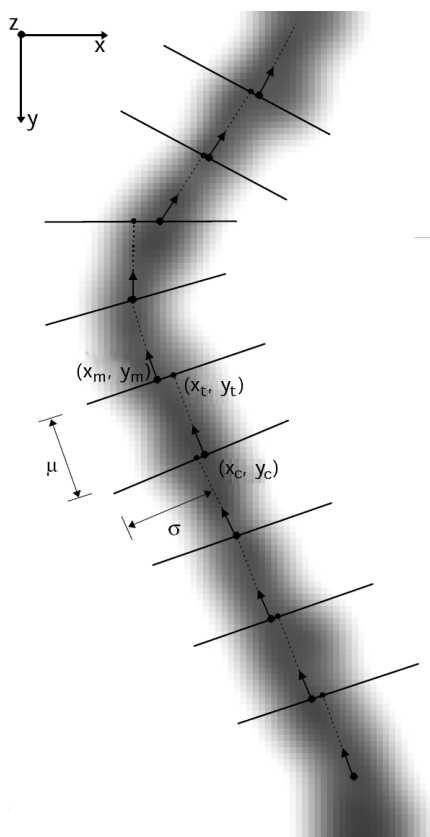
Rysunek 2.11: Mapa kierunków dla przykładowego odcisku palca

Szukany kąt możemy łatwo wyliczyć jako:

$$\varphi_0 = \begin{cases} \arctan\left(\frac{t_2}{t_1}\right) & \text{jeżeli } t_1 \neq 0 \\ \frac{\pi}{2} & \text{w przeciwnym przypadku} \end{cases}.$$

Śledzenie przebiegu linii

Mając przygotowaną mapę kierunków oraz znając obszar zajmowany przez odcisk palca można rozpocząć główny etap algorytmu. Jego idea polega na analizie po kolei wszystkich linii papilarnych znajdujących się na obrazie. Algorytm podąża za każdą linią do momentu, kiedy przetnie się ona z inną – co odpowiada minucji rozgałęziającej – lub skończy się – co oznacza minucję kończącą.



Rysunek 2.12: Kolejne etapy śledzenia linii papilarnej [2]

Obraz reprezentowany jest jako dwuwymiarowa tablica, której poszczególne komórki reprezentują poziomy jasności pikseli. Krawędzie odcisku palca oznaczone są kolorem ciemniejszym, a zagłębienia między krawędziami kolorem jaśniejszym. Algorytm rozpoczyna działanie z punktu (x_s, y_s) , w którym lokalny kierunek linii papilarnych wynosi φ_0 . W każdym kolejnym kroku przecina śledzoną linię prostopadłą sekcją (oznaczymy Ω) o długości $2\sigma + 1$, a następnie znajduje minimum (x_m, y_m) tej sekcji. Na podstawie lokalnego kierunku φ_c linii papilarnych w znalezionym punkcie, algorytm przemieszcza

się wzdłuż linii z punktu (x_m, y_m) o μ pikseli (Rysunek 2.12). Przebyty kawałek linii jest oznaczany, aby zapobiec jego ponownemu przechodzeniu w przyszłości oraz aby umożliwić wykrywanie przecięć z innymi liniami. Iteracyjny proces jest kontynuowany do momentu spełnienia warunków stopu. Zapis w pseudokodzie jest następujący:

```

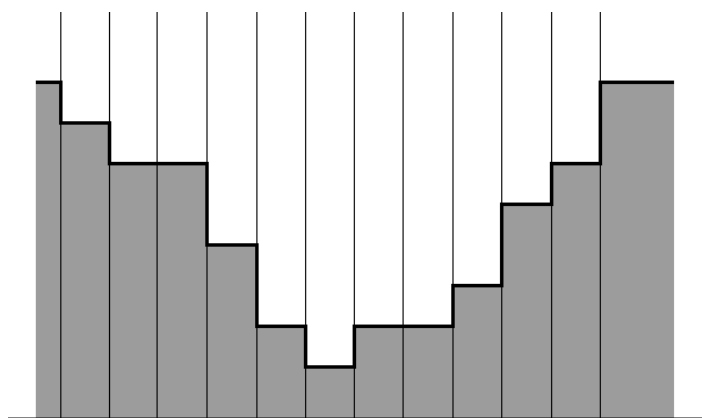
śledź_linie( $x_s, y_s, \varphi_0$ ) {
    czy_koniec := false;
    ( $x_c, y_c$ ) := ( $x_s, y_s$ );
     $\varphi_c$  :=  $\varphi_0$ ;
    while( !czy_koniec ) {
        ( $x_t, y_t$ ) := ( $x_c, y_c$ ) +  $\mu$  pikseli w kierunku  $\varphi_c$ ;
         $\Omega$  := sekcja ze środkiem w punkcie ( $x_t, y_t$ ) o długości  $2\sigma + 1$  i kierunku  $\varphi_c + \pi/2$ ;
        ( $x_m, y_m$ ) := minimum w  $\Omega$ ;
        czy_koniec := sprawdź warunki stopu dla ( $x_c, y_c$ ), ( $x_t, y_t$ ) i ( $x_m, y_m$ );
        oznacz odcinek od ( $x_c, y_c$ ) do ( $x_m, y_m$ ) jako przebyty;
        ( $x_c, y_c$ ) := ( $x_m, y_m$ );
         $\varphi_c$  := kierunek krawędzi w ( $x_c, y_c$ );
    }
}
    
```

Poszczególne kroki algorytmu zostały szczegółowo omówione poniżej.

Tworzenie sekcji

Sekcję linii papilarnej wyznacza się poprzez przecięcie obrazu płaszczyznę prostopadłą do linii w danym punkcie. Sekcja Ω jest jednoznacznie określona przez jej punkt centralny (x_t, y_t) , długość $2\sigma + 1$ i kierunek ϕ , który jest prostopadły do kierunku linii papilarnej w punkcie (x_t, y_t) . Formalnie, sekcja jest ciągiem punktów leżących na odcinku łączącym punkty krańcowe sekcji (Rysunek 2.13). Punkty te wyznacza się z wzorów:

$$\begin{aligned}
 (x_{start}, y_{start}) &:= (\lfloor x_t - \sigma \cdot \cos \phi \rfloor, \lfloor y_t - \sigma \cdot \sin \phi \rfloor) \quad , \\
 (x_{koniec}, y_{koniec}) &:= (\lceil x_t + \sigma \cdot \cos \phi \rceil, \lceil y_t + \sigma \cdot \sin \phi \rceil) \quad .
 \end{aligned}$$



Rysunek 2.13: Graficzna reprezentacja sekcji

Ilość pikseli w sekcji, na podstawie tak wyliczonych punktów krańcowych, wyniesie około $2\sigma + 1$ ¹⁵. Kolejne piksele sekcji zostają wyznaczone za pomocą algorytmu rysowania linii Bresenhama [17].

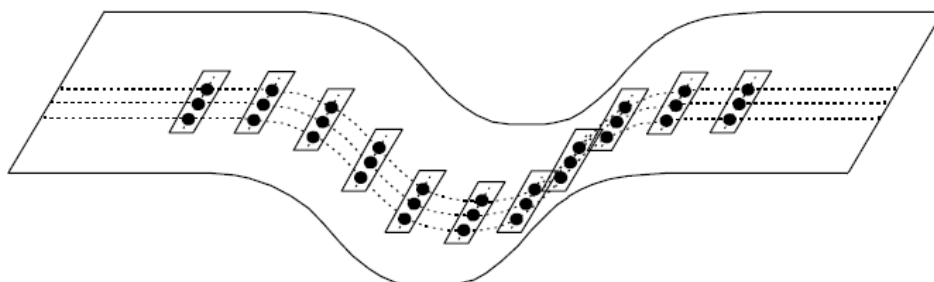
Wyznaczanie minimum sekcji

Dysponując ciągiem pikseli stanowiącym sekcję należy wyznaczyć jej minimum, będące środkiem linii papilarnej. Można tego dokonać bardzo łatwo, poprzez porównanie poziomów jasności poszczególnych pikseli należących do sekcji. Autorzy pracy [2] zwracają jednak uwagę, że takie podejście jest podatne na zakłócenia i nadaje się tylko do stosowania w obrazach wysokiej jakości. Zdarza się bowiem, że w miejscu, w którym powinno znajdować się minimum w wyniku zakłócenia znajduje się wartość większa od wartości sąsiednich pikseli. Może to być też wynikiem widoczności porów skórnych na zeskanowanym odcisku. Autorzy wspomnianej pracy proponują więc poprawę jakości sekcji w dwóch krokach.

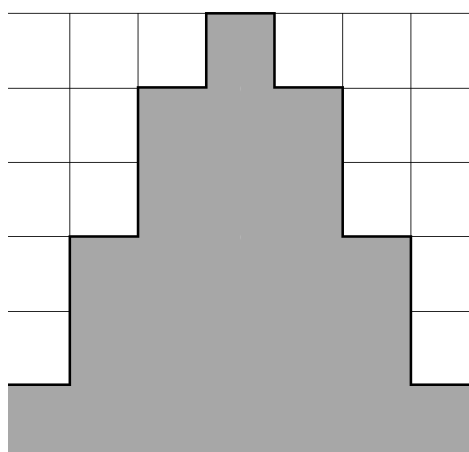
Pierwszy krok polega na uśrednieniu wartości pikseli w sekcji na podstawie wartości pikseli w sekcjach sąsiednich (Rysunek 2.14). Osiągamy to poprzez przecinanie linii papilarnej za pomocą $2l + 1$ równoległych sekcji (gdzie $l \geq 0$). Poziom jasności każdego piksela w centralnej sekcji określany jest jako średnia arytmetyczna odpowiednich pikseli ze wszystkich sekcji.

Krok drugi oparty jest na splocie z maską przypominającą kształtem funkcję Gaussa (Rysunek 2.15). Oznaczmy punkty należące do sekcji Ω jako $(x_1, y_1) \dots (x_n, y_n)$.

¹⁵Błędy zaokrągleń mogą doprowadzić do wahań długości sekcji w zakresie jednego piksela.



Rysunek 2.14: Graficzna reprezentacja trzech sekcji równoległych [2].
 Prostokątami zaznaczono grupy uśrednianych pikseli



Rysunek 2.15: Graficzna reprezentacja maski stosowanej w kroku drugim poprawy jakości sekcji. W tym przypadku $k = 3$,
 $m = [\frac{1}{24}, \frac{3}{24}, \frac{5}{24}, \frac{6}{24}, \frac{5}{24}, \frac{3}{24}, \frac{1}{24}]$

Ponadto oznaczmy poziomy jasności tych pikseli wyznaczone w pierwszym kroku jako $p(x_1, y_1) \dots p(x_n, y_n)$. Oznaczmy elementy maski jako m_k , gdzie $k = 1, \dots, 2j + 1$ oraz $j \geq 1, m_k \geq 0, \sum m_k = 1$. Nowe poziomy jasności pikseli oblicza się według wzoru:

$$p'(x_k, y_k) = \frac{1}{2j + 1} \sum_{v=-j}^j m_{j+1+v} \cdot p(x_{k+v}, y_{k+v}) \quad , \quad \text{gdzie } k = j + 1, \dots, n - j \quad .$$

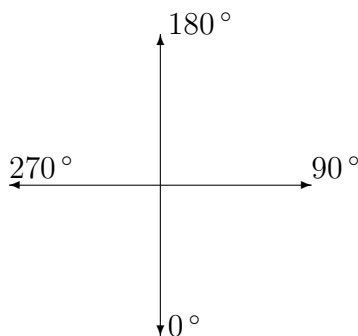
Splot ten odpowiada obliczeniu nowej wartości każdego piksela sekcji za pomocą średniej ważonej, gdzie wagi są określone przez wartości elementów maski.

Dysponując tak poprawioną sekcją można wyznaczyć minimum poprzez porówna-

nie poziomów jasności poszczególnych pikseli. W przypadku gdy minimum nie jest właściwe, należy wybrać piksel znajdujący się najbliżej środka sekcji.

Wyznaczanie kierunku linii

W każdym kroku algorytm posuwa się z punktu (x_m, y_m) o μ pikseli w kierunku φ_c określającym lokalny zwrot linii papilarnych. Sposób wyliczania tego kierunku został wcześniej omówiony. Należy jednak pamiętać o konieczności wybrania układu odniesienia i dostosowania zaprezentowanego algorytmu do tego układu. W tej pracy zdecydowano się na układ współrzędnych zaprezentowany na Rysunku 2.16.



Rysunek 2.16: Zastosowany układ odniesienia

Obliczanie kierunków dla wszystkich pikseli obrazu jest procesem czasochłonnym. Stosuje się więc usprawnienie, które polega na wyliczeniu kierunków dla dyskretnej siatki pikseli. Następnie stosuje się interpolację liniową w celu określenia kierunku dla pikseli umiejscowionych pomiędzy węzłami tej siatki.

Oznaczanie przebytych linii

Każda przeanalizowana linia musi zostać zapamiętana. Jest to konieczne w celu wykrywania przecięć linii papilarnych, jak również, by zagwarantować, że żadna linia nie będzie odwiedzona więcej niż raz. By zapamiętywać odwiedzone linie tworzy się dodatkowy obraz odpowiadający analizowanemu obrazowi odcisku palca. Na tym dodatkowym obrazie w każdym kolejnym kroku rysuje się linię łączącą minimum sekcji z poprzedniego kroku algorytmu z minimum nowej sekcji. Linia łącząca te dwa punkty musi mieć szerokość odpowiadającą szerokości linii papilarnych na zeskanowanym obrazie odcisku palca.

Warunki stopu

Śledzenie linii może zakończyć się z powodu następujących warunków stopu:

- *Wyjście poza odcisk palca*: Nowy punkt (x_t, y_t) leży poza obszarem wysegmentowanym z tła, co oznacza, że linia opuszcza zeskanowany obszar odcisku palca.
- *Zakończenie linii*: Minimum sekcji Ω , a więc punkt (x_t, y_t) tworzy zbyt duży kąt względem kierunku linii w punkcie (x_c, y_c) . Oznacza to, że linia się kończy, a więc wykryto minucję.
- *Przecięcie z inną linią*: Nowy punkt (x_t, y_t) był już wcześniej odwiedzony. Oznacza to, że dwie linie papilarne się łączą, a więc wykryto minucję.
- *Nadmierne wyginanie linii*: W trakcie śledzenia linii zapamiętywane są kierunki, w których przemieszczano się w kilku ostatnich krokach. Jeśli nowy kierunek odbiega za bardzo od średniego kierunku, w którym poruszano się w ostatnich krokach, śledzenie linii zostaje przerwane. Uzasadnieniem takiego podejścia jest to, że linie papilarne wyginają się łagodnie i jeśli na krótkim odcinku wystąpi nagle zmiana kierunku oznacza to najczęściej błąd w procesie śledzenia linii.

Wykrywanie minucji

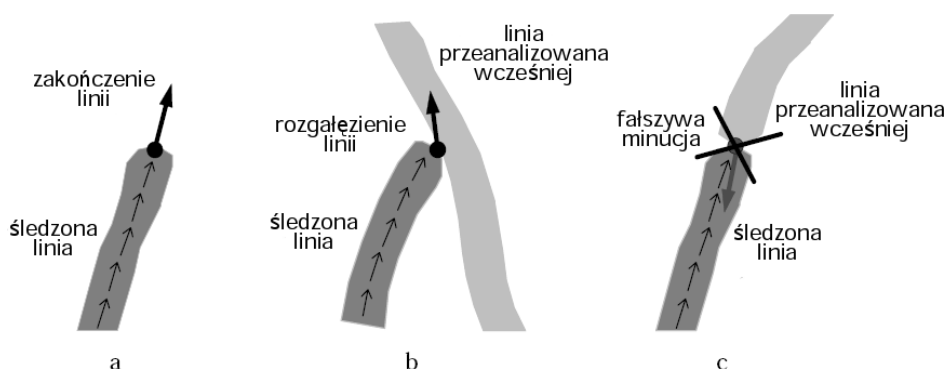
We wcześniejszych punktach zaprezentowany został algorytm ekstrakcji pojedynczej linii papilarnej, mając dany na starcie podany punkt oraz kierunek początkowy. W przypadku, gdy śledzona linia skończy się lub przetnie z linią już prześledzoną, algorytm zatrzymuje swoje działanie i zwraca parametry znalezionej minucji. Stajemy teraz przed zagadnieniem wykrycia wszystkich minucji danego odcisku palca. Jednocześnie trzeba się liczyć z możliwością pojawienia się fałszywych – błędnie wykrytych – minucji, a co za tym idzie, z koniecznością wykrycia takiej sytuacji i usunięcia nieprawidłowości. Ogólny algorytm można więc zapisać w postaci następującego pseudokodu:

```
znajdź_minucje() {
  czy_koniec := false;
  while( !czy_koniec ) {
     $(x_s, y_s) := \text{wyznacz\_nowy\_punkt\_początkowy};$ 
  }
}
```

```
 $\varphi_0 :=$  wyznacz kierunek w punkcie  $(x_s, y_s)$ ;  
śledź linię  $(x_s, y_s, \varphi_0)$ ;  
if ( zakończenie  $\vee$  nadmierne wyginanie linii ) then {  
    //odnaleziono zakończenie linii  
    zapamiętaj zakończenie linii;  
}  
if ( przecięcie ) then {  
    if ( przecięcie linii poprawne ) {  
        zapamiętaj przecięcie z inną linią;  
    } else {  
        skasuj błędne zakończenie linii;  
    }  
}  
//wykonaj te same operacje dla kierunku  $\varphi_0 + \pi$   
śledź linię  $(x_s, y_s, \varphi_0 + \pi)$ ;  
...  
}  
}
```

Pojawiają się tutaj kolejne zagadnienia istotne dla prawidłowego działania algorytmu. Oczywiście jest, że każdą linię papilarną należy przeanalizować dokładnie jeden raz. Oznacza to potrzebę stworzenia algorytmu odnajdującego kolejne, nie odwiedzone jeszcze, linie papilarne. Autorzy pracy [2] proponują, aby na obraz odcisku palca nałożyć dyskretną siatkę o ustalonym rozmiarze i wyznaczać kolejne punkty początkowe poprzez iteracyjne przechodzenie kolejnych węzłów siatki. W każdym kolejnym punkcie siatki tworzona jest sekcja. Wyznaczane jest jej minimum, które staje się punktem wyjściowym dla algorytmu śledzącego linię, chyba że tak wyznaczony punkt został już wcześniej zakwalifikowany jako należący do linii. Ta metoda gwarantuje odnalezienie wszystkich linii papilarnych znajdujących się na obrazie.

Gdy śledzenie linii zostanie zakończone należy przeanalizować przyczynę zatrzymania algorytmu. Jeśli obszar odcisku palca zostanie opuszczony, to nie odnaleziono żadnej minucji. W przypadku zakończenia śledzonej linii wykryta zostaje minucja kończąca – jest ona zapamiętywana. Zakończenie śledzenia linii z powodu przecięcia z już



Rysunek 2.17: Wykrywanie poszczególnych typów minucji [2]

odwiedzoną linią zachodzi wtedy, kiedy w wyniku poruszania się wzdłuż linii natrafimy na punkt oznaczony na dodatkowym obrazie jako już odwiedzony. Może do tego dojść w wyniku rozgałęzienia się dwóch linii – a więc istnienia w tym miejscu minucji – albo w wyniku natrafienia na linię, której śledzenie zostało wcześniej przerwane z powodu wykrycia błędnego zakończenia linii. W pierwszym przypadku należy zaznaczyć nową minucję. Drugi przypadek oznacza, że obie linie tworzą tak naprawdę jedną linię papilarną, a więc trzeba usunąć błędnie wykryty koniec linii. Obie sytuacje zostały pokazane na Rysunku 2.17. W przypadku zakończenia śledzenia linii z powodu jej nadmiernego wyginania również zaznaczana jest minucja kończąca. Może się zdarzyć, że nadmierne wyginanie linii zostaje wykryte w wyniku zgubienia przez algorytm śledzonej linii. Nie powinno to być jednak problemem, ponieważ fałszywa minucja zostanie wykryta w dalszych krokach algorytmu w wyniku śledzenia tej samej linii papilarnej z innego punktu początkowego.

2.3.3 Zastosowanie sieci neuronowych

W literaturze zaproponowano różne ulepszenia powyższej metody. Jednym z nich jest zastosowanie wielowarstwowego perceptronu do filtrowania minucji znalezionych za pomocą algorytmu zaprezentowanego powyżej [3]. Po wykryciu minucji jej najbliższe otoczenie podawane było na wejście sieci neuronowej wyuczonej w rozpoznawaniu zakończeń i rozgałęzień linii. Do uczenia sieci wykorzystano wspomniany już wcześniej dualizm pomiędzy minucjami linii i zagłębieniami. To podejście miało na celu odfiltrowanie błędnie wykrytych minucji.

2.4 Dopasowywanie odcisków palców

Zadaniem algorytmu dopasowującego odciski palców do siebie jest porównanie odcisku zapisanego w bazie danych jako szablon, z odciskiem uzyskanym z czytnika i określenie stopnia ich podobieństwa, jako liczbę pomiędzy 0 a 1 lub jako binarną decyzję – zaakceptowany/odrzucony. Większość metod dopasowywania odcisków palców operuje na reprezentacji odcisku w postaci wektora cech (np. minucji), a tylko nieliczne opierają się bezpośrednio na obrazie zeskanowanego odcisku palca.

Zagadnienie dopasowania do siebie dwóch odcisków jest problemem bardzo złożonym, ze względu na częste występowanie dużych różnic pomiędzy obrazami tych samych palców. Problemy mogą być wywołane następującymi czynnikami:

Przesunięcie : Odcisk palca może być umiejscowiony w innym miejscu na obrazie niż odcisk zapisany w bazie.

Obrót : Obraz może być obrócony względem szablonu w bazie.

Zeskanowanie fragmentu palca : Odcisk podczas skanowania może zostać przesunięty poza rejon czujnika, w wyniku czego fragment odcisku nie zostanie zeskanowany. W przypadku tanich czujników dostępnych na rynku ich rejon skanowania jest na tyle mały, że niemożliwe jest zeskanowanie całego odcisku, a jedynie jego fragmentu.

Zniekształcenie : Spowodowane jest przekształcaniem trójwymiarowego odcisku palca na dwuwymiarową reprezentację. Zniekształcenia powstają zarówno jako efekt zniekształcenia optycznego czujnika oraz jako rezultat plastyczności skóry. Zniekształcenia takie mają charakter nieliniowy. Wiele algorytmów zakłada, że odcisk palca nie jest zniekształcony.

Siła nacisku na czujnik i stan skóry : Powierzchnia palca mogłaby zostać w pełni poprawnie zeskanowana tylko wtedy, gdyby wszystkie krawędzie były w kontakcie z powierzchnią czujnika w jednakowy sposób. Jednak takie czynniki jak niejednorodna siła nacisku, wilgotność lub suchość skóry, choroby dermatologiczne czy zranienia powodują, że zeskanowane linie papilarne stają się niewyraźne. Zakłócenia takie sprawiają, że ten sam odcisk może się bardzo różnić podczas kolejnych skanowań.

Błędna ekstrakcja cech : Algorytmy ekstrakcji cech są niedoskonałe. Prowadzi to do błędnej ekstrakcji cech, w szczególności w przypadku obrazów niskiej jakości. Błędy mogą obejmować pomijanie istniejących minucji, wykrywanie minucji tam, gdzie ich nie ma oraz mylenie rodzajów minucji (wykrywanie zakończenia tam, gdzie mamy do czynienia z rozgałęzieniem i na odwrót).

2.4.1 Metody korelacyjne

Jednymi z metod dopasowywania odcisków palców są metody korelacyjne. Nie wymagają one ekstrakcji cech. Zamiast tego, w celu porównania dwóch odcisków palców, ich obrazy nakłada się na siebie w taki sposób, aby zmaksymalizować współczynnik korelacji $C(T,I)$, gdzie T oznacza obraz odcisku zapisany w bazie danych, a I oznacza weryfikowany odcisk palca:

$$C(T, I) = \sum_i \sum_j T(i, j) \cdot I(i, j) \quad .$$

Ponieważ obraz porównywany z wzorcem jest najczęściej obrócony o jakiś kąt θ oraz przesunięty o wektor $(\Delta x, \Delta y)$, należy wyznaczyć taką wartość tego kąta i wektora, która maksymalizuje wzajemną korelację obrazów:

$$S(T, I) = \max_{\Delta x, \Delta y, \theta} C(T, I^{(\Delta x, \Delta y, \theta)}) \quad . \quad (2.1)$$

W praktyce nie jest możliwe zastosowanie metody opartej bezpośrednio o równanie 2.1, ponieważ:

1. Jest ona bardzo kosztowna obliczeniowo.
2. Odcisk palca jest często nieznacznie zniekształcony. Zniekształcenia powstają w procesie skanowania linii papilarnych, ponieważ znajdujące się na palcu trójwymiarowe odciski są przekształcane na obraz dwuwymiarowy. Zniekształcenia te mają charakter globalny, a więc nie powodują w żaden sposób zakłóceń lokalnej struktury linii. Powodują jednak, że cały obraz nie może być wiarygodnie dopasowany za pomocą funkcji korelacji.
3. Siła nacisku na czujnik, stan skóry czy stopień jej wilgotności powodują istotne różnice w kontraście, jasności i szerokości linii papilarnych na zeskanowanym obrazie.

Większość rozwiązań pierwszego problemu opiera się o transformatę Fouriera (lub podobne) i obliczanie korelacji w dziedzinie częstotliwości. Problem drugi jest najczęściej rozwiązywany poprzez podział obrazu na mniejsze rejony i ich lokalne dopasowywanie do szablonu. Często do dopasowywania wybiera się tylko niektóre fragmenty obrazu, np. rejony wokół minucji (wtedy wymagana jest ekstrakcja cech) albo obszary, w których linie są mocno powyginane. Pozwala to także na rozwiązanie pierwszego problemu, ponieważ dopasowywanie tylko kilku fragmentów znacząco redukuje złożoność obliczeniową. Trzeci problem może być rozwiązany za pomocą odpowiedniej filtracji równoważącej kontrast i jasność na obu obrazach.

Zaproponowano również systemy dokonujące korelacji metodami optycznymi za pomocą systemu soczewek. Takie systemy jednak są drogie ze względu na wysoką złożoność i nie zapewniają dostatecznie wysokiej poprawności rozpoznawania.

2.4.2 Metody oparte o minucje

Metody te zakładają, że każdy odcisk opisany jest za pomocą wektora cech. Poszczególne elementy tego wektora – minucje – są opisane za pomocą współrzędnych położenia, rodzaju i kąta pod jakim są skierowane¹⁶. Zagadnienie dopasowania wzorca sprowadza się więc do określenia czy dwa wektory cech opisują ten sam odcisk palca, a więc czy minucje znajdują się w tych samych miejscach na obu odciskach. Oczywiście należy pamiętać, że cechy są punktami rozlokowanymi w przestrzeni dwuwymiarowej. W daktyloskopii powszechnie przyjmuje się, że dwa odciski są identyczne, jeżeli pokrywa się na nich 12 minucji¹⁷. Problemy z dopasowaniem rodzą się z powodu czynników wymienionych w Rozdziale 2.4.

Formalnie problem dopasowania minucji można zdefiniować następująco. Przez T oznaczmy wektor cech zapisany w bazie danych, natomiast przez I oznaczmy wektor cech weryfikowanego odcisku, który będziemy próbowali dopasować do wzorca:

$$T = \{m_1, m_2, \dots, m_m\}, m_i = \{x_i, y_i, \theta_i\}, i = 1 \dots m \quad ,$$
$$I = \{m'_1, m'_2, \dots, m'_n\}, m'_j = \{x'_j, y'_j, \theta'_j\}, j = 1 \dots n \quad ,$$

¹⁶Wiele metod ignoruje rodzaj minucji, ponieważ na różnych obrazach, w zależności od jakości, ta sama minucja może wyglądać raz jak rozgałęzienie, a raz jak zakończenie.

¹⁷Wynika to z rachunku prawdopodobieństwa. Obecnie kryminaliści w niektórych krajach odchodzą od tej zasady.

gdzie m oznacza ilość minucji we wzorcu, a n ilość minucji w weryfikowanym odcisku. Minucję m'_j należącą do I uznaje się za pasującą do minucji m_i należącej do T, jeżeli ich odległość jest mniejsza od ustalonego progu r_0 oraz różnica ich kierunków jest mniejsza od ustalonego kąta tolerancji θ_0 :

$$sd(m'_j, m_i) = \sqrt{(x'_j - x_i)^2 + (y'_j - y_i)^2} \leq r_0 \quad , \quad (2.2)$$

$$dd(m'_j, m_i) = \min(|\theta'_j - \theta_i|, 360^\circ - |\theta'_j - \theta_i|) \leq \theta_0 \quad . \quad (2.3)$$

Wprowadzenie pewnych zakresów tolerancji jest konieczne do zrównoważenia błędów w procesie ekstrakcji cech oraz zniekształceń odcisku palca powstających w trakcie skanowania.

W celu dopasowania do siebie T oraz I należy najpierw wyrównać odciski względem siebie, a więc znaleźć ich wzajemne przesunięcie oraz obrót¹⁸.

Niech $map(\cdot)$ oznacza funkcję, która realizuje mapowanie minucji m'_j z I na minucję m''_j . Mapowanie odpowiada przekształceniu geometrycznemu realizującemu obrót o kąt θ oraz przesunięcie o wektor $[\Delta x, \Delta y]$:

$$map_{\Delta x, \Delta y, \theta}(m'_j = \{x'_j, y'_j, \theta'_j\}) = m''_j = \{x''_j, y''_j, \theta'_j + \theta\} \quad ,$$

gdzie:

$$\begin{bmatrix} x''_j \\ y''_j \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} x'_j \\ y'_j \end{bmatrix} + \begin{bmatrix} \Delta x_j \\ \Delta y_j \end{bmatrix} \quad . \quad (2.4)$$

Niech $mm(\cdot)$ będzie funkcją, która zwraca 1, jeśli minucje m''_j i m_i pasują do siebie w sensie równań 2.2 i 2.3:

$$mm(m''_j, m_i) = \begin{cases} 1 & \text{jeżeli } sd(m'_j, m_i) \leq r_0 \text{ i } dd(m'_j, m_i) \leq \theta_0 \\ 0 & \text{w przeciwnym wypadku} \end{cases} \quad . \quad (2.5)$$

Ostatecznie problem dopasowania wzorców może zostać zapisany jako zadanie optymalizacyjne, którego celem jest maksymalizacja ilości dopasowanych minucji:

$$\max_{\Delta x, \Delta y, \theta, P} \sum_{i=1}^m mm(map_{\Delta x, \Delta y, \theta}(m'_{P(i)}), m_i) \quad , \quad (2.6)$$

¹⁸W przypadku stosowania do weryfikacji czytników o innej rozdzielczości niż podczas pobierania odcisku należy również uwzględnić skalę. W tej pracy założono, że skalowanie nie jest potrzebne.

gdzie $P(i)$ jest nieznaną funkcją określającą dopasowanie między poszczególnymi minucjami, taką że:

1. $P(i) = j$ oznacza, że minucja m_i z T została przypisana do minucji m'_j z I¹⁹
2. $P(i) = \text{null}$ oznacza, że minucja m_i z T nie ma odpowiednika w I
3. minucja m'_j z I, taka że $\forall_{i=1\dots m} P(i) \neq j$ nie ma odpowiednika w T
4. $\forall_{i=1\dots m, k=1\dots m} i \neq k \Rightarrow P(i) \neq P(k) \vee P(i) = P(k) = \text{null}$ (oznacza to, że każda minucja z I jest przypisana do co najwyżej jednej minucji z T)

Jeśli dwie minucje pasują do siebie w sensie równań 2.2 i 2.3, to ich wkład do wzoru 2.6 jest taki sam, bez względu na jakość dopasowania. Można zaprojektować alternatywne wersje równania 2.5, w których wartości zwracane przez funkcję $mm(\cdot)$ nie będą binarne, a będą ilościowo zależę od jakości danego dopasowania.

Znalezienie funkcji $P(i)$ jest zadaniem bardzo prostym, jeśli znane jest wzajemne przesunięcie $[\Delta x, \Delta y]$ i obrót θ szablonu T i I. Wtedy dopasowanie $P(i)$ może zostać wyznaczone poprzez wykonanie dla każdej minucji m_i należącej do T ($i = 1, \dots, m$) następującego algorytmu:

- $P(i) = j$, jeśli $m''_j = \text{map}_{\Delta x, \Delta y, \theta}(m'_j)$ jest minucją najbliższą minucji m_i spośród minucji $\{ m''_k = \text{map}_{\Delta x, \Delta y, \theta}(m'_k), k = 1 \dots n : mm(m''_k, m_i) = 1 \}$.
- $P(i) = \text{null}$, jeśli $\forall_{k=1\dots n} mm(\text{map}_{\Delta x, \Delta y, \theta}(m'_k), m_i) = 0$.

Należy pamiętać, że każda minucja m''_j z I, która zostanie przypisana do minucji z T musi zostać oznaczona, by zapewnić spełnienie warunku numer 4 opisanego powyżej. Zaprezentowany algorytm ma charakter zachłanny i nie zawsze musi prowadzić do wyznaczenia funkcji $P(i)$ maksymalizującej równanie 2.6.

Maksymalizacja równania 2.6 może zostać łatwo zrealizowana, jeśli funkcja $P(i)$ jest znana. W praktyce jednak nie są znane ani parametry $(\Delta x, \Delta y, \theta)$ przekształcenia geometrycznego ani funkcja $P(i)$, co sprawia, że zadanie dopasowania wzorców staje się bardzo skomplikowane. Metody brute-force, testujące wszystkie możliwe rozwiązania, nie dają się efektywnie zastosować, ze względu na dużą liczbę możliwych kombinacji dopasowania minucji. W literaturze zaproponowano liczne sposoby praktycznego

¹⁹Nie jest to równoznaczne ze spełnieniem przez przypisane minucje równań 2.2 i 2.3.

rozwiązania tego zagadnienia. Obejmują one m.in. zastosowanie algorytmów genetycznych, metod opartych o algorytmy drzewowe i sieci przepływowe czy też metod opartych o funkcję energii. Metody te nie są jednak wystarczająco skuteczne. Wymagają albo bardzo długich obliczeń (jak metody oparte o energię czy algorytmy genetyczne, które w dodatku nie gwarantują znalezienia optymalnego rozwiązania), albo też spełnienia nierealnych założeń. Przykładowo, metody drzewowe wymagają istnienia takiej samej liczby minucji na dopasowywanych obrazach, co w praktyce wymagałoby skanowania dokładnie tego samego obszaru palca za każdym razem i jednocześnie całkowitej poprawności w procesie ekstrakcji cech. Dlatego w tej pracy zdecydowano się na zastosowanie metody opartej o transformatę Hough'a, która jest metodą szybką i odporną na niedokładności w procesie ekstrakcji cech.

Wyznaczanie parametrów transformacji geometrycznej przy pomocy transformaty Hough'a

Idea algorytmu opartego o transformatę Hough'a opiera się o proces głosowania, w wyniku którego wyznaczane jest przekształcenie pozwalające dopasować do siebie największą liczbę minucji.

Przestrzeń, w której dokonywana jest transformacja, składa się z uporządkowanych trójek $(\Delta x, \Delta y, \theta)$ odpowiadających parametrom przekształcenia. Przestrzeń rozważań musi zostać zdyskretyzowana (wartości dyskretne oznaczono symbolem $^+$) tak, aby poszczególne parametry miały skończony zestaw wartości:

$$\Delta x^+ \in \{\Delta x_1^+, \Delta x_2^+, \dots, \Delta x_a^+\} \quad , \quad \Delta y^+ \in \{\Delta y_1^+, \Delta y_2^+, \dots, \Delta y_b^+\} \quad ,$$

$$\theta^+ \in \{\theta_1^+, \theta_2^+, \dots, \theta_c^+\} \quad .$$

Tworzona jest tablica A (tzw. akumulator) o ilości wymiarów równej ilości parametrów opisujących przestrzeń rozważań (w tym wypadku trzy). Tablica posiada jedną komórkę dla każdej możliwej kombinacji dyskretnych wartości parametrów. Akumulator jest wykorzystywany w procesie wyznaczania optymalnego przekształcenia do zapamiętywania wyników głosowania:

dla każdej minucji $m_i \in T$, $i = 1, \dots, m$

dla każdej minucji $m'_j \in I$, $j = 1, \dots, n$

dla każdego kąta $\theta^+ \in \{\theta_1^+, \theta_2^+, \dots, \theta_c^+\}$

jeśli $dd(\theta_j^+ + \theta^+, \theta_i) < \theta_0$ { (1)

$$\begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} x_i \\ y_i \end{bmatrix} - \begin{bmatrix} \cos \theta^+ & -\sin \theta^+ \\ \sin \theta^+ & \cos \theta^+ \end{bmatrix} \begin{bmatrix} x'_j \\ y'_j \end{bmatrix} \quad (2)$$

$\Delta x^+, \Delta y^+ =$ kwantyzacja $\Delta x, \Delta y$ do najbliższych wartości dyskretnych

$$A[\Delta x^+, \Delta y^+, \theta^+] ++; \quad (3)$$

}

Zasada działania algorytmu jest następująca. Próbujemy dopasować każdą minucję z T do każdej minucji z I. Rozważamy wszystkie możliwe wartości kąta obrotu pomiędzy T a I. Jeżeli okaże się, że po wykonaniu obrotu o kąt θ^+ minucje pasują do siebie w sensie równania 2.3 (linijka oznaczona 1), wtedy obliczana jest potencjalna wartość przesunięcia w oparciu o równanie 2.4 (linijka 2). Wartość przesunięcia jest dyskretyzowana, aby za jej pomocą można było zaadresować komórkę tablicy A. Wartość zapisana w tej komórce jest zwiększana o jeden (linijka 3), co oznacza "oddanie głosu" na obliczone wartości $(\Delta x^+, \Delta y^+, \theta^+)$. Po zakończeniu etapu głosowania najlepsze parametry $(\Delta x^*, \Delta y^*, \theta^*)$ są wyznaczone jako adres komórki akumulatora posiadającej największą wartość (a więc tej, na którą oddano najwięcej głosów):

$$(\Delta x^*, \Delta y^*, \theta^*) = \arg \max_{\Delta x^+, \Delta y^+, \theta^+} A[\Delta x^+, \Delta y^+, \theta^+] \quad .$$

Znając parametry przekształcenia geometrycznego można przeprowadzić dopasowywanie minucji w celu określenia stopnia podobieństwa odcisków palców. Algorytm dopasowywania został zaproponowany w Rozdziale 3.3.2.

2.4.3 Metody oparte o krawędzie

Ekstrakcja minucji w obrazach niskiej jakości jest zagadnieniem trudnym. Jako rozwiązanie tego problemu powstały metody dopasowywania oparte m.in. o lokalne kierunki krawędzi linii papilarnych i częstość występowania krawędzi. Są one zdolne do operowania na obrazach gorszej jakości, dla których poprawne rozpoznanie minucji może nie być możliwe. Z drugiej jednak strony struktura linii papilarnych może być mniej unikalna niż struktura minucji. Metody te nie są powszechnie wykorzystywane z uwagi na coraz lepszą jakość obrazu dostarczanego przez czytniki.

2.5 Podsumowanie

W celu zwiększenia efektywności działania systemów, opartych o odciski palców, wymagane jest udoskonalenie zarówno sprzętu pobierającego odciski, jak również metod i algorytmów rozpoznawania linii papilarnych. Wśród zaprezentowanych metod ekstrakcji cech, najbardziej obiecująca wydaje się metoda śledzenia linii papilarnych. Działa ona bez konieczności przetwarzania wstępnego takiego jak binaryzacja i operacje morfologiczne, co znacząco redukuje ilość wymaganych obliczeń jak również zmniejsza prawdopodobieństwo błędnej ekstrakcji cech. Metoda może również operować na obrazach o niskim kontraście, co pozwala na skompensowanie niedoskonałości sprzętu. Spośród istniejących metod dopasowywania do siebie dwóch odcisków najbardziej perspektywiczne wydają się być metody oparte o dopasowywanie do siebie minucji. Są znacznie szybsze oraz dokładniejsze od metod alternatywnych. Wśród metod opartych o minucje najbardziej obiecujące zdają się być algorytmy oparte o transformatę Hough'a. Wynika to z szybkiego działania tych algorytmów oraz najwyższej odporności na błędy w porównaniu do innych metod.

Rozdział 3

Ulepszone algorytmy rozpoznawania odcisków palców

3.1 Przetwarzanie wstępne

Istotną cechą wybranego algorytmu ekstrakcji cech jest to, że teoretycznie powinien on działać bez konieczności stosowania dodatkowego przetwarzania wstępnego, jak na przykład binaryzacja. Można jednak zastosować wstępne polepszenie jakości obrazu w celu uzyskania wyższej skuteczności w procesie ekstrakcji cech. W niniejszej pracy zaproponowano wykorzystanie dwóch technik dodatkowego przetwarzania wstępnego.

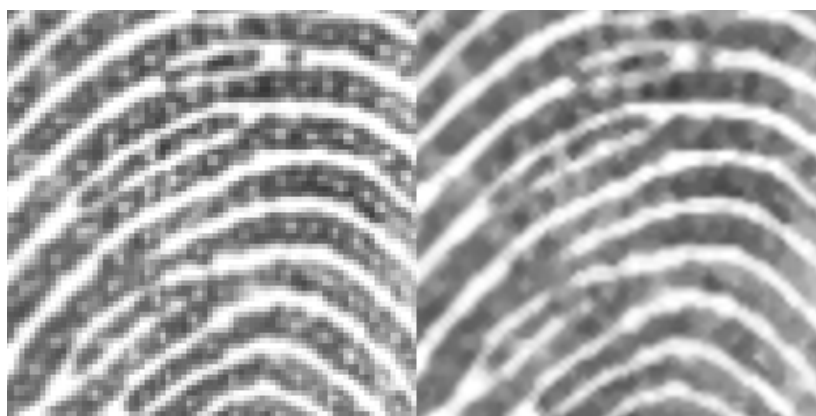


Rysunek 3.1: Obraz odcisku przed i po przycięciu

Pierwsza technika polega na zmniejszeniu rozmiarów przetwarzanego obrazu poprzez jego przycięcie do rejonu zajmowanego przez odcisk palca. Idea ta została zaproponowana na podstawie analizy wyglądu obrazów należących do zastosowanej bazy danych. Na przetwarzanych obrazach obszar zajmowany przez odcisk palca stanowi z reguły niewielką część całego obrazu (przykład pokazany jest na Rysunku 3.1). Uznano, że dzięki zmniejszeniu rozmiarów przetwarzanego obrazu, będzie można oczekiwać redukcji ilości pamięci potrzebnej na jego przechowywanie, a także skrócenia procesu segmentacji odcisku z tła. Przycięcie nie powinno mieć wpływu na samą skuteczność ekstrakcji cech.

Druga z zaproponowanych technik przetwarzania wstępnego to filtracja za pomocą filtru medianowego. Na obrazach odcisków palców dosyć dokładnie widoczne są pory potne w skórze. Ich obecność może zakłócać proces śledzenia linii papilarnej. Jedną z zastosowanych technik radzenia sobie z tym problemem jest filtracja każdej sekcji linii papilarnej. Zmiana jasności poszczególnych pikseli w sekcji odbywa się na podstawie jasności pikseli sąsiednich – stosuje się średnią ważoną, gdzie piksele położone bliżej przetwarzanego piksela uzyskują większą wagę.

Zaobserwowano jednak, że zastosowanie filtru medianowego usuwa pory potne z obrazu odcisku palca (Rysunek 3.2). Sugeruje się więc zastosowanie filtru medianowego w celu usunięcia zakłóceń w postaci widocznych porów w skórze, a tym samym zapewnienia dokładniejszej ekstrakcji cech.



Rysunek 3.2: Po lewej fragment odcisku przed filtracją medianową. Po prawej ten sam fragment po przefiltrowaniu. Widać usunięcie porów skórnych znajdujących się na liniach papilarnych

3.2 Ekstrakcja cech

3.2.1 Segmentacja odcisku z tła

Zaproponowany w tej pracy algorytm segmentacji odcisku z tła został ściśle dostosowany do wybranej bazy danych. Obraz uzyskiwany z wybranego czytnika charakteryzuje się tym, że jego tło jest jednolicie białe, a jedynym rejonem gdzie piksele obrazu mają inny kolor jest sam rejon odcisku palca (idea przycinania obrazu omówiona we wcześniejszym podrozdziale oparta została o dokładnie to samo spostrzeżenie).

Segmentacja oparta została o określony w pliku konfiguracyjnym kolor tła oraz zakres tolerancji dla tego koloru. Każdy wiersz obrazu przechodzony jest najpierw od lewej, a potem od prawej krawędzi. Piksel, w którym przekroczony zostanie po raz pierwszy podany przez użytkownika zakres tolerancji dla jasności tła, zostaje uznany za początek, odpowiednio lewej albo prawej, krawędzi odcisku w danym wierszu. Algorytm zapisany w pseudokodzie wygląda następująco:

```
segmentuj_odcisk( piksele_obrazu ) {
  dla każdego rzędu {
    //najpierw znajdowana jest lewa krawędź odcisku
    dla każdej kolumny (przechodząc od lewej strony) {
      jeśli ( piksele_obrazu[ rząd ][ kolumna ] nie należy do tła ) {
        lewa_krawędź[ rząd ] = kolumna;
        przerwij wewnętrzną pętlę (przejdź do szukania prawej krawędzi);
      }
    }
    //teraz wyznaczana jest prawa krawędź odcisku
    dla każdej kolumny (przechodząc od prawej strony) {
      jeśli ( piksele_obrazu[ rząd ][ kolumna ] nie należy do tła ) {
        prawa_krawędź[ rząd ] = kolumna;
        przerwij wewnętrzną pętlę (przejdź do następnego rzędu);
      }
    }
  }
}
```

Jako wynik działania tego algorytmu otrzymuje się dwie tablice. Jedna reprezentuje ciąg numerów kolumn wyznaczających lewy brzeg odcisku palca, druga reprezentuje prawy brzeg. Na podstawie znajomości współrzędnych lewej i prawej krawędzi można jednoznacznie wyznaczyć obszar odcisku palca:

```
piksel_jest_w_rejonie_odcisku( rząd, kolumna ) {
    return lewa_krawędź[ rząd ] <= kolumna ∧ prawa_krawędź[ rząd ] >= kolumna;
}
```

Zaobserwowano, że obszar odcisku wyznaczony tą metodą ma mocno poszarpane krawędzie. Mogłoby to prowadzić do błędnego wykrywania opuszczenia rejonu odcisku palca w procesie śledzenia linii papilarnych. Wpłynęło by to negatywnie na poprawność procesu ekstrakcji cech. Należy więc skorygować wyznaczone krawędzie. W tym celu każdy z dwóch wyznaczonych ciągów współrzędnych poddawany jest filtracji medianowej. Ilość sąsiednich wartości uwzględnianych w procesie filtracji może zostać zmieniona w pliku konfiguracyjnym. Rysunek 3.3 przedstawia odcisk wysegmentowany z tła przed i po wygładzeniu krawędzi.



Rysunek 3.3: Segmentacja odcisku z tła. Po lewej widoczny odcisk przed wygładzeniem krawędzi. Po prawej krawędzie po wygładzeniu

3.2.2 Wyznaczanie kierunku linii

Wyznaczanie mapy kierunków dla odcisku palca dokonywane jest przed rozpoczęciem właściwej ekstrakcji cech. Ponieważ jest to proces czasochłonny stosuje się metodę mającą na celu przyspieszenie obliczeń. Kierunki linii papilarnych wyznacza się jedynie w punktach pewnej dyskretnej siatki. W trakcie śledzenia linii papilarnej, w celu uzyskania kierunku dla punktu leżącego pomiędzy węzłami tej siatki, stosuje się interpolację liniową pomiędzy czterema najbliższymi węzłami.

W niniejszej pracy proponuje się, aby wykorzystać interpolację metodą najbliższego sąsiada zamiast interpolacji liniowej. Spowodowane jest to tym, że linie papilarne nie zmieniają swojego kierunku w sposób gwałtowny. Oznacza to, że wartości kierunków w czterech sąsiadujących ze sobą węzłach są najczęściej bardzo zbliżone. Przybliżenie za pomocą najbliższego sąsiada powinno być więc wystarczająco dokładne. Interpolacja tą metodą nie wymaga – w przeciwieństwie do interpolacji liniowej – dodatkowych obliczeń, a jedynie odczytu z odpowiedniej komórki tablicy. Jeśli więc ta technika pozwoliłaby na uzyskanie równie dobrych wyników co interpolacja liniowa, należałoby korzystać z niej do wyznaczania kierunku jako mniej kosztownej obliczeniowo.

3.2.3 Wyznaczanie nowych punktów początkowych

Wyznaczanie kolejnych punktów początkowych dla algorytmu śledzącego linie oparte jest o dyskretną siatkę nałożoną na obraz. Kolejne węzły tej siatki są przechodzone iteracyjnie. W każdym węźle wyznaczana jest sekcja linii papilarnej. Jej minimum staje się nowym punktem początkowym dla algorytmu śledzącego linie, chyba że tak wyznaczony punkt był już wcześniej oznaczony jako należący do linii papilarnej.

Zaobserwowano jednak pewne nieprawidłowości w przypadku gdy linie papilarne w którymś fragmencie obrazu były zbyt oddalone od siebie. W takiej sytuacji zdarzało się, że punkt początkowy wyznaczany był dokładnie pomiędzy liniami, co w rezultacie prowadziło do wykrycia w tym miejscu fałszywego zakończenia linii. Błąd ten nie zawsze był możliwy do skorygowania poprzez zwiększenie odległości węzłów siatki nałożonej na obraz.

W niniejszej pracy proponuje się, by przy wyborze nowego punktu początkowego uwzględniać także jego jasność. Jeśli nowo wyznaczony punkt na podstawie poziomu jasności nie może zostać zakwalifikowany jako należący do linii papilarnej, wtedy jest

odrzucający i algorytm wybierający kolejne punkty początkowe przechodzi do następnego węzła siatki. Tak sformułowany algorytm można zapisać w następujący sposób:

```
wyznacz_nowy_punkt_początkowy( ) {
  do {
    while( aktualny węzeł należy do tła  $\vee$  aktualny węzeł był odwiedzony ) {
      przejdź do następnego węzła siatki;
    }
    punkt_początkowy := aktualny węzeł siatki;
    wyznacz kierunek linii w nowym punkcie początkowym;
    utwórz sekcję w oparciu o wyznaczony punkt początkowy i kierunek linii;
    punkt_początkowy := minimum utworzonej sekcji;
  } while ( punkt_początkowy należy do tła  $\vee$  punkt_początkowy był odwiedzony );
  return punkt_początkowy;
}
```

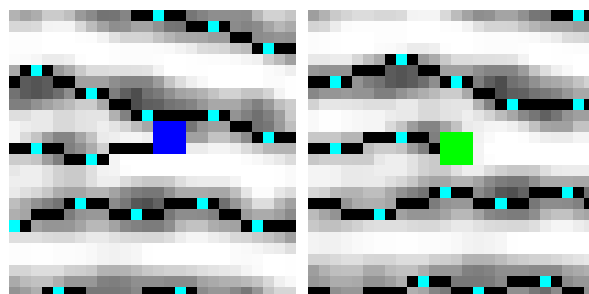
Proponuje się również, by przetestować skuteczność działania techniki, polegającej na traktowaniu jako punkt początkowy po kolei wszystkich pikseli obrazu (odległość węzłów siatki równa jeden).

3.2.4 Warunki stopu

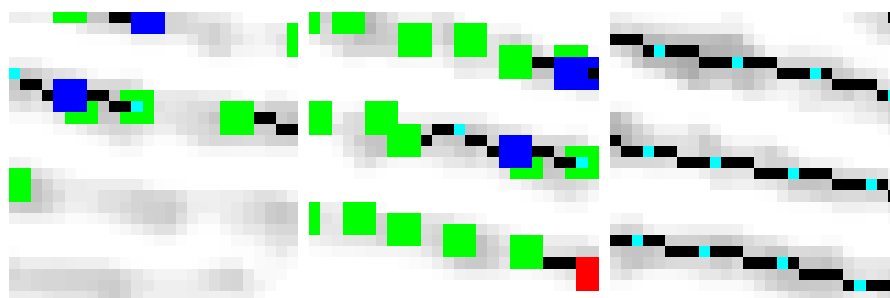
Algorytm śledzący linię oparty jest o cztery kryteria stopu:

- Wyjście linii papilarnej poza obszar odcisku palca wyznaczony w procesie segmentacji.
- Odcinek łączący minimum nowej sekcji z minimum aktualnej sekcji tworzy zbyt duży kąt w stosunku do kierunku linii papilarnych w aktualnym minimum. Oznacza to, że linia się kończy, a więc wykryto minucję.
- Punkt, będący minimum nowej sekcji, był już wcześniej odwiedzony. Oznacza to, że dwie linie papilarne się łączą, a więc wykryto minucję.
- Linia papilarna nadmiernie się wygina. Warunek sprawdzany jest w oparciu o zapamiętane kierunki poruszania się w kilku ostatnich krokach. Spełnienie tego

warunku jest również traktowane jako odnalezienie minucji kończącej linię.



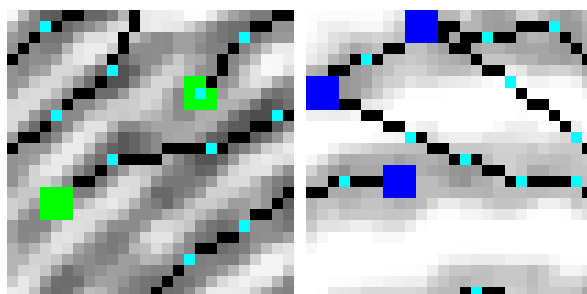
Rysunek 3.4: Przykład błędnego wykrywania zakończenia linii



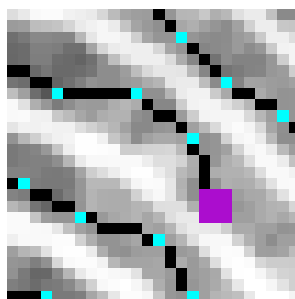
Rysunek 3.5: Wpływ uwzględniania poziomów jasności pikseli na proces śledzenia linii

W trakcie analizowania działania algorytmu, przy tak sformułowanych kryteriach stopu, okazało się, że często dochodzi do sytuacji, w której zakończenie linii nie zostaje wykryte, co prowadzi do przeskoczenia na inną linię lub – w sytuacji gdy linia była już odwiedzona – do wykrycia fałszywego rozgałęzienia linii. Okazało się również, że w trakcie śledzenia jednej linii może dochodzić do dużych zmian kierunku pomiędzy kolejnymi sekcjami, jednak nie jest to zjawisko wywołane błędem.

W niniejszej pracy proponuje się zrezygnowanie z analizowania stopnia wyginania linii papilarniej. Ponadto proponuje się zastąpienie warunku wykrywania zakończenia linii innym. Nowe kryterium oparte jest na badaniu, w oparciu o poziom jasności pikseli, przeskakiwania algorytmu śledzenia linii pomiędzy sąsiednimi liniami papilarnymi. Po wyznaczeniu minimum kolejnej sekcji analizuje się jasność pikseli na linii łączącej stare i nowe minimum. Jeśli na podstawie jasności okaże się, że którykolwiek piksel leżący na tym odcinku należy do tła, wtedy algorytm śledzenia linii zatrzymuje się, uznając,



Rysunek 3.6: Przeskakiwanie pomiędzy sąsiednimi liniami papilarnymi przy ignorowaniu poziomów jasności pikseli



Rysunek 3.7: Błędnie wykryte nadmierne wyginanie linii

że nastąpił przeskok między sąsiednimi liniami. Jest to równoznaczne ze znalezieniem minucji kończącej linię.

Skuteczność działania pierwotnych oraz nowych kryteriów stopu pokazana jest na rysunkach 3.4, 3.5, 3.6 i 3.7. Widać z nich, że uwzględnianie bądź ignorowanie poziomów jasności pikseli oraz analizowanie nadmiernego wyginania w trakcie śledzenia przebiegu linii może prowadzić do bardzo różnych rezultatów.

Na Rysunku 3.4 po lewej stosowany jest zwykły algorytm wykrywania zakończenia linii. Wykryte zostaje nieprawidłowe rozgałęzienie linii (kolor niebieski). Na rysunku po prawej zastosowano nową metodę wykrywania zakończeń opartą o poziom jasności. Zakończenie linii – kolor zielony – zostaje wykryte prawidłowo.

Rysunku 3.5 pokazuje wpływ uwzględniania poziomów jasności na działanie algorytmu. Pierwszy rysunek po lewej pokazuje co dzieje się gdy jasność pikseli zostanie uwzględniona zarówno w procesie śledzenia linii jak i wyznaczania nowych punktów początkowych. Na środkowym rysunku uwzględniono kolor pikseli tylko przy wykrywaniu zakończeń linii. Na rysunku po prawej poziomy jasności nie były uwzględniane.

Rysunek po prawej pokazuje, że nieuwzględnianie poziomów jasności może pozwalać na poprawne prześledzenie linii w rejonach o niskim kontraście.

Rysunek 3.6 pokazuje jednak, że ignorowanie poziomu jasności prowadzi do błędów w procesie śledzenia linii. Algorytm przeskakuje pomiędzy sąsiednimi liniami papilarnymi. Wykrycie błędu w procesie śledzenia poprzez analizę stopnia wyginania linii byłoby, w zaprezentowanych sytuacjach, niemożliwe.

Na Rysunku 3.7 zaprezentowano sytuację, w której dochodzi do nieprawidłowego stwierdzenia nadmiernego wyginania linii (kolor fioletowy) mimo, że w dalszym ciągu śledzona jest ta sama linia. W zaledwie dwóch krokach dochodzi do zmiany kierunku linii o prawie 90°.

Ostatecznie sprawdzanie kryteriów stopu dla algorytmu śledzącego linię można zapisać następująco:

```
sprawdź_kryteria_stopu( ) {  
    kryterium_stopu = niespełnione;  
    if ( nowe_minimum_leży_poza_rejonem_wysegmentowanym_z_tła ) {  
        kryterium_stopu = opuszczenie_obszaru_odcisku;  
    } else if ( przeskoczono_pomiędzy_liniami ) {  
        kryterium_stopu = zakończenie_linii;  
    } else if ( nowe_minimum_jest_oznaczone_jako_już_odwiedzzone ) {  
        kryterium_stopu = przecięcie_linii;  
    }  
    return kryterium_stopu;  
}
```

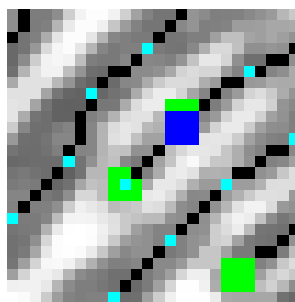
3.2.5 Wykrywanie fałszywych minucji

Do znalezienia minucji dochodzi w kilku przypadkach. Wykrycie minucji kończącej ma miejsce w przypadku stwierdzenia zakończenia linii lub jej nadmiernego wyginania. Wykrycie minucji oznaczającej rozgałęzienie zachodzi, gdy, w wyniku poruszania się wzdłuż linii, algorytm natrafi na punkt wcześniej odwiedzony.

Część minucji wykrytych w ten sposób może okazać się fałszywa. Przykładowo, do wykrycia minucji może dojść w wyniku rozgałęzienia się dwóch linii – a więc ist-

nienia minucji w tym miejscu – albo w wyniku natrafienia na linię, której śledzenie zostało wcześniej przerwane z powodu wykrycia błędnego zakończenia linii. W drugim przypadku mamy do czynienia aż z dwoma fałszywymi minucjami - błędnym zakończeniem wykrytym w wyniku stwierdzenia nadmiernego wyginania linii oraz błędnym przecięciem stwierdzonym w wyniku natrafienia na linię, której śledzenie zostało wcześniej błędnie przerwane. Konieczna jest więc metoda odróżniania fałszywych minucji od prawdziwych, aby możliwe było usunięcie niepoprawnie wykrytych minucji.

W niniejszej pracy zaproponowano dwie metody wykrywania takich sytuacji. Pierwsze kryterium oparte jest o analizę wzajemnej odległości minucji. Wynika ono z tego, że na rzeczywistym odcisku palca dwie minucje nie mogą się na siebie nakładać. Jeśli więc dochodzi do sytuacji, w której dwie minucje znajdują się w odległości mniejszej od pewnego zdefiniowanego progu, zostają one uznane za fałszywe i usunięte z listy wykrytych minucji. Na Rysunku 3.8 widoczne są dwa poprawnie wykryte zakończenia linii oraz dwie błędnie wykryte minucje. Fałszywe zakończenie wykryte zostało w wyniku zgubienia śledzonej linii. Fałszywe rozgałęzienie znaleziono w wyniku natrafienia na linię, której śledzenie zostało przerwane w wyniku błędu. Dwie błędnie wykryte minucje zostaną usunięte dzięki zastosowaniu powyższego kryterium.



Rysunek 3.8: Przykład wykrycia fałszywych minucji, które zostaną usunięte

Drugie kryterium usuwania błędnych minucji oparte jest o analizę odległości minucji od krawędzi obszaru odcisku palca wysegmentowanego z tła. Jeśli minucja znajduje się zbyt blisko brzegu odcisku, zostaje usunięta. Uzasadnieniem takiego posunięcia jest fakt, że za każdym razem skanowany jest nieco inny obszar odcisku palca, co wynika z tego, że użytkownik nie zawsze przykładá palec do czytnika w ten sam sposób. Ponadto, na obrzeżach odcisku jakość linii często jest słabsza niż w centrum, co wynika

ze słabszej siły nacisku na czujnik. W wyniku zmniejszenia się kontrastu pomiędzy liniami papilarnymi a tłem, dochodzi do błędów w procesie śledzenia linii, a tym samym do wykrywania błędnych minucji, w szczególności zakończeń linii. Przykład takiej sytuacji pokazany został na Rysunku 3.9. Na lewym rysunku znajduje się wykryte rozgałęzienie linii papilarnych. Na prawym pokazano wykrycie licznych zakończeń. W obu przypadkach doszło do błędnego wykrycia minucji na obrzeżu odcisku palca. Widać więc, że minucje znajdujące się na obrzeżu odcisku mają mniejsze znaczenie od minucji znajdujących się w centralnym rejonie odcisku palca, który jest najczęściej skanowany w dobrej jakości.



Rysunek 3.9: Błędnie wykryte minucje na obrzeżu odcisku palca

3.3 Dopasowywanie wzorców

3.3.1 Wyznaczanie przekształcenia geometrycznego

W celu porównania ze sobą dwóch odcisków palców należy wyznaczyć ich wzajemne przesunięcie i obrót. Przyjmuje się więc założenie, że odciski te są identyczne, a następnie próbuje odnaleźć takie przekształcenie geometryczne, które prowadzi do ich najlepszego dopasowania. Wykorzystuje się w tym celu metodę opartą o transformatę Hough'a. W tej metodzie zakłada się, że każda minucja z pierwszego odcisku może być potencjalnie dopasowana do każdej minucji z drugiego odcisku. Dla każdej pary minucji oblicza się parametry przekształcenia geometrycznego – przesunięcie i obrót – jakiego trzeba by dokonać aby te dwie minucje się pokryły. Wyznaczone parametry są dyskretyzowane, a następnie zapamiętywane – mówi się, że na wyznaczone wartości parametrów oddany został jeden głos. Po przeprowadzeniu głosowania, czyli po oddaniu

głosów przez wszystkie możliwe kombinacje minucji, jako optymalne przekształcenie wybierane jest to, na które oddano najwięcej głosów.

W standardowej wersji algorytmu w każdej iteracji oddawany jest tylko jeden głos. Takie podejście może jednak prowadzić do niestabilności rozwiązania. Spowodowane jest to dyskretyzacją parametrów przekształcenia, w wyniku której jedna zdyskretyzowana wartość reprezentuje pewien przedział liczbowy. Jeśli rzeczywiste optymalne parametry transformacji znajdują się na krawędzi dyskretyzowanych przedziałów, wtedy może się zdarzyć, że w wyniku błędów zaokrągleń oraz niedokładności w procesie ekstrakcji cech, głosy rozłożą się między kilka sąsiednich parametrów przekształcenia geometrycznego. W wyniku takiego podziału głosów może się okazać, że inne – w rzeczywistości suboptymalne – rozwiązanie uzyskało największą liczbę głosów.

W celu zapobiegania takim sytuacjom proponuje się, żeby w każdej iteracji oddawać głos nie tylko na wyznaczone wartości parametrów, ale także na wartości sąsiednie. Aby nie doprowadzić w ten sposób do powstania kilku rozwiązań, na które oddana zostanie taka sama ilość głosów, proponuje się, by na główne parametry oddawany był głos podwójny, w celu podkreślenia jej wyższej wagi, a na parametry sąsiednie tylko głos pojedynczy. Zmodyfikowany algorytm zapisać można w następujący sposób:

dla każdej minucji $m_i \in T, i = 1, \dots, m$

dla każdej minucji $m'_j \in I, j = 1, \dots, n$

dla każdego kąta $\theta^+ \in \{\theta_1^+, \theta_2^+, \dots, \theta_c^+\}$

jeśli $dd(\theta'_j + \theta^+, \theta_i) < \theta_0$ {

$$\begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} x_i \\ y_i \end{bmatrix} - \begin{bmatrix} \cos \theta^+ & -\sin \theta^+ \\ \sin \theta^+ & \cos \theta^+ \end{bmatrix} \begin{bmatrix} x'_j \\ y'_j \end{bmatrix}$$

$\Delta x^+, \Delta y^+ =$ kwantyzacja $\Delta x, \Delta y$ do najbliższych wartości dyskretnych

$A[\Delta x^+, \Delta y^+, \theta^+] ++;$

dla $i = -1, \dots, 1$ {

dla $j = -1, \dots, 1$ {

dla $k = -1, \dots, 1$ {

$A[\Delta x^+ + i, \Delta y^+ + j, \theta^+ + k] ++;$

}

}

}

}

3.3.2 Wyznaczanie stopnia podobieństwa odcisków

Gdy wyznaczone zostanie optymalne przekształcenie geometryczne jakiemu trzeba poddać dwa odciski palców, aby najlepiej je do siebie dopasować, należy zdecydować czy odciski te są identyczne czy też nie. W niniejszej pracy proponuje się zastosowanie algorytmu, który próbuje znaleźć na dwóch odciskach określoną liczbę pokrywających się minucji. Jeśli liczba pokrywających się minucji okaże się większa od pewnej ustalonej ilości, odciski zostaną uznane za identyczne. W przeciwnym wypadku uznaje się, że pochodzą one z dwóch różnych palców. Zaproponowana metoda jest analogiczna do podejścia stosowanego do stwierdzania identyczności odcisków palców w kryminalistyce. Zaproponowany algorytm można zapisać za pomocą następującego pseudokodu (T oznacza szablon odcisku palca, a I odcisk który próbujemy do tego szablonu dopasować):

```
czy_odciski_są_identyczne( T, I ) {  
  ilość_dopasowanych_minucji = 0;  
  dla każdej minucji  $m_i \in T, i = 1, \dots, m$   
    dla każdej minucji  $m'_j \in I, j = 1, \dots, n$   
      jeśli ( minucje_się_pokrywają(  $m_i, m'_j$  ) ) {  
        ilość_dopasowanych_minucji++;  
        oznacz minucję  $m'_j$  jako dopasowaną;  
        przerwij wewnętrzną pętlę (przejdź do następnej minucji z T);  
      }  
    return ilość_dopasowanych_minucji >= ustalony_próg;  
}
```

Sprawdzanie, czy dwie minucje się pokrywają, odbywa się poprzez sprawdzenie ich wzajemnego położenia oraz kierunku. W celu skompensowania drobnych błędów, zarówno w procesie ekstrakcji cech jak i w trakcie wyznaczania przekształcenia geometrycznego, wprowadza się pewne zakresy tolerancji. Tak więc dwie minucje uznawane są za pokrywające się, jeżeli ich odległość jest mniejsza od pewnego ustalonego progu oraz różnica ich kierunków jest mniejsza od pewnego kąta tolerancji. Odpowiada to funkcji $mm(\cdot)$ opisanej w Rozdziale 2.4. To proste podejście pozwala na uzyskanie dobrych rezultatów praktycznych.

Rozdział 4

Technologie i narzędzia

Poniższy rozdział zawiera opis narzędzi i technologii wykorzystanych do stworzenia aplikacji rozpoznającej odciski palców oraz do napisania pracy. Zdecydowano się na użycie otwartych technologii oraz na wykorzystanie oprogramowania Open Source, w szczególności systemu operacyjnego Linux.

4.1 Język programowania

Do stworzenia implementacji zaproponowanego algorytmu wybrany został język Java [19] w wersji 1.5. Wybór języka podyktowany był kilkoma czynnikami. Najnowsze wersje języka Java, dzięki optymalizacji programu przed jego uruchomieniem¹, oferują wydajność, która może być znacznie szybsza od C++. Oczywiście podczas uruchamiania aplikacji potrzebny jest dodatkowy, niewielki nakład czasu na skompilowanie kodu. Dlatego też wzrost wydajności daje się najlepiej odczuć w przypadku działających przez długi czas aplikacji serwerowych. Aby więc osiągnąć maksymalne korzyści oferowane przez język, można by w przyszłości wykorzystać stworzony algorytm w aplikacji serwerowej do obsługi bazy danych odcisków palców.

Nie bez znaczenia dla wyboru języka była także bardzo dobra jego znajomość, pozwalająca na bezproblemowe zmiany w strukturze programu oraz szybkie testowanie nowych pomysłów. Istotnym czynnikiem była również dostępność dla języka Java roz-

¹ang. *Just-In-Time compilation*, w skrócie JIT. Metoda polegająca na kompilowaniu bajtkodu do kodu maszynowego tuż przed wykonaniem programu. Pozwala to na optymalizację kodu pod kątem maszyny, na której program jest uruchamiany.

budowanych narzędzi wspierających proces tworzenia oprogramowania. Jednym z nich jest środowisko programistyczne omówione w następnym punkcie.

4.2 Środowisko programistyczne

Do wykonania projektu wybrano środowisko programistyczne Eclipse [20]. Czynnikiem decydującym była dobra znajomość tego środowiska, pozwalająca na pełne wykorzystanie jego możliwości. Ważnym aspektem okazała się również możliwość dokonywania w krótkim czasie zmian w dużych partiach kodu. Eclipse umożliwia to poprzez rozbudowane opcje refaktoringu oraz automatycznego generowania kodu. Istotnym atutem tego środowiska jest także jego wysoka stabilność i szybkość działania oraz dostępność dodatkowych wtyczek. Dzięki wtyczkom możliwa była łatwa integracja projektu z systemem kontroli wersji Subversion (za pomocą wtyczki Subclipse [29]) oraz kontrola przejrzystości tworzonego kodu (za pomocą wtyczki Checkstyle [30]).

4.3 Techniki programistyczne

Za kluczowy element związany z powodzeniem części praktycznej pracy uznano tworzenie implementacji zgodnie z regułami inżynierii oprogramowania oraz stosowanie metodyk, które w dużym stopniu pozwolą na unikanie błędów.

Algorytm identyfikacji odcisków palców wykonuje wiele obliczeń numerycznych oraz liczne operacje na strukturach danych. Poprawne działanie wszystkich algorytmów składowych jest niewątpliwym elementem mającym istotny wpływ na poprawne działanie procesu rozpoznawania. Potrzebne więc było narzędzie, które pozwoliłoby na testowanie fragmentów powstającego kodu na długo przed tym jak ukończona zostanie cała aplikacja. Doskonałym wyjściem z sytuacji było zastosowanie testów jednostkowych [11] i integracyjnych w połączeniu z metodyką *Programowania Sterowanego Testami* (ang. *Test Driven Development*, w skrócie TDD). W tym podejściu do programowania, zanim powstanie jakikolwiek fragment kodu, tworzone są testy mające sprawdzać poprawność jego działania. Testy, które sprawdzają działanie poszczególnych funkcji nazywane są jednostkowymi. Jeśli zaś testuje się wzajemne współdziałanie kilku rzeczywistych obiektów, mówi się o testach integracyjnych.

Zgodnie z podejściem Test Driven Development dla każdej tworzonej funkcji określano najpierw przykładowe zestawy danych testowych wraz z wynikami, jakie powinny zostać otrzymane dla tych danych. Gdy testy były gotowe, przystępowano do tworzenia kodu, który przechodziłby wszystkie stworzone dla niego testy. Oczywiście jest, że nie da się przetestować wszystkich możliwych wariantów danych wejściowych. Najważniejsze jest więc testowanie różnych przypadków szczególnych. Jeśli bowiem algorytm radzi sobie poprawnie z przypadkami szczególnymi, to nie powinien mieć także problemu ze zwykłymi². Dzięki zastosowaniu testów uzyskano pewność, że tworzone algorytmy działają tak, jak się tego od nich oczekuje oraz, że nie popełniono podczas ich implementacji błędów. Stosowanie Test Driven Development powoduje spowolnienie procesu tworzenia oprogramowania, ze względu na konieczność tworzenia podwójnej ilości kodu. Nakład czasowy włożony w przygotowanie testów zwraca się jednak w końcowych etapach tworzenia aplikacji, gdyż ilość błędów w aplikacji zostaje radykalnie zmniejszona.

Uznano, że ważnym elementem tworzenia aplikacji jest zastosowanie wzorców projektowych, gdyż zapewniają one gotowe rozwiązania pojawiających się problemów projektowych. Dzięki ich zastosowaniu proces tworzenia kodu aplikacji stał się dużo szybszy. Tworzony kod stał się dużo bardziej elastyczny i gotowy na bezproblemowe przyszłe rozszerzenia. Był to element istotny, ze względu na planowane testowanie różnych zmian w algorytmie oraz dodawanie do niego nowych elementów. Jednocześnie należy zaznaczyć, że tworzenie kodu przygotowanego na przyszłe modyfikacje zaowocowało nieznacznym spadkiem wydajności aplikacji. Jedną z przyczyn tego może być mechanizm polimorfizmu, który pojawia się w wielu fragmentach kodu. Jego zastosowanie umożliwia łatwe dodawanie nowych elementów do aplikacji (wystarczy odziedziczyć odpowiednią klasę bądź interfejs). Z drugiej jednak strony takie podejście powoduje wolniejsze wykonywanie programu, ponieważ w trakcie działania należy określić rzeczywistą klasę danego obiektu. Zdecydowano jednak, że tworzona aplikacja ma charakter badawczy i najważniejsze jest osiągnięcie jak najwyższej skuteczności działania oraz przeanalizowanie różnych wersji stosowanych algorytmów. Wydajność została więc uznana za kwestię drugorzędą.

²Przykład: przypadkiem szczególnym w algorytmie pobierania prostej linii z obrazka może być sytuacja, gdy fragment tej linii miałby wyjść poza obrazek. Nie można wtedy dopuścić do próby adresowania pikseli poza granicami obrazu, gdyż spowodowałoby to wyjątek.

4.4 Użyte biblioteki

Nie sposób tworzyć projekt bez korzystania z gotowych, już istniejących bibliotek zewnętrznych. Oferują one gotowe implementacje często wykorzystywanych funkcjonalności, które nie są wbudowane bezpośrednio w bibliotekę języka. W tworzonej aplikacji zdecydowano się na wykorzystanie kilku pomocniczych darmowych bibliotek dostępnych dla języka Java. Poniżej zaprezentowano ich omówienie wraz z uzasadnieniem dlaczego, zdecydowano o ich wykorzystaniu.

TestNG

Biblioteka TestNG [21] związana jest z zastosowaniem testów jednostkowych i integracyjnych w procesie tworzenia aplikacji. O jej wyborze zdecydowała przede wszystkim jej dobra znajomość. Biblioteka dostarcza funkcjonalności potrzebnych do skutecznego tworzenia testów. Możliwe jest szybkie dodawanie nowych testów, rozszerzanie istniejących oraz dostarczanie wielu zestawów różnych danych testowych do jednego testu. Istnieje wiele możliwych asercji, które można sprawdzać (jedną z bardziej ciekawych jest żądanie, by przy danym zestawie danych wejściowych wywołanie metody testowej powodowało rzucenie przez nią określonego wyjątku). Stworzone testy mogą być łączone w zestawy i uruchamiane jednocześnie, jak również można stosować warunkowe uruchamianie testów (uruchomienie jednego testu zależy od powodzenia bądź niepowodzenia innego testu).

EasyMock i EasyMock Class Extension

Ideą stosowania testów jednostkowych jest testowanie działania pojedynczych metod. Jednak każdy system komputerowy składa się z wielu powiązanych ze sobą elementów i bardzo często okazuje się, że chcąc przetestować jedną metodę wymagane jest utworzenie obiektów innych klas niezbędnych do działania testowanej klasy. Rodzi się więc problem, ponieważ działanie testowanej metody zależy nie tylko od niej samej, ale także od działania innych klas. Wyjściem z tej kłopotliwej sytuacji jest stosowanie tzw. moków (ang. *mock objects*), czyli obiektów, które symulują działanie obiektów dowolnych klas [11]. Obiekt taki tworzony jest na czas testu i zastępuje rzeczywisty obiekt symulowanej klasy. Biblioteki EasyMock oraz EasyMock Class Extension [25] dostar-

czają funkcjonalności niezbędnych do tworzenia moków. O ich wyborze zdecydowała szeroka gama oferowanych możliwości. Jedną z najważniejszych jest możliwość tzw. nagrywania moków, czyli definiowania, jakich odwołań do danej klasy spodziewamy się w trakcie działania testowanej metody oraz jakie wyniki ma zwrócić mok w odpowiedzi na wywołanie tych metod. Razem z biblioteką EasyMock Class Extension do projektu dołączona została biblioteka cglib, niezbędna do jej działania.

log4j

Biblioteka log4j [24] służy do tworzenia logów w trakcie działania aplikacji. Logi uznano za niezbędny element informujący o przebiegu obliczeń w programie. O wyborze biblioteki zdecydowała jej duża elastyczność. Cała konfiguracja systemu logowania skupiona jest w jednym pliku, który pozwala na przełączanie szczegółowości logowania bez ingerencji w kod³. Za pomocą tego samego pliku właściwości można również modyfikować format logów oraz wyjście, na które będą kierowane (przykładowo może to być plik lub konsola). Parametry mogą zostać określone osobno dla jednej klasy, wybranego pakietu bądź całej aplikacji. Istotnym pytaniem, które pojawia się przy wprowadzaniu rozbudowanego systemu logowania, jest jego wpływ na szybkość działania aplikacji. Praktyka pokazuje, że po włączeniu pełnego logowania wszystkich klas program działa znacznie wolniej. Spowodowane jest to czasem potrzebnym na konwersję logów do formatu określonego przez użytkownika oraz zapisanie logów na podany strumień wyjściowy. W przypadku wyłączenia wszystkich logów obecność w kodzie instrukcji logowania nie prowadzi do spowolnienia szybkości działania. Dodatkowy narzut czasowy jest wtedy związany jedynie z czasem potrzebnym na oszacowanie wartości parametrów funkcji oraz wywołanie tej funkcji i natychmiastowy powrót z niej. Dlatego też w ostatecznej wersji aplikacji logowanie należy wyłączyć.

XStream

Biblioteka XStream [22] służy do serializacji obiektów języka Java do formatu XML. Za pomocą jednej linijki kodu można przekształcić dowolny istniejący obiekt do formatu XML, jak również deserializować obiekt z poprawnie stworzonego pliku. W projek-

³Biblioteka dostarcza też możliwość konfigurowania systemu logów z poziomu kodu. Jest to jeden z przykładów wspomnianej elastyczności.

cie zastosowano bibliotekę do łatwego wczytywania konfiguracji oraz podczas testów jednostkowych, do porównywania obiektów⁴. Zaletą tej biblioteki jest jej wysoka wydajność i niskie zużycie pamięci. Ponadto nie wymaga ona – w przeciwieństwie do innej popularnej biblioteki Betwixt [23] – tworzenia dodatkowych plików określających mapowanie danych pomiędzy obiektami Java a plikiem XML. Nie są wymagane również żadne modyfikacje obiektów w celu serializacji pól prywatnych. Jest to niewątpliwą zaletą, ponieważ korzystanie z biblioteki nie pociąga za sobą konieczności modyfikacji programu.

4.5 Baza odcisków palców

Aby móc porównać skuteczność stworzonego algorytmu z już istniejącymi potrzebna była baza danych odcisków palców. Posłużono się jedną z czterech darmowych baz udostępnionych w ramach konkursu Fingerprint Verification Competition 2004 [26].



Rysunek 4.1: Przykładowy odcisk palca z użytej bazy danych

Baza ta została stworzona na Uniwersytecie Bolońskim, który jest organizatorem konkursu. Odciski palców pochodzą od studentów, którzy zgłosili się jako ochotnicy.

⁴Próba porównania identyczności dwóch obiektów za pomocą asercji TestNG skończyłaby się na porównaniu wskaźników, które z racji porównywania dwóch instancji zawsze będą różne. Dopiero po serializacji wszystkich pól możliwe staje się porównywanie zawartości obiektów.

Odciski pobierane były w trzech sesjach, każda odległa od poprzedniej o przynajmniej dwa tygodnie. Każdemu z ochotników pobrano odciski palców wskazującego i środkowego obu dłoni (łącznie cztery), w celu zmaksymalizowania różnic w sposobie umieszczenia palca na czytniku. Nie dołożono żadnych specjalnych starań w celu zapewnienia dobrej jakości pobieranych odcisków, powierzchnie czytników nie były czyszczone pomiędzy poszczególnymi sesjami. W czasie pierwszej sesji pobierania odcisków palców ochotników proszono, by umieszczali palec na czytniku nieco przesuwając go od środka oraz by zmieniali siłę nacisku na czytnik. W trakcie drugiej sesji ochotników proszono by postarali się zniekształcić odcisk poprzez jego dociśnięcie i przesunięcie bądź obrót. W trakcie trzeciej sesji palce były zwilżane lub wysuszane. W wybranej bazie znajdują się odciski 100 różnych palców, każdy pobrany na 8 różnych sposobów co daje łącznie 800 próbek. Dostępna jest także druga baza wykonana z użyciem tego samego czytnika zawierająca odciski 10 różnych palców, każdy zeskanowany po 8 razy. Mniejsza baza udostępniana była uczestnikom konkursu w celu dostrojenia algorytmów. Większa baza służyła do testowania skuteczności działania nadesłanych algorytmów i została udostępniona dopiero po konkursie. Przykładowy odcisk z tej bazy danych pokazano na Rysunku 4.1.



Rysunek 4.2: Sztucznie wygenerowane odciski palców

Do stworzenia wybranej bazy wykorzystano czytnik optyczny Verifier 300 firmy CrossMatch [27]. Pobrane za jego pomocą odciski zapisane są w plikach graficznych o wymiarach 640 na 480 z rozdzielczością 500dpi w skali szarości.

Dodatkowym aspektem, który zadecydował o wyborze tej bazy danych jest dostęp-

ność na stronie organizatorów wyników skuteczności działania poszczególnych algorytmów nadesłanych do konkursu. Dzięki temu istniała możliwość łatwego porównania skuteczności stworzonego algorytmu z już istniejącymi.

Ponadto zdecydowano o wykorzystaniu zestawu sztucznie wygenerowanych odcisków palców. Do ich utworzenia użyto programu SFinGe [18] dostępnego w darmowej wersji. Program pozwala na wygenerowanie odcisków bardzo wysokiej jakości, co było istotne podczas testowania działania algorytmu na etapie projektowania aplikacji. Możliwe jest także generowanie odcisków z zakłóceniami tak, aby były one jak najbardziej zbliżone do odcisków naturalnych. Przykłady odcisków utworzonych przy użyciu SFinGE pokazane są na Rysunku 4.2

4.6 Aplikacja

Informacje zawarte w tym podrozdziale mają umożliwić zrozumienie budowy projektu i ułatwić odnalezienie w jego strukturze fragmentów kodu odpowiedzialnych za realizowanie poszczególnych algorytmów składowych omówionych w Rozdziałach 2 i 3. Instrukcja obsługi aplikacji została zamieszczona w Dodatku A.

4.6.1 Struktura katalogów projektu

Pliki projektu zostały rozdzielone pomiędzy różne katalogi. Każdy z podkatalogów znajdujących się w katalogu głównym projektu został utworzony do przechowywania plików ściśle określonego typu. Poniżej zamieszczono omówienie poszczególnych katalogów wraz z ich zawartością.

bin : skompilowane pliki źródłowe wraz z wszystkimi niezbędnymi plikami zasobów.

debug : pliki używane do debugowania programu w trakcie jego powstawania. W podkatalogu `steps/` umieszczane są klatki pokazujące kolejne kroki algorytmu, które można następnie wykorzystać do utworzenia filmu za pomocą linuxowego skryptu `create_movie.sh`.

doc : dokumentacja programu. Zawiera ten dokument, jego wersję źródłową sformatowaną za pomocą \LaTeX (pliki z rozszerzeniem `.tex`), pliki graficzne wykorzy-

stywane w tym dokumencie oraz skrypt `korekta.sh` wykonujący automatyczną korektę tekstu w plikach z rozszerzeniem `.tex`.

itest : testy integracyjne. Struktura podkatalogów odpowiada strukturze pakietów projektu. Plik `itestng.xml` określa wszystkie uruchamiane testy integracyjne.

lib : zewnętrzne biblioteki dołączone do projektu.

rc : pliki zasobów niezbędne do działania aplikacji. Katalog zawiera pliki konfiguracyjne, plik właściwości systemu logów `log4j.properties` oraz plik `MANIFEST.MF` używany do generowania pliku `.jar` (dystrybucyjnej wersji aplikacji).

src : pliki źródłowe aplikacji. Struktura podkatalogów katalogu `src/` tworzy jednocześnie strukturę pakietów projektu.

test : testy jednostkowe. Struktura podkatalogów odpowiada strukturze pakietów projektu. Plik `testng.xml` określa wszystkie uruchamiane testy. W podkatalogu `rc/` znajdują się zasoby potrzebne do testów.

Pliki i katalogi, których nazwy poprzedzone są kropką, zawierają dane konfiguracyjne środowiska Eclipse oraz systemu kontroli wersji Subversion. Dzięki umieszczeniu testów jednostkowych oraz testów integracyjnych w osobnych katalogach, uzyskuje się oddzielenie kodu testowego od właściwego kodu aplikacji. Pozwala to na łatwe usunięcie kodu testowego z dystrybucyjnej wersji aplikacji.

4.6.2 Struktura pakietów

Klasy tworzące kod źródłowy zostały pogrupowane na pakiety w zależności od realizowanych funkcjonalności. Poniżej zaprezentowano omówienie roli poszczególnych pakietów.

data : struktury danych, na których operuje aplikacja

debug : klasy wykorzystywane do debugowania aplikacji w trakcie jej powstawania

exceptions : wyjątki rzucane przez aplikację

extraction : klasy odpowiedzialne za proces ekstrakcji cech

background : segmentacja odcisku palca z tła

configuration : dostarczanie parametrów konfiguracyjnych do pozostałych klas odpowiedzialnych za ekstrakcję cech

directions : obliczanie mapy kierunków

minutiae : śledzenie linii papilarnych, ekstrakcja minucji oraz wykrywanie fałszywych minucji

nextline : wyznaczanie nowych punktów początkowych dla algorytmu śledzącego linie

section : tworzenie sekcji, poprawa jakości sekcji oraz wyznaczanie minimum

image : wczytywanie plików graficznych z odciskami palców oraz ich przetwarzanie wstępne

main : klasy służące do uruchamiania aplikacji i przetwarzania parametrów programu

matching : klasy odpowiedzialne za dopasowywanie wzorców

alignment : wyznaczanie optymalnego przekształcenia geometrycznego pozwalającego na dopasowanie odcisków

configuration : dostarczanie parametrów konfiguracyjnych do pozostałych klas odpowiedzialnych za dopasowywanie wzorców

util : klasy odpowiedzialne m.in. za zarządzanie zasobami i wykonywanie obliczeń pomocniczych

Rozdział 5

Badania eksperymentalne

W poniższym rozdziale stworzony program rozpoznający odciski palców zostanie poddany weryfikacji. Celem jest sprawdzenie skuteczności zaproponowanych rozwiązań oraz zbadanie wpływu parametrów na proces rozpoznawania.

Do badań użyto dwóch baz odcisków palców. Jedna baza zawiera odciski wygenerowane sztucznie za pomocą programu SFinGe, druga zawiera odciski rzeczywiste zaczerpnięte z bazy danych opisanej w Rozdziale 4.5. Każda z dwóch baz składa się z ośmiu odcisków palców. Każdy odcisk zeskanowany jest czterokrotnie, przy czym poszczególne ujęcia tego samego odcisku różnią się pomiędzy sobą przesunięciem i obrotem. Należy podkreślić, że taka struktura bazy danych służy wyłącznie testowaniu skuteczności aplikacji. W rzeczywistych systemach każdemu użytkownikowi systemu odpowiada jeden wpis w bazie. Wymagana przez program struktura bazy danych została omówiona w Dodatku A.

Za najważniejsze mierniki jakości rozpoznawania uznano wskaźniki FNMR i FMR omówione w Rozdziale 1.3. Wskaźnik FNMR oblicza się poprzez porównanie każdego obrazu odcisku palca z innymi ujęciami tego samego odcisku. W przypadku użytych baz każdy odcisk jest porównywany z trzema innymi ujęciami. Na podstawie stosunku ilości niedopasowań do całkowitej liczby porównań obliczana jest wartość wskaźnika. Wskaźnik FMR oblicza się dokonując porównania każdego obrazu odcisku palca z wszystkimi pozostałymi ujęciami innych odcisków. W przypadku użytych baz jest to 28 innych obrazów. Wartość wskaźnika wyznacza się na podstawie stosunku ilości dopasowań do całkowitej liczby porównań. Pomiar skuteczności działania systemu wykonane zostały dla każdej z baz.

5.1 Ogólne wyniki

Metodą doświadczalną wyznaczono, dla każdej z baz, wartości parametrów dające najlepsze rezultaty (najmniejsze wartości współczynników FMR i FNMR). Stały się one podstawą wykorzystywaną w dalszych badaniach. W każdym z następných pomiarów testowano zmianę jednej wartości parametru konfiguracyjnego (chyba, że zaznaczono inaczej), przy zachowaniu optymalnych wartości pozostałych parametrów. Poniżej zaprezentowano wyniki działania programu przy optymalnych ustawieniach.

5.1.1 Skuteczność rozpoznawania

Wskaźnik	Obrazy sztuczne	Obrazy rzeczywiste
FNMR	1.04%	27.08%
FMR	0.89%	1.34%

Tabela 5.1: Skuteczność weryfikacji użytkowników

Skuteczność rozpoznawania sztucznych odcisków palców jest, jak widać w Tabeli 5.1, wysoka. Wskaźnik FMR większy od 0 oznacza, że system mógłby dopuścić osoby nieuprawnione. Dalej zostanie pokazane, że można zmniejszyć FMR do zera poprzez zmianę parametrów dopasowywania wzorców. Wyniki uzyskane dla obrazów rzeczywistych są gorsze. Szczególnie wysoki jest wskaźnik FNMR (27.08%).

5.1.2 Szybkość działania

Ilość odcisków	Obrazy sztuczne	Obrazy rzeczywiste
1	1.55	1.91
8	0.57	0.76
16	0.45	0.59
32	0.31	0.51

Tabela 5.2: Średni czas ekstrakcji cech podany w sekundach

Zmierzone średnie czasy ekstrakcji cech (Tabela 5.2) i dopasowywania wzorców (Tabela 5.3) potwierdzają, poczynione w Rozdziale 4.1, założenia dotyczące szybkości działania aplikacji napisanej w języku Java. Widać wyraźnie, że w przypadku jednokrotnej

Ilość porównań	Średni czas
1	1.18
56	0.26
240	0.15
992	0.17

Tabela 5.3: Średni czas dopasowywania wzorców podany w sekundach

ekstrakcji cech bądź dopasowywania wzorców, narzut czasowy, związany z kompilacją podczas uruchamiania, jest spory. Jeśli jednak obliczenia dokonywane są wielokrotnie, średni czas operacji kilkukrotnie spada. Widać też, że dla obrazów rzeczywistych czas ekstrakcji cech jest nieznacznie dłuższy. Zmierzony czas działania obejmuje również czas wczytywania pliku z dysku, przetwarzania wstępnego oraz zapisywania wyników na dysk.

5.2 Ekstrakcja cech

W tym podrozdziale zaprezentowana zostanie dyskusja wyników uzyskanych podczas badania ulepszeń ekstrakcji cech zaproponowanych w Rozdziale 3.

5.2.1 Przetwarzanie wstępne

W Rozdziale 3.1 zaproponowano zastosowanie dwóch technik przetwarzania wstępnego. Przycięcie obrazu miało prowadzić do ograniczenia rejonu działania algorytmów do obszaru faktycznie zajmowanego przez odcisk palca. Filtracja medianowa powinna poprawić jakość obrazu usuwając z niego część zakłóceń, a tym samym podnieść skuteczność rozpoznawania.

Wskaźnik	Obrazy sztuczne	Obrazy rzeczywiste
FNMR	2.08%	33.33%
FMR	0.00%	4.02%

Tabela 5.4: Skuteczność rozpoznawania przy braku przetwarzania wstępnego

Tabela 5.4 przedstawia rezultaty uzyskane w wyniku wyłączenia przetwarzania wstępnego w postaci filtracji medianowej. W przypadku obrazów sztucznych (które są czarno-białe) filtracja medianowa nie powinna w istotny sposób wpłynąć na poprawę jakości obrazu, dlatego też niewielkie zmiany współczynników FMR i FNMR należy uznać za nieistotne. Dużo ważniejsze są wyniki uzyskane dla rzeczywistych odcisków palców. Widać, że brak filtracji medianowej pogorszył FNMR o 6.25%, a FMR o 2.68%. Dowodzi to, że zastosowanie filtracji jest celowe, gdyż prowadzi do poprawy skuteczności ekstrakcji cech.

5.2.2 Mapa kierunków

W Rozdziale 3.2.2 zaproponowano wykorzystanie nieinterpolowanej mapy kierunków. Postawiono hipotezę, że można w ten sposób uzyskać zbliżoną skuteczność działania przy jednoczesnym skróceniu czasu obliczeń.

Ilość odcisków	Obrazy sztuczne	Obrazy rzeczywiste
1	1.51	1.88
8	0.54	0.75
16	0.39	0.58
32	0.32	0.51

Tabela 5.5: Średni czas ekstrakcji przy zastosowaniu nieinterpolowanej mapy kierunków

Z Tabeli 5.5 widać, że średni czas ekstrakcji cech, zarówno dla obrazów sztucznych jak i rzeczywistych, uległ zmniejszeniu, które jest jednak tak małe, że mieści się w granicach błędu pomiarowego.

Wskaźnik	Obrazy sztuczne	Obrazy rzeczywiste
FNMR	7.29%	31.25%
FMR	0.00%	2.68%

Tabela 5.6: Skuteczność rozpoznawania przy zastosowaniu nieinterpolowanej mapy kierunków

Tabela 5.6 pokazuje uzyskaną skuteczność rozpoznawania przy zastosowaniu nieinterpolowanej mapy kierunków. W przypadku obrazów sztucznych widać wzrost FNMR

o 6.25%, przy jednoczesnym spadku FMR do zera. Dla obrazów rzeczywistych wskaźniki pogorszyły się: FNMR o 4.17%, FMR o 1.34%. Pokazuje to, że postawiona hipoteza dotycząca nieinterpolowanej mapy kierunków nie sprawdziła się i celowe jest korzystanie z mechanizmu interpolacji.

5.2.3 Wyznaczanie nowych punktów początkowych

W Rozdziale 3.2.3 zaproponowano, aby ulepszyć algorytm wyznaczania nowych punktów początkowych poprzez uwzględnianie poziomu jasności pikseli. Uznano, że to podejście pozwoli na wyeliminowanie nieprawidłowo wyznaczonych punktów początkowych w sytuacji, gdy linie papilarne w którymś fragmencie obrazu są zbyt oddalone od siebie.

Wskaźnik	Obrazy sztuczne	Obrazy rzeczywiste
FNMR	0.00%	28.12%
FMR	0.45%	1.79%

Tabela 5.7: Wyznaczanie punktów początkowych bez uwzględniania poziomu jasności pikseli

W Tabeli 5.7 zaprezentowano wyniki uzyskane przy ignorowaniu poziomów jasności pikseli w trakcie wyznaczania punktów początkowych. Jak widać – wbrew oczekiwaniom teoretycznym – dla obrazów sztucznych uzyskano znaczącą poprawę wyników. FNMR zmniejszył się do zera, a FMR spadł o 0.44%. Zwiększenie ilości wymaganych dopasowanych minucji do ośmiu spowodowało wzrost FNMR do 2.08% i spadek FMR do zera. Jednak w przypadku obrazów rzeczywistych wskaźniki nieznacznie się pogorszyły: FNMR wzrósł o 1.04%, a FMR o 0.45%. Wskazuje to na słuszność zastosowanego podejścia.

Zaproponowano również, aby przetestować jaki wpływ na proces rozpoznawania będzie miało przyjęcie takiej gęstości siatki, aby każdy punkt obrazu był rozważany jako punkt początkowy. Wyniki zaprezentowano w Tabeli 5.8.

Jak widać, wartości wskaźników dla obrazów sztucznych zostały nieznacznie zmniejszone. Jednak, w przypadku obrazów rzeczywistych, nastąpiło niemożliwe do zaakceptowania, pomimo spadku FNMR o 13.54%, pogorszenie wskaźnika FMR o 11.27%. Prowadzi to do wniosku, że celowe jest zastosowanie metody wyznaczania punktów

Wskaźnik	Obrazy sztuczne	Obrazy rzeczywiste
FNMR	1.04%	13.54%
FMR	0.33%	12.61%

Tabela 5.8: Wyznaczanie punktów początkowych poprzez przeglądanie wszystkich pikseli obrazu (uwzględniono poziom jasności pikseli)

początkowych opartej na dyskretnej siatce o ustalonej odległości węzłów. Badania tego zagadnienia będą kontynuowane w Rozdziale 5.2.4.

5.2.4 Kryteria stopu

W Rozdziale 3.2.4 zaproponowano nowe warunki stopu dla algorytmu śledzącego przebieg linii papilarnej. Poniżej zaprezentowana została eksperymentalna analiza wpływu zastosowanych kryteriów stopu na skuteczność rozpoznawania.

Nadmierne wyginanie

Tabela 5.9 prezentuje wskaźniki skuteczności działania systemu przy włączonym nadmiernym wyginaniu jako kryterium stopu.

Wskaźnik	Obrazy sztuczne	Obrazy rzeczywiste
FNMR	2.08%	31.25%
FMR	0.00%	1.67%

Tabela 5.9: Wykorzystanie nadmiernego wyginania linii papilarnej jako kryterium stopu

Jak widać, dla obrazów sztucznych FMR zmalało do zera przy wzroście FNMR o 1.04%. Identyczna sytuacja miała miejsce przy wyłączeniu filtracji medianowej. Dla obrazów rzeczywistych zaobserwowano – zgodnie z przypuszczeniami – spadek skuteczności rozpoznawania. FNMR spadł o 4.17%, a FMR o 0.33%. Jest to bardzo niewielkie pogorszenie, co można wyjaśnić tym, że wprawdzie dużo przypadków wykrycia nadmiernego wyginania linii jest błędnych, jednak wiele fałszywych minucji jest później poprawnie usuwanych.

Przeprowadzono drugie badanie, w którym zmieniono parametr określający ilość

Wskaźnik	Obrazy sztuczne	Obrazy rzeczywiste
FNMR	6.25%	34.38%
FMR	0.00%	0.33%

Tabela 5.10: Wykorzystanie nadmiernego wyginania linii papilarnej jako kryterium stopu, przy wymaganym dopasowaniu ośmiu minucji

wymaganych dopasowanych minucji z 7 na 8 (Tabela 5.2.4). Tym razem wynik przedstawiał się zdecydowanie mniej korzystnie. Dla obrazów sztucznych FNMR wzrósł o 5.21%, a FMR spadł do zera. Dla obrazów rzeczywistych FNMR wzrósł o 7.3%, FMR spadł o 1.01%. Wzrost FNMR jest w pełni zrozumiały – jeśli wymagamy większej ilości dopasowanych minucji, maleje margines dla niedokładności w procesie ekstrakcji cech. Porównując uzyskane rezultaty z wynikami zaprezentowanymi w Tabeli 5.20 okazuje się, że są one gorsze. Prowadzi to do ostatecznego wniosku, że analiza wyginania linii, jako jedno z kryteriów stopu, nie sprawdza się.

Zakończenie linii

Wskaźnik	Obrazy sztuczne	Obrazy rzeczywiste
FNMR	14.58%	47.92%
FMR	32.37%	6.81%

Tabela 5.11: Wyłączenie ulepszonego sposobu wykrywania zakończeń linii

Tabela 5.11 przedstawia wyniki uzyskane przy wykorzystaniu niezmodyfikowanego kryterium wykrywania zakończeń linii opisanego w Rozdziale 2.3.2. Jak widać, nastąpiło pogorszenie wyników. W przypadku obrazów sztucznych FNMR wzrósł o 14.54%, a FMR o 31.48%. Dla obrazów rzeczywistych wskaźniki te spadły odpowiednio o 20.84% i 5.47%. Pokazuje to całkowitą słusność zastosowania zmodyfikowanej metody wykrywania zakończeń linii.

5.2.5 Wykrywanie fałszywych minucji

W Rozdziale 3.2.5 zaproponowano traktowanie minucji położonych zbyt blisko brzegu odcisku palca jako fałszywych. Tabela 5.12 pokazuje rezultaty uzyskane po wyłączeniu

czeniu usuwania minucji położonych na obrzeżu odcisku.

Wskaźnik	Obrazy sztuczne	Obrazy rzeczywiste
FNMR	0.00%	10.42%
FMR	11.83%	10.27%

Tabela 5.12: Nieusuwanie minucji położonych na obrzeżu odcisku palca

Jak widać, w przypadku obrazów sztucznych doprowadziło to do wzrostu FMR o 10.94%, przy jednoczesnym spadku FNMR do zera. Dla obrazów rzeczywistych FNMR spadł o 16.66%, przy wzroście FMR o 9.93%. Zgodnie z informacjami zawartymi w Rozdziale 1.3 można zmniejszyć wartość FMR – kosztem wzrostu FNMR – przesuwając próg t , który decyduje o zakwalifikowaniu odcisków jako identycznych. W przypadku stworzonej aplikacji progiem tym jest liczba dopasowanych minucji. Przeprowadzono więc kolejne badania.

Wskaźnik	Obrazy sztuczne	Obrazy rzeczywiste
FNMR	3.12%	20.83%
FMR	0.33%	0.33%

Tabela 5.13: Nieusuwanie minucji położonych na obrzeżu odcisku palca przy wymaganym dopasowaniu 10 minucji

Tabela 5.13 pokazuje wyniki uzyskane przy wymaganym dopasowaniu 10 minucji, przy jednoczesnym nieusuwaniu minucji na obrzeżach odcisku. Jak widać, w przypadku obrazów sztucznych FNMR wzrósł o 3.12%, przy spadku FMR o 11.5% w stosunku do poprzednich wyników. Dla obrazów rzeczywistych uzyskano polepszenie wartości wskaźnika FMR o 9.94%, przy pogorszeniu FNMR o 10.41%.

Wskaźnik	Obrazy sztuczne	Obrazy rzeczywiste
FNMR	7.29%	21.88%
FMR	0.00%	0.00%

Tabela 5.14: Zmniejszenie minimalnej odległości minucji od krawędzi odcisku palca przy wymaganym dopasowaniu 10 minucji

Aby możliwe było wyciągnięcie wiarygodnych wniosków, przeprowadzono kolejne badanie, którego wynik pokazano w Tabeli 5.14. Tym razem zdecydowano się na usuwanie minucji położonych bardzo blisko krawędzi odcisku – w odległości mniejszej niż 5 pikseli (jako ustawienie bazowe do wszystkich badań przyjęto wartość 15). Spadek FMR do zera w obu przypadkach należy uznać za bardzo dobry rezultat. Jednocześnie można zauważyć wzrost FNMR w obu przypadkach. Dla obrazów sztucznych wynosi on 4.17% w stosunku do poprzednich wyników, czego nie można uznać za dopuszczalny rezultat. Jednak dla obrazów rzeczywistych wzrost FNMR jest niewielki – tylko o 1.05% w porównaniu z wcześniejszym badaniem.

Powyższe eksperymenty pokazują, że wskazane jest usuwanie minucji położonych w odległości mniejszej niż kilka pikseli od krawędzi odcisku palca. Umożliwia to uzyskanie lepszych wyników, nawet przy zwiększeniu wymaganej ilości dopasowanych minucji (co oznacza, że wynik rozpoznania jest bardziej wiarygodny).

5.2.6 Polepszanie jakości sekcji

W Rozdziale 2.3.2 opisano dwie metody stosowane do usunięcia szumów z wyznaczonej sekcji. W Tabeli 5.15 zaprezentowano wyniki uzyskane po wyłączeniu tych dwóch mechanizmów.

Wskaźnik	Obrazy sztuczne	Obrazy rzeczywiste
FNMR	7.29%	17.71%
FMR	7.14%	9.15%

Tabela 5.15: Wyłączone polepszanie jakości sekcji

Jak widać z Tabeli 5.15, w przypadku obrazów sztucznych nastąpił znaczący spadek skuteczności rozpoznawania. Zarówno FNMR jak i FMR spadły o 6.25%. Spowodowane jest to tym, że w przypadku tych obrazów przekrój poprzeczny linii ma płaski profil. Z tego powodu algorytm znajdujący minimum sekcji z reguły nie wyznacza minimum znajdującego się po środku linii papilarnej, lecz na jej obrzeżu. Prowadzi to do częstego gubienia linii, a tym samym wykrywania fałszywych zakończeń linii. Aby możliwe było skuteczne śledzenie linii na obrazie wygenerowanym sztucznie, konieczne jest zastosowanie drugiego kroku poprawy jakości sekcji, opisanego w Rozdziale 2.3.2. Dzięki temu przekrój linii traci swój płaski profil i możliwe staje się wyznaczenie prawidłowe-

go minimum. W przypadku obrazów rzeczywistych obserwujemy wzrost FMR o 7.81%, przy spadku FNMR o 9.37%. Pokazuje to, że poprawa jakości sekcji jest ważnym dla skuteczności algorytmu krokiem, z którego nie należy rezygnować.

5.3 Dopasowywanie wzorców

Ten podrozdział poświęcony zostanie dyskusji wyników uzyskanych podczas badania ulepszeń algorytmów dopasowywania wzorców zaproponowanych w Rozdziale 3.

5.3.1 Wyznaczanie przekształcenia geometrycznego

W Rozdziale 3.3.1 zaproponowano, by zastosować zmodyfikowany sposób głosowania, w którym głos oddawany jest na kilka komórek akumulatora.

Wskaźnik	Obrazy sztuczne	Obrazy rzeczywiste
FNMR	3.12%	23.96%
FMR	0.11%	2.68%

Tabela 5.16: Zastosowanie zmodyfikowanego sposobu głosowania

Tabela 5.16 pokazuje wyniki uzyskane przy zastosowaniu zmodyfikowanego sposobu głosowania. Dla obrazów sztucznych uzyskano poprawę FMR o 0.78%, kosztem wzrostu FNMR o 2.08%. W przypadku obrazów rzeczywistych zachodzi jednak sytuacja dokładnie odwrotna – to FNMR maleje o 3.12%, a FMR rośnie o 1.34%. Ponieważ nie pozwala to na wyciągnięcie jednoznacznego wniosku, przeprowadzono kolejny eksperyment. Zastosowano te same ustawienia co dla badań w Tabeli 5.14 (oczywiście z wyjątkiem zmodyfikowanego sposobu głosowania).

Wskaźnik	Obrazy sztuczne	Obrazy rzeczywiste
FNMR	10.42%	32.29%
FMR	0.00%	0.33%

Tabela 5.17: Zastosowanie zmodyfikowanego sposobu głosowania przy ustawieniach identycznych jak w Tabeli 5.14

Jak widać, w Tabeli 5.17 uzyskano gorsze wyniki niż w Tabeli 5.14. Dla obrazów sztucznych FMR utrzymało się na poziomie 0%, przy wzroście FNMR o 3.13% w sto-

sunku do wyników w Tabeli 5.14. Dla obrazów rzeczywistych FNMR wzrósł o 10.41%, a FMR wzrósł o 0.33%. Widać więc, że niewielkiemu pogorszeniu FMR towarzyszy także duży wzrost FNMR. Prowadzi to do konkluzji, że zmodyfikowany sposób głosowania nie przynosi zadowalających rezultatów i jego stosowanie nie jest wskazane.

Dokładność		Obrazy sztuczne		Obrazy rzeczywiste	
Obrót	Przesunięcie	FNMR	FMR	FNMR	FMR
4	4	3,12%	0.11%	27.08%	1.34%
5	5	0.00%	0.00%	25.00%	2.90%
6	6	4.17%	1.67%	32.29%	2.01%

Tabela 5.18: Zależność między dokładnością dyskretyzacji, a skutecznością rozpoznawania

Tabela 5.18 pokazuje zależność między dokładnością dyskretyzacji kąta i przesunięcia, a skutecznością rozpoznawania. Dla obrazów sztucznych możliwe staje się uzyskanie wyników, przy których zarówno FNMR jak i FMR uzyskują wartość 0, co jest rezultatem idealnym. Dla tych samych wartości dyskretyzacji dla obrazów rzeczywistych uzyskujemy zminimalizowanie FNMR i maksymalizację FMR (dla zastosowanych ustawień). Teoretycznie zmniejszenie stopnia dyskretyzacji powinno prowadzić do wzrostu dokładności dopasowania. W praktyce jednak niedokładności procesu ekstrakcji cech sprawiają, że idealne dopasowanie wzorców jest niemożliwe i zmniejszanie dokładności poniżej pewnej granicy prowadzić będzie do wzrostu błędu.

Tolerancja kątowna	Obrazy sztuczne		Obrazy rzeczywiste	
	FNMR	FMR	FNMR	FMR
5	26.04%	0.00%	65.62%	0.00%
15	5.21%	0.22%	27.08%	1.34%
25	1.04%	0.89%	25.00%	2.79%
180	4.46%	1.04%	31.25%	19.87%

Tabela 5.19: Zależność między tolerancją różnicy kątów, a skutecznością rozpoznawania

Tabela 5.19 prezentuje zależność skuteczności rozpoznawania od tolerancji kąto-

wej¹. Wyraźnie widać, że dla zbyt małej wartości dochodzi do bardzo dużego wzrostu FNMR. Wynika to z niedokładności w procesie ekstrakcji cech, a mówiąc ściślej z niedokładności przy ustalaniu kierunku minucji. Widać, że wzrost tolerancji prowadzi do zmniejszenia FNMR, przy jednoczesnym niewielkim wzroście FMR. Jeśli kierunek minucji zostanie zignorowany (ustawienie tolerancji na 180°) w przypadku obrazów rzeczywistych dochodzi do bardzo dużego wzrostu obu współczynników. Pokazuje to, że kierunek minucji ma duże znaczenie i powinien być uwzględniany przy jednoczesnym dopuszczeniu pewnej niedokładności jego wyznaczenia.

5.3.2 Wyznaczanie stopnia podobieństwa odcisków

Jako ostatni etap badania wpływu parametrów na działanie algorytmu, rozpatrzono zależność pomiędzy wymaganą ilością dopasowanych minucji, a skutecznością rozpoznawania. Wymagana ilość dopasowanych minucji jest opisanym w Rozdziale 1.3 progiem t , decydującym o uznaniu odcisków za identyczne bądź różne. Dlatego też oczekuje się, że zmniejszanie tej wartości prowadzi do wzrostu FMR i spadku FNMR oraz na odwrót – zwiększanie wartości prowadzi do zmniejszania się FMR i wzrostu FNMR.

Ilość minucji	Obrazy sztuczne		Obrazy rzeczywiste	
	FNMR	FMR	FNMR	FMR
5	1.04%	14.84%	18.08%	8.33%
6	1.04%	4.24%	19.79%	7.59%
7	1.04%	0.89%	27.08%	1.34%
8	3.12%	0.00%	30.21%	0.11%
9	7.29%	0.00%	37.50%	0.00%

Tabela 5.20: Ilość minucji koniecznych do dopasowania, aby uznać odciski za identyczne

Zaprezentowane w Tabeli 5.20 wyniki potwierdzają przewidzianą zależność. Wybór optymalnej wartości progu uzależniony jest od innych parametrów. Jak pokazano przy wcześniejszych pomiarach, dzięki dostosowaniu reszty parametrów możliwy jest

¹Patrz równanie 2.3 w Rozdziale 2.4.2

wybór większych wartości progu dających jednocześnie lepsze rezultaty od uzyskanych w Tabeli 5.20.

5.4 Optymalne ustawienia

Na podstawie przeprowadzonych eksperymentów wybrano ostateczne optymalne ustawienia. Zaprezentowano je w Tabeli 5.21.

Parametr	Wartość	
	Obrazy sztuczne	Obrazy rzeczywiste
Parametry ekstrakcji cech		
backgroundColor	255	255
backgroundThreshold	85	85
bendingHistorySize	2	2
bendingThreshold	60	60
directionThreshold	30	30
cropPadding	10	10
cropThreshold	45	45
directionMapStrategy	IM ²	IM ²
distanceBetweenSections	6	6
extendedSectionSize	1	1
gridSize	8	10
medianMaskSize	1	1
minimumDistanceBetweenMinutiae	7.0	7.0
minimumMinutiaeDistanceFromBorder	5.0	5.0
nextLineProviderStrategy	SNLP ³	SNLP ³
ridgeColor	30	30
ridgeThreshold	70	70
sectionSize	4	4
smoothingWindowSize	10	10
tangentWindowSize	8	8
useAverageSectionEnhancement	true	true

²INTERPOLATED_MAP

³STANDARD_NEXT_LINE_PROVIDER

Parametr	Wartość	
	Obrazy sztuczne	Obrazy rzeczywiste
useCropImagePreprocessing	true	true
useExcessiveBending	false	false
useGaussianSectionEnhancement	true	true
useImprovedNextLineProvider	false	true
useImprovedStopCriteria	true	true
useMedianImagePreprocessing	true	true
Parametry dopasowywania wzorców		
alignmentCalculatorType	SAC ⁴	SAC ⁴
matchingDirectionThreshold	25	15
matchingDistanceThreshold	20	20
minimumMatchedMinutiae	8	10
rotationBinSize	5	4
translationBinSize	5	4

Tabela 5.21: Optymalne wartości parametrów

Testowanie skuteczności systemu biometrycznego musi być oparte o inną bazę danych, niż ta, na której system był dostrajany. Spowodowane jest to tym, że w trakcie pracy system spotyka się danymi, z którymi nie miał wcześniej do czynienia (nie istnieje możliwość określenia z góry jak dokładnie będą wyglądały skanowane w przyszłości obrazy odcisków palców). W celu ostatecznego przetestowania skuteczności stworzonej aplikacji zastosowano dwie nowe bazy danych. Ich struktura jest identyczna jak baz wykorzystywanych do analizy parametrów aplikacji. Ostateczne wyniki skuteczności rozpoznawania przedstawiono w Tabeli 5.22.

Wskaźnik	Obrazy sztuczne	Obrazy rzeczywiste
FNMR	0.00%	16.67%
FMR	0.00%	3.91%

Tabela 5.22: Ostateczna zmierzona skuteczność weryfikacji użytkowników

⁴STANDARD_ALIGNMENT_CALCULATOR

5.5 Porównanie z istniejącymi algorytmami

Wyniki zaprezentowane w Tabeli 5.22 porównano z już istniejącymi algorytmami. W tym celu skorzystano z publicznie opublikowanych wyników konkursu Fingerprint Verification Competition 2004 [26]. Wzięły w nim udział 43 algorytmy nadesłane przez firmy, ośrodki akademickie oraz programistów działających na własną rękę. Tabela 5.23 przedstawia porównanie wyników uzyskanych na wybranej bazie danych przez kilka najlepszych algorytmów biorących udział w konkursie.

Algorytm	FMR	Średni czas ekstrakcji	Średni czas dopasowywania
P047	2.86%	1.95	1.87
P101	3.86%	0.10	3.19
P097	5.54%	0.74	0.75
P009	5.54%	0.25	0.23
P049	7.11%	0.32	0.47
P004	5.64%	0.77	0.75
P103	6.36%	0.16	0.16
Stworzony algorytm	3.91%	0.43	0.22

Tabela 5.23: Porównanie stworzonej aplikacji z innymi algorytmami rozpoznawania odcisków palców

Stworzona w ramach tej pracy aplikacja wypada, w porównaniu z najlepszymi wynikami z konkursu FVC, dosyć dobrze. Należy jednak pamiętać, że do testowania stworzonego algorytmu wykorzystano jedynie niewielki fragment bazy używanej w konkursie. Ponadto organizatorzy FVC nie podali dokładnej metody wyliczania wskaźnika FMR. W rzeczywistości korzystali oni z dwóch wskaźników – FMR100 i FMR1000 – nie podając jednak dokładnego ich znaczenia i sposobu obliczania. Tym samym zaprezentowane porównanie nie może być w pełni miarodajne.

Podsumowanie

W ramach powyższej pracy zaprezentowano przegląd teoretycznych zagadnień związanych z systemami biometrycznymi oraz zagadnieniem rozpoznawania odcisków palców. Przedstawiono już istniejące metody rozpoznawania odcisków, a także zaproponowano własne innowacje w tych metodach. Zastosowano nowe kryterium wykrywania minucji, polegające na uwzględnianiu poziomu jasności pikseli w trakcie śledzenia przebiegu linii papilarnej. Ponadto wprowadzono modyfikację algorytmu wyznaczania kierunku linii papilarnej. Jej idea oparta jest o wykorzystanie interpolacji metodą najbliższego sąsiada zamiast interpolacji liniowej, co ma na celu przyspieszenie obliczeń. W celu weryfikacji zaproponowanych rozwiązań stworzono aplikację komputerową implementującą omówione teoretycznie algorytmy. Eksperymenty wykazały wzrost poprawności rozpoznawania w przypadku zastosowania nowych kryteriów wykrywania minucji. Dla obrazów rzeczywistych błąd FMR – ilość fałszywych dopasowań – wyniósł 3.91%, podczas gdy błąd FNMR – ilość fałszywych niedopasowań – osiągnął wartość 16.67%. W przypadku zastosowania zaproponowanych w literaturze kryteriów wykrywania minucji błąd FMR wynosił 6.81%, a błąd FNMR osiągnął wartość 47.92%. Oznacza to, że zaproponowana metoda przyniosła prawie dwukrotny spadek błędów FMR i prawie trzykrotny spadek błędów FNMR. Jednocześnie wykazano, że nowa metoda wyznaczania kierunku linii papilarnych nie przynosi oczekiwanych rezultatów. Jej zastosowanie spowodowało wzrost błędów rozpoznawania: współczynnik FMR osiągnął 2.68%, a FNMR - 31.25%.

Wyniki osiągnięte przez stworzoną aplikację są zbliżone do rezultatów uzyskanych przez algorytmy biorące udział w konkursie Fingerprint Verification Competition w roku 2004. Należy jednak podkreślić, że w przeprowadzonej w 2006 roku edycji konkursu FVC, najlepsze algorytmy uzyskały błędy FMR rzędu 0.01%. Wyniki te pokazują, że zaimplementowana metoda powinna być w dalszym ciągu rozwijana i udoskonalana.

Dalsze prace w kierunku ulepszenia stworzonej aplikacji mogą obejmować:

- Udoskonalenie procesu ekstrakcji cech poprzez szczegółową analizę działania algorytmu śledzącego linie. Wskazane byłoby stworzenie interfejsu graficznego dającego możliwość wglądu w bieżące wartości zmiennych na każdym etapie działania algorytmu.
- Opracowanie metody pozwalającej na skuteczną ekstrakcję cech w przypadku obrazów słabej jakości.
- Zaprojektowanie algorytmu dopasowywania wzorców wykorzystującego mniejszą ilość pamięci. Wskazane byłoby stworzenie metody pozwalającej na jeszcze większą dokładność dopasowywania (wyznaczania przekształcenia geometrycznego).
- Stworzenie algorytmu umożliwiającego dopasowywanie jedynie fragmentów odcisków palców.
- Opracowanie metody wyliczania stopnia podobieństwa uwzględniającej nie tylko minucje, które do siebie pasują, ale także te, które do siebie nie pasują.

Bibliografia

- [1] Davide Maltoni, Dario Maio, Anil Jain, Salil Prabhakar, *Handbook of Fingerprint Recognition*, Springer 2003
- [2] Dario Maio, Davide Maltoni, *Direct Gray-Scale Minutiae Detection In Fingerprints*, 1997
- [3] Dario Maio, Davide Maltoni, *Neural Network Based Minutiae Filtering in Fingerprints*, 1998
- [4] Ryszard Tadeusiewicz, Przemysław Korohoda, *Komputerowa analiza i przetwarzanie obrazów*, Wydawnictwo Fundacji Postępu Telekomunikacji 1997
- [5] Marie Sandström, *Liveness Detection in Fingerprint Recognition Systems*, 2004
- [6] Tsai-Yang Jea, Venu Govindaraju, *A minutia-based partial fingerprint recognition system*
- [7] Yi Chen, Sarat Dass, Arun Ross, Anil Jain, *Fingerprint deformation models using minutiae locations and orientations*, 2005
- [8] Nalini K. Ratha, Andrew Senior, Ruud M. Bolle, *Automated Biometrics*, IBM Thomas J. Watson Research Center
- [9] Czesław Grzeszyk, *Daktyloskopia*, PWN 1992
- [10] International Biometric Group, <http://www.biometricgroup.com/>
- [11] Andrew Hunt, David Thomas, *Pragmatic Unit Testing in Java with JUnit*, Addison-Wesley 2003
- [12] M. J. Donahue, S. I. Rokhlin, *On the use of level curves in image analysis*, 1993

- [13] Richard Dawkins, *Rozplatanie tęczy*, Prószyński i S-ka, Warszawa 2007
- [14] *Encyklopedia fizyki*, PWN 1973
- [15] <http://www.fingerprints.tk>
- [16] <http://ridgesandfurrows.homestead.com/>
- [17] <http://www.cs.helsinki.fi/group/goa/mallinnus/lines/bresenh.html>
- [18] <http://biolab.csr.unibo.it/>
- [19] <http://www.java.sun.com>
- [20] <http://www.eclipse.org>
- [21] <http://www.testng.org>
- [22] <http://xstream.codehaus.org/>
- [23] <http://commons.apache.org/betwixt/>
- [24] <http://logging.apache.org/log4j/>
- [25] <http://www.easymock.org/>
- [26] <http://bias.csr.unibo.it/fvc2004/>
- [27] <http://www.crossmatch.com/>
- [28] <http://www.fbi.gov/hq/cjisd/iafis.htm>
- [29] <http://subclipse.tigris.org>
- [30] <http://checkstyle.sourceforge.net/>

Dodatek A

Obsługa aplikacji

A.1 Uruchamianie

Do uruchomienia aplikacji wymagane jest środowisko uruchomieniowe Java (JRE) w wersji 1.5 zainstalowane w systemie operacyjnym. Stworzona aplikacja dostarczona jest w formie pliku `.jar` - wykonywalnego archiwum Java. Może być uruchamiana zarówno pod systemem Linux, Windows jak i Solaris. Aplikacja do pracy wymaga obecności dodatkowych bibliotek w katalogu `lib/` oraz istnienia dwóch plików konfiguracyjnych (patrz A.2) w katalogu `rc/`. Katalog `dist/` na płycie zawiera aplikację gotową do uruchomienia (szczegółowy opis zawartości płyty znajduje się w Dodatku B).

Aplikacja jest programem konsolowym. Sterowana jest za pomocą parametrów przekazywanych z linii poleceń. W celu uruchomienia aplikacji należy – będąc w konsoli – wejść do katalogu, w którym znajduje się plik `jar` z aplikacją, a następnie uruchomić go z odpowiednimi parametrami, przykładowo:

```
java -jar fingerprintMatcher.jar -bc fingerprints/templates/synthetic
```

Tryb pracy programu określany jest poprzez użycie odpowiednich opcji. Dostępne możliwości to:

- `-e`, `--extractMinutiae` : program dokona tylko ekstrakcji cech. Wymagane jest podanie co najmniej jednego parametru, określającego położenie pliku graficznego z odciskiem palca. Drugi parametr jest opcjonalny i określa położenie

pliku szablonu, w którym mają zostać zapisane wyekstrahowane cechy. Jeśli drugi parametr nie zostanie podany, wtedy program zapisze szablon w tym samym katalogu co plik graficzny, zmieniając rozszerzenie pliku na xml.

- **-c, --compareTemplates** : program porówna ze sobą dwa szablony z wyekstrahowanymi cechami. Wymagane jest podanie dwóch parametrów określających położenie dwóch plików do porównania.
- **-be, --batchExtract** : wsadowa ekstrakcja cech. Wymagane jest podanie dwóch parametrów. Pierwszy określa katalog, w którym znajdują się pliki graficzne z odciskami palców. Podany katalog nie może być pusty i musi zawierać tylko pliki graficzne z nazwami w formacie: "prefiks_A_B.rozszerzenie". Prefiks jest ignorowany, ale musi istnieć i nie może zawierać podkreślników. A oznacza liczbę, która musi być identyczna dla różnych obrazów tego samego palca. B oznacza kolejne numery różnych ujęć tego samego odcisku palca. Przykładowo nazwy plików w katalogu to: synthetic_1_1.png, synthetic_1_2.png, synthetic_1_3.png, synthetic_2_1.png, synthetic_2_2.png, synthetic_2_3.png. Wtedy pierwsze trzy pliki zawierają trzy różne ujęcia jednego odcisku palca. Kolejne trzy pliki zawierają obrazy drugiego odcisku palca. Drugi podawany parametr musi określać nazwę pustego katalogu, w którym zostaną zapisane szablony.
- **-bc, --batchCompare** : wsadowe porównywanie wzorców. Wymagane jest podanie jednego parametru określającego katalog z szablonami odcisków palców. Nazwy plików muszą być zgodne ze standardem określonym w poprzednim podpunkcie. Program dokonuje obliczenia wskaźników FMR i FNMR zgodnie z metodą opisaną w Rozdziale 5.
- **brak opcji** : w przypadku braku jakiegokolwiek opcji wymagane jest podanie dwóch parametrów określających położenie plików graficznych z odciskami palców do porównania.

A.2 Konfiguracja parametrów

Parametry zastosowanych algorytmów zapisane są w dwóch plikach konfiguracyjnych zapisanych w katalogu w katalogu rc/. Plik `extraction-configuration.xml`

zawiera konfigurację związaną z ekstrakcją cech. Plik `matching-configuration.xml` zawiera konfigurację dopasowywania wzorców. Są to pliki w formacie XML, a więc wartości poszczególnych parametrów zapisane są pomiędzy znacznikami. Wszystkie znaczniki związane z parametrami konfiguracji ujęte są pomiędzy głównym znacznikiem (Rysunek A.1).

```
<extraction.configuration.ExtractionConfiguration>
  <backgroundColor>255</backgroundColor>
  <ridgeColor>0</ridgeColor>
  <cropThreshold>30</cropThreshold>
  ...
</extraction.configuration.ExtractionConfiguration>
```

Rysunek A.1: Przykładowy fragment pliku konfiguracyjnego

Parametry ekstrakcji cech

backgroundColor : odcień (w skali szarości) tła obrazka zawierającego odcisk palca. Przyjmuje wartości od 0 (czarny) do 255 (biały).

backgroundThreshold : zakres tolerancji podczas segmentacji odcisku palca z tła. Musi być nieujemny.

bendingHistorySize : ilość zapamiętywanych poprzednich kierunków poruszania. Wykorzystywane przy wykrywaniu nadmiernego wyginania się linii. Musi być liczbą dodatnią.

bendingThreshold : zakres tolerancji dla wyginania się linii. Wykorzystywane przy wykrywaniu nadmiernego wyginania się linii. Po przekroczeniu tej wartości algorytm uznaje, że linia zbyt gwałtownie zmieniła kierunek. Wartość podana w stopniach. Musi być liczbą nieujemną.

cropPadding : rozmiar obwódki zostawianej wokół odcisku palca podczas przycinania obrazu (przetwarzanie wstępne). Musi być liczbą nieujemną.

cropThreshold : zakres tolerancji jasności podczas przycinania obrazu z odciskiem palca (przetwarzanie wstępne). Musi być liczbą nieujemną.

directionMapStrategy : wybór strategii określania kierunku. Przyjmuje wartość INTERPOLATED_MAP (dla mapy interpolowanej) lub STANDARD_MAP (dla mapy bez interpolacji).

directionThreshold : zakres tolerancji kątowej przy wyznaczaniu nowego maksimum sekcji. Wykorzystywany przy sprawdzaniu warunku stopu. Po przekroczeniu algorytm uznaje, że nastąpiło zakończenie linii. Wartość podana w stopniach. Musi być liczbą nieujemną.

distanceBetweenSections : odległość pomiędzy dwoma kolejnymi sekcjami linii papilarnej. Musi być większe od sectionSize.

extendedSectionSize : parametr wykorzystywany w algorytmie poprawiania jakości sekcji linii papilarnej. Określa ilość dodatkowych sekcji, po każdej stronie aktualnej sekcji, pobieranych z obrazka. Musi być liczbą nieujemną.

gridSize : wielkość siatki stosowanej przy standardowym algorytmie wyznaczania kolejnych punktów startowych w algorytmie śledzenia linii. Musi być większe od 1.

medianMaskSize : rozmiar maski dla filtru medianowego (przetwarzanie wstępne). Musi być liczbą dodatnią.

minimumDistanceBetweenMinutiae : minimalna dopuszczalna odległość między minucjami. Poniżej tej wartości obie minucje zostaną uznane za fałszywe. Musi być liczbą nieujemną.

minimumMinutiaeDistanceFromBorder : minimalna odległość minucji od krawędzi odcisku palca. Poniżej tej wartości minucja zostanie uznana za fałszywą. Musi być liczbą nieujemną.

nextLineProviderStrategy : wybór algorytmu wyznaczania kolejnych punktów początkowych w algorytmie śledzenia linii. Jedyna dopuszczalna wartość to STANDARD_NEXT_LINE_PROVIDER (opcja stworzona w celu późniejszych rozszerzeń).

ridgeColor : odcień (w skali szarości) krawędzi linii papilarnej. Przyjmuje wartości od 0 (czarny) do 255 (biały).

ridgeThreshold : zakres tolerancji dla jasności linii papilarnej. Wykorzystywane przy określaniu warunków stopu. Musi być liczbą nieujemną.

sectionSize : szerokość sekcji pobieranej z linii papilarnej. Rzeczywista długość sekcji wynosi $2 \cdot sectionSize + 1$. Musi być liczbą dodatnią.

smoothingWindowSize : parametr wykorzystywany w algorytmie wygładzania krawędzi obszaru wysegmentowanego z tła. Określa szerokość okna wygładzającego. Musi być liczbą dodatnią.

tangentWindowSize : rozmiar okna wykorzystywanego przy obliczeniu mapy kierunków. Musi być liczbą większą od 1.

useAverageSectionEnhancement : określa, czy stosować poprawę jakości sekcji poprzez uśrednianie z sąsiednimi sekcjami. Przyjmuje **true** albo **false**.

useCropImagePreprocessing : określa, czy stosować przetwarzanie wstępne obrazu w postaci jego przycięcia. Przyjmuje **true** albo **false**.

useExcessiveBending : określa, czy korzystać z warunku stopu sprawdzającego nadmierne wyginanie się linii papilarnej. Przyjmuje **true** albo **false**.

useGaussianSectionEnhancement : określa, czy stosować poprawę jakości sekcji poprzez splot z maską m_k . Przyjmuje **true** albo **false**.

useImprovedNextLineProvider : określa, czy uwzględniać jasność piksela przy wyznaczaniu nowego punktu początkowego. Opcja ta jest niezależna od strategii wykorzystywanej do wyznaczania kolejnych nowych punktów (opcja `nextLineProviderStrategy`). Przyjmuje **true** albo **false**.

useImprovedStopCriteria : określa czy korzystać z ulepszonych kryterium wykrywania zakończenia linii (opisanego w Rozdziale 3.2.4) czy z oryginalnego kryterium stopu (opisanego w Rozdziale 2.3.2). Przyjmuje **true** albo **false**.

useMedianImagePreprocessing : określa, czy stosować przetwarzanie wstępne obrazu w postaci filtru medianowego. Przyjmuje **true** albo **false**.

Parametry dopasowywania wzorców

alignmentCalculatorType : określa rodzaj algorytmu głosowania w procesie wyznaczania transformacji geometrycznej. Algorytm, który oddaje tylko jeden głos w każdej iteracji oznaczono jako STANDARD_ALIGNMENT_CALCULATOR. Wartość IMPROVED_ALIGNMENT_CALCULATOR oznacza stosowanie ulepszonego algorytmu opisanego w Rozdziale 3.3.1.

matchingDirectionThreshold : zakres tolerancji różnicy kierunków minucji. Używane zarówno przy wyznaczaniu transformacji geometrycznej jak i późniejszym porównywaniu wzorców. Musi być liczbą nieujemną.

matchingDistanceThreshold : zakres tolerancji odległości między minucjami. Używane zarówno przy wyznaczaniu transformacji geometrycznej jak i późniejszym porównywaniu wzorców. Musi być liczbą nieujemną.

minimumMatchedMinutiae : ilość minucji jaka musi się pokrywać żeby uznać dwa odciski za identyczne. Musi być liczbą dodatnią.

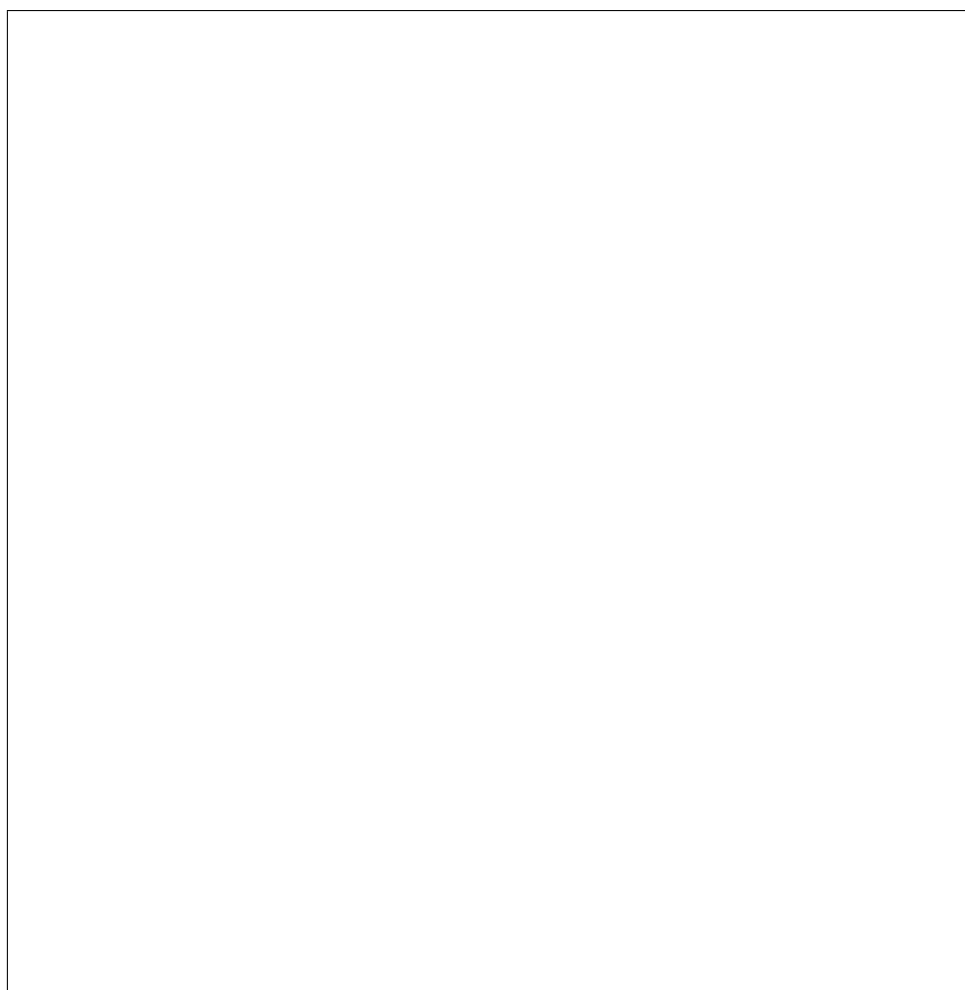
minutiaeMatcherType : wybór algorytmu wyznaczania stopnia podobieństwa odcisków. Jedyna dopuszczalna wartość to NAIVE_MINUTIAE_MATCHER (opcja stworzona w celu późniejszych rozszerzeń).

rotationBinSize : określa, co ile dyskretyzowany jest kąt w procesie wyznaczania optymalnego przekształcenia geometrycznego. Musi zawierać się pomiędzy 4, a 90 włącznie.

translationBinSize : określa, co ile dyskretyzowane jest przesunięcie w procesie wyznaczania optymalnego przekształcenia geometrycznego. Musi być liczbą większą od 3.

Dodatek B

Płyta CD



Zawartość katalogów na płycie:

- db** : bazy danych sztucznych i naturalnych odcisków palców wykorzystywane do badania dostrajania paramterów (podkatalog `tuning/`) i skuteczności działania aplikacji (podkatalog `test/`)
- dist** : dystrybucyjna wersja aplikacji przeznaczona do uruchamiania
- doc** : elektroniczna wersja pracy magisterskiej oraz dwie prezentacje wygłoszone podczas seminarium dyplomowego
- jre** : środowisko uruchomieniowe Java (JRE) potrzebne do uruchomienia aplikacji (wersja dla Windows, Windows 64-bit, Linux i Linux 64-bit)
- src** : kod źródłowy aplikacji (projekt środowiska Eclipse)