

**SZYBKA EKSTRAKCJA CECH OBROTOWO NIEZMIENNICZYCH
ZA POMOCĄ OBRAZÓW CAŁKOWYCH
W ZADANIACH DETEKCJI**

Rozprawa doktorska

Autor:

mgr inż. Aneta Bera

Promotor:

dr hab. inż. Przemysław Klęsk, prof. ZUT

Szczecin, 2021

Streszczenie

Szybka ekstrakcja cech obrotowo niezmienniczych za pomocą obrazów całkowych w zadaniach detekcji

Autor: mgr inż. Aneta Bera

Promotor: dr hab. inż. Przemysław Klęsk, prof. ZUT

Celem niniejszej pracy było opracowanie algorytmu pozwalającego na szybkie ekstrakowanie cech obrotowo niezmienniczych z fragmentów obrazu na potrzeby zadania detekcji obiektów w ramach klasycznej procedury przebiegającej obraz oknem przesuwным. W pracy skupiono się na cechach generowanych na podstawie momentów Zernike'a i przedstawiono propozycję pozwalającą na obliczanie każdej cechy w czasie stałym — $O(1)$ — tj. niezależnym od liczby pikseli w rozważanym fragmencie obrazu. Zaproponowana technika opiera się na przygotowaniu zbioru specjalnych obrazów całkowych o wartościach zespolonych. W przedstawionym podejściu podstawowym powstają błędy numeryczne ze względu na obliczenia na liczbach zmiennoprzecinkowych oraz użyte rozwinięcia dwumianowe. Dlatego też w dalszej części pracy zaproponowano dodatkową technikę podziału obrazów całkowych oraz bezpieczniejszy numerycznie wzór do wyznaczania momentów Zernike'a. Ponadto, opisana została także rozszerzona przestrzeń cech (obrotowych niezmienników) wykorzystująca zarówno moduły momentów Zernike'a jak i ich odpowiednie iloczyny. Praca zawiera dwa eksperymenty z zadaniami detekcji o różnym poziomie trudności. Na etapie uczenia maszynowego, detektory zostały nauczone (z wykorzystaniem zaproponowanych cech w różnych wariantach) za pomocą algorytmu RealBoost.

Słowa kluczowe: zespolone obrazy całkowe, momenty Zernike'a, detekcja, stałoczasowa ekstrakcja cech, obrotowa niezmienniczość

Abstract

Fast extraction of rotationally invariant features backed with integral images in detection tasks

Author: mgr inż. Aneta Bera

Supervisor: dr hab. inż. Przemysław Kłęsk, prof. ZUT

The aim of this doctoral dissertation was to develop an algorithm that is suitable for fast extraction of rotationally invariant features computed from image fragments within a dense detection procedure, in which the image is scanned by a sliding window. This work focuses on features based on Zernike moments and proposes a way to calculate each feature in constant time — $O(1)$ — i.e. the extraction time does not depend on the number of pixels in the considered image fragment. The proposed technique is based on a set of special complex-valued integral images. In the basic variant of the presented approach, numerical errors can occur due to calculations on floating point numbers and due to applied binomial expansions. Hence, in further parts of the dissertation, an enhanced variant is proposed, which applies piecewise integral images and a safer formula for calculating Zernike moments. Moreover, an extended space of features (rotational invariants) is presented. It uses both moduli and suitable products of Zernike moments. This work includes two experiments with detection tasks of different difficulty level. In the machine learning phase, detectors were trained (using proposed features in various variants) by the RealBoost algorithm.

Keywords: complex-valued integral images, Zernike moments, detection, constant-time feature extraction, rotational invariance

Spis treści

Wprowadzenie	7
Cel i teza pracy	7
1 Wstęp teoretyczny	14
1.1 Zadanie detekcji i procedura detekcyjna	14
1.2 Podstawowy obraz całkowy	16
1.3 Boosting	17
1.3.1 Discrete AdaBoost	17
1.3.2 RealBoost	18
1.4 Słabe klasyfikatory	19
1.4.1 Drzewo decyzyjne	19
1.4.2 Kosze z odpowiedzią rzeczywistoliczbową	21
1.4.3 Zastosowanie koszy w drzewach	22
2 Momenty Zernike’a	23
2.1 Wielomiany i momenty Zernike’a	23
2.2 Niezmienniczość ze względu na obrót i skalowanie	25
2.3 Aproksymacja funkcji poprzez częściową sumę rozwinięcia	26
2.4 Związek momentów Zernike’a z ortogonalnymi momentami Fouriera–Mellina	27
2.5 Porównanie momentów Zernike’a z ortogonalnymi momentami Fouriera–Mellina	28
2.5.1 Bazy ortogonalne i rozkład zer w wielomianach radialnych . .	28
2.5.2 Tłumienie informacji	32
2.6 Momenty Zernike’a w literaturze	34
3 Stałoczasowe obliczanie momentów Zernike’a z wykorzystaniem ob- razów całkowych	36
3.1 Przekształcenia momentów Zernike’a służące wprowadzeniu obrazów całkowych	36

3.1.1	Odwrócona kolejność sumowania	36
3.1.2	Reprezentacja kartezjańska	37
3.1.3	Wyznaczanie momentów Zernike'a dla fragmentu obrazu	38
3.2	Zespolone obrazy całkowe i stałoczasowe obliczanie momentów Zernike'a podczas procedury detekcyjnej	40
3.3	Własności	43
3.3.1	Zespolone sprzężenie obrazów całkowych	43
3.3.2	Dokładna liczba potrzebnych obrazów całkowych	44
3.3.3	Liczba operacji przyrostu	46
3.3.4	Liczba cech	47
3.4	Wariant pomysłu podstawowego	48
3.5	Przyspieszenie obliczeń poprzez tablicowanie	51
3.5.1	Tablica dla wyrażeń zawierających indeks centralny	51
3.5.2	Tablica na wyniki zwracane przez operator przyrostu	51
3.5.3	Algorytm ekstrakcji cech wykorzystujący tablicowanie	52
4	Rezultaty pierwszych eksperymentów	55
4.1	Zbiór 1 — litera A	56
4.2	Zbiór 2 — samoloty	60
5	Redukcja błędów numerycznych	63
5.1	Błędy numeryczne	63
5.2	Podział obrazów całkowych	64
5.2.1	Numerycznie bezpieczny wzór dla momentów Zernike'a	67
5.2.2	Rzędy wielkości składników biorących udział w obliczaniu momentów Zernike'a z wykorzystaniem obrazów całkowych — przykład	72
5.3	Dodatkowe operacje występujące w przypadku numerycznie bezpiecznego podejścia	73
6	Rozszerzona przestrzeń niezmienników Zernike'a	76
6.1	Grupy iloczynów i schemat indeksowania	78
6.2	Procedura ekstrakcji cech	81
7	Eksperymenty	84
7.1	Zbiór 1 — litera A	84
7.2	Zbiór 2 — samoloty	87
7.3	Porównanie wydajności czasowych	92

8 Podsumowanie pracy	95
8.1 Dalsze kierunki badań	96
Spis rysunków	98
Spis tablic	98
Spis algorytmów	99
Bibliografia	101

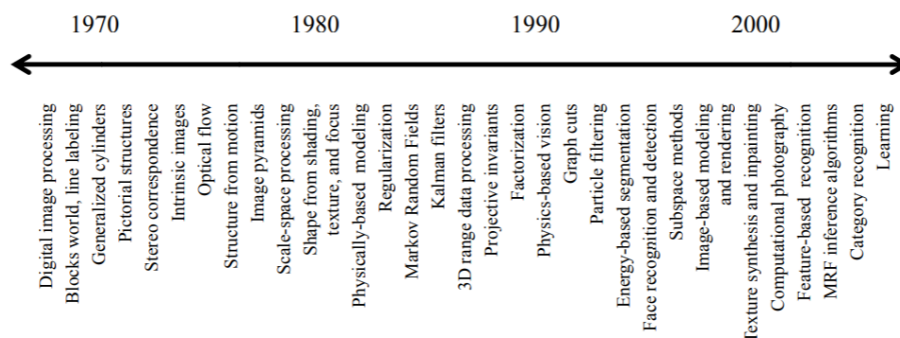
Wprowadzenie

Uczenie maszynowe to nauka o algorytmach komputerowych, które, wykorzystując nabyte doświadczenie (informacje z przeszłości dostarczone do nauki w formie przykładów), potrafią udoskonalać model pewnego zjawiska i dokonywać trafnych predykcji. Rozmiar i jakość danych wykorzystywanych do nauki mają istotny wpływ na poprawność późniejszych predykcji. Kluczowymi elementami, które określają jakość tych algorytmów są złożoność czasowa i pamięciowa oraz dodatkowo ilość danych potrzebna, aby algorytm został prawidłowo nauczony. Algorytmy uczenia są stosowane od wielu lat w wielu zastosowaniach np.: klasyfikacji, regresji, tworzeniu rankingów, grupowaniu, czy redukcji wymiarowości (Mohri, Rostamizadeh i Talwalkar, 2012).

W literaturze można wskazać wiele zastosowań uczenia maszynowego. Na przykład w zagadnieniu cyberprzestępczości, Pu i in. (2021) zaprezentowali metodę bazującą na połączeniu algorytmu grupowania podprzestrzeni (ang. *Sub-Space Clustering*) oraz odmiany metody wektorów nośnych (ang. *One-class Support Vector Machines*), która jest przydatna w momencie, gdy dostępne są jedynie dane z jednej klasy. Na tej podstawie autorzy starają się wykryć atak bez żadnej wcześniejszej wiedzy osiągając zadowalające rezultaty. Meena, Sharma i Mahrishi (2020) przedstawili narzędzie, bazujące na algorytmach uczenia maszynowego, do predykcji dla inteligentnego systemu zarządzania ruchem, tak aby uczestnicy ruchu mogli podejmować świadome decyzje o trasie przejazdu, omijając np. korki czy wypadki. C.-J. H. Huang i in. (2005) używając algorytmów uczenia maszynowego takich jak SVM, algorytm k -najbliższych sąsiadów, czy naiwny klasyfikator Bayesa starają się wskazywać użytkownikowi przeglądarki internetowej „wyszukiwania”, które najlepiej będą odpowiadać jego zainteresowaniom. Pahwa i Agarwal (2019) zaproponowali, by użyć algorytmów uczenia maszynowego do predykcji przyszłych cen akcji na giełdzie. Ross i in. (2013) przedstawiają sposób pozwalający na określenie, czy studenci uczestniczący w zajęciach uważają na nich. Autorzy wykorzystują algorytm K -średnich oraz metodę wektorów nośnych do automatycznej klasyfikacji uwagi studentów, bazując na kamerze RGB z sensorem głębi.

Algorytmy uczenia są także z powodzeniem stosowane m.in. w zadaniach widzenia komputerowego (ang. *computer vision*). Widzenie komputerowe jest dziedziną naukową, która ma na celu rozwijać techniki, które pozwoliłyby komputerom „zobaczyć” i zrozumieć zawartość obrazów cyfrowych (mogą to być obrazy lub filmy). Na pierwszy rzut oka rozwiązywany problem wydaje się być bardzo prosty, gdyż człowiek bez problemu rozpoznaje twarze, rozróżnia pojazdy, dostrzega trójwymiarowe struktury świata wokół nas. Jesteśmy w stanie określić np. jaki jest kształt obiektów na obrazie, ile osób znajduje się na fotografii, czy też jakie są ich emocje. Problem zaczyna pojawiać się w momencie, gdy chce się nauczyć komputer robić to samo, gdyż nie do końca rozumiany jest mechanizm widzenia u człowieka i złożoność samej percepcji (Szeliski, 2011). Przykładowym zadaniem, które zostało już dobrze rozwiązane, jest np. automatyczne rozpoznawanie tablic rejestracyjnych samochodów na sekwencjach wideo. Nawet tak z pozoru proste zadanie, które człowiek rozwiązuje bez zastanowienia, nie jest zadaniem jednoetapowym. Można tu wskazać kilka problemów: pozyskanie obrazu, wykrycie pojazdu (pojazdów) na obrazie, wyodrębnienie fragmentu z pojazdem, wyodrębnienie tablicy rejestracyjnej, segmentacja znaków i dopiero na końcu rozpoznanie ich (Qadri i Asif, 2009).

Od wielu lat naukowcy poszukują technik matematycznych, które pozwoliłyby na pozyskiwanie potrzebnych informacji o obiektach na fotografiach czy filmach. Rys. 1 przedstawia główne osiągnięcia w dziedzinie widzenia komputerowego na przestrzeni lat. Jak można zauważyć, poczynając od lat 70-tych, pojawiały się coraz to bardziej złożone problemy oraz metody rozwiązywania ich (Huffman, 1971).



Rysunek 1: Główne osiągnięcia w dziedzinie widzenia komputerowego na przestrzeni lat. (Źródło: Szeliski (2011))

Część zadań powiązanych ze stosowaniem algorytmów uczenia maszynowego oraz technik widzenia komputerowego dotyczy obrazów cyfrowych i ich przetwarzania. Na przestrzeni wielu lat naukowcy proponowali metody, dzięki którym możliwe byłoby wykrycie pewnych obiektów na obrazach. Karimunnisa i in. (2020) wykrywali złamanie kości na obrazach RTG, wykorzystując cechy powiązane z kształtem

i orientacją, obliczane po uprzednim wstępnym przetworzeniu obrazów: filtrowaniu i detekcji krawędzi. Wykorzystując sieć neuronową jako klasyfikator, autorom udało się uzyskać zadowalające rezultaty detekcji złamań. Gurbină, M. Lascu i D. Lascu (2019) wykrywali guzy mózgu na podstawie obrazu z rezonansu magnetycznego. W tym celu zastosowali transformacje falkowe oraz maszynę wektorów nośnych.

W ostatnich latach, w związku ze zwiększoną popularnością tematu pojazdów autonomicznych, ważne stały się tematy m.in. rozpoznawania pieszych, czy znaków drogowych na obrazach oraz filmach. Gaddigoudar i in. (2017) wykrywali oraz śledzili ruch pieszych bazując na filtrach cząsteczkowych. Wu, Yang i Zhang (2011) stosując jako cechy deskryptor HOG, a jako algorytm uczący AdaBoost, także poradzili sobie z zadaniem wykrywania pieszych. Hechri i Abdellatif (2012) zaprezentowali metodę pozwalającą na detekcję i rozpoznawanie znaków drogowych. Wykorzystali metody filtrowania kształtów oraz wielowarstwowe sieci neuronowe. Aby wspomniane metody mogły zostać zastosowane w autonomicznych samochodach ważne jest, aby zadanie detekcji było wykonywane szybko, gdyż od szybkości wykrycia np. pieszego, który pojawia się przed maską samochodu, zależy bezpieczeństwo uczestników ruchu drogowego.

Wybór cech (lub inaczej atrybutów czy też zmiennych wejściowych) ma kluczowe znaczenie w wielu zadaniach przetwarzania obrazów. Pomimo podobieństw koncepcyjnych, wyniki zadań rozpoznawania i detekcji (czyli wykrywania obiektów na obrazie) mogą być bardzo różne w zależności od wybranej przestrzeni cech. Klasyczna procedura detekcyjna polega na przebieganiu obrazu tzw. *oknem przesuwnym* (ang. *sliding window*). Procedura ta jest bardzo wymagająca obliczeniowo. Szukając interesujących nas obiektów, należy przejść cały obraz wejściowy (lub klatkę wideo) oknem przesuwnym, najczęściej występującym w wielu skalach. Prowadzi to do analizy dużej liczby okien (fragmentów obrazu), może to być liczba rzędu 10^5 lub nawet 10^6 okien na obraz. W związku z tym, oczywiste jest, że cechy ekstrahowane z pikseli każdego okna nie mogą być zbyt wymagające obliczeniowo, tak aby cała procedura mogła się odbyć w rozsądnym czasie. Prowadzi to więc często do wyboru prostszych cech. Z drugiej strony, w zadaniach rozpoznawania, często wybierane są te bardziej skomplikowane i kosztowne obliczeniowo cechy. Jest to możliwe, ponieważ zadanie to zakłada, że określony fragment obrazu został wcześniej przygotowany (np. wycięty) i w związku z tym nie ma się do czynienia z procedurą detekcyjną.

W ostatnich czasach, na popularności w wielu systemach detekcji zyskały cechy Haara, które można wyznaczać szybciej dzięki *obrazowi całkowemu* (ang. *integral image*) (Crow, 1984; Lewis, 1995; Papageorgiou, Oren i Poggio, 1998; Viola i Jones, 2001). Zostały one wykorzystane m.in. do detekcji twarzy, dłoni, pojazdów, znaków

drogowych, pieszych itd. Swoją popularność zawdzięczają one prostej technice ich wyznaczania jako różnic średnich jasności pikseli pomiędzy obszarami określonymi przez pewne szablony. Mają jednak jedną podstawową wadę — nie są obrotowo niezmiennicze. Mimo, że znaki drogowe czy piesi, w większości przypadków, występują na obrazach w swojej naturalnej pionowej pozycji, nie musi już tak być w przypadku np. dłoni czy twarzy.

W związku z tym, gdy zadanie wymaga wykrywania obiektów w dowolnym obrocie, można spotkać dwa podejścia. Pierwsze z nich polega na przygotowywaniu wielu detektorów, każdy odpowiedzialny za inny zakres kąta, np. 16 detektorów, każdy dla wycinka kąтового o szerokości 22.5° . W takim przypadku, konieczne jest wielokrotne skanowanie obrazu oknem przesuwным. Drugie podejście zakłada przygotowanie jednego detektora, który jest uczony na przykładach z obiektami występującymi w różnych obrotach, nie przejmując się brakiem obrotowej niezmienniczości. Nauczony tak detektor jest jednak narażony na istotne zwiększenie liczby fałszywych alarmów.

Zaproponowane cechy Haara okazały się być (na tamte lata) cechami, które można wyznaczyć w szybkim czasie. Popularyzatorzy pomysłu — Viola i Jones (2001) — wskazywali, że ich rozwiązanie było 15 razy szybsze od tego opisanego w pracy (Schneiderman i Kanade, 2000), gdzie bazowano na histogramach i współczynnikach falkowych w celu opisu obiektów, oraz aż 600 razy szybsze od (Rowley, Baluja i Kanade, 1998), gdzie wykorzystano sieć neuronową oraz algorytm filtrujący. Podobnie jak Viola i Jones (2001), autorzy wspomnianych pracy skupiali się na zagadnieniu detekcji twarzy.

Warto wspomnieć o innym podejściu do zadania detekcji niż zastosowanie okna przesuwного (jak np. w przypadku cech Haara), jakie reprezentuje transformacja Hougha (Hough, 1962), która początkowo była używana do detekcji prostych kształtów geometrycznych. Przykładowo, wykrywanie linii bazuje na głosowaniu punktów charakterystycznych, które mogą zostać wykryte np. poprzez wykrywanie krawędzi. Każdy taki punkt głosuje na jedną z prostych. Każda prosta (możliwa do utworzenia) może zostać jednoznacznie opisana przez kąt nachylenia do układu współrzędnych oraz odległości od jego początku. Dzięki temu można utworzyć nowy układ współrzędnych, w którym każdy punkt będzie reprezentować inną prostą i to właśnie na nie głosują piksele charakterystyczne, zwiększając wartość przypisaną prostym, które przechodzą przez ten piksel. Wartości te można utożsamić z jasnością pikseli. Wtedy, docelowe wykryte proste reprezentowane są przez najjaśniejsze piksele. Jak widać, w podejściu tym nie odnajdzie się typowej procedury detekcyjnej powiązanej z oknem przesuwным. Kilka lat później transformacja Hougha została wykorzystana

do wykrywania okręgów (Duda i Hart, 1972), a docelowo uogólniona do wykrywania dowolnych kształtów (Ballard, 1981).

Kolejnym podejściem do zadań detekcji, które nie korzysta z okna przesuwne jest YOLO (ang. *You Only Look Once*), czyli rewolucyjny system do wykrywania obiektów w czasie rzeczywistym bazujący na regresji (Redmon i Farhadi, 2017). Autorzy odchodzą od klasycznego podejścia związanego z oknem przesuwnym. Obraz dzielony jest na siatkę 19×19 tzw. komórek. Dla każdej komórki stosowana jest sieć neuronowa, która przewiduje (poprzez regresję) współrzędne obwiedni obiektów oraz prawdopodobieństwa poszczególnych klas obiektów. Obliczenia są zatem wykonywane tylko np. dla $361 = 19 \cdot 19$ okien obrazu zamiast dla liczby okien rzędu $10^{5\pm 1}$ w klasycznym podejściu. W 2018 roku autorzy zaproponowali aktualizację pomysłu, która charakteryzuje się m.in. propozycją większej sieci. Podejście to pozwoliło dodatkowo zwiększyć dokładność detekcji, pozostając przy tym nadal bardzo szybkim rozwiązaniem (Redmon i Farhadi, 2018). Ze względu na fakt, że podejście YOLO pojawiło się w momencie, gdy prace nad niniejszą rozprawą doktorską były już mocno zaawansowane, nie będzie ono już dalej w pracy analizowane, mimo że przypuszczalnie zwiastuje ono na przyszłość stopniowe odchodzenie od klasycznych procedur detekcyjnych z oknem przesuwnym.

Istnieją pewne rodziny momentów matematycznych, na bazie których można generować niezmienniki tzn. cechy obrotowo niezmiennicze. Są to np.: momenty Zernike'a (Zernike, 1934), ortogonalne momenty Fouriera–Mellina (Sheng i Shen, 1994), momenty geometryczne (Hu, 1962), deskryptor SIFT czy SURF (Juan i Gwun, 2009). Dla uściślenia, to moduły zespolone momentów Zernike'a i ortogonalnych momentów Fouriera–Mellina są niezmiennie ze względu na obrót, a nie same momenty. Wszystkie wymienione przykłady są w sensie obliczeniowym bardziej wymagające od cech Haara. Ich bezpośrednie zastosowanie w procedurze detekcyjnej jest niemożliwe, tzn. wyznaczenie takich cech z definicji dla każdej pozycji okna przesuwne nie jest możliwe do zrealizowania w rozsądnym czasie. W rzeczywistości deskryptory SIFT i SURF są powszechnie stosowane w zadaniu rozpoznawania obrazów, czy szukania punktów charakterystycznych, a nie w zadaniach detekcji. Ich obliczanie jest procesem wieloetapowym i zawiera następujące operacje: różnica filtrów Gaussa (DoG), indeksowanie kluczowych lokalizacji, indeksowanie klastrów z wykorzystaniem transformacji Hougha, weryfikacja metodą najmniejszych kwadratów (LSQ) i przypisanie orientacji. Ostatni krok dotyczy obrotowej niezmienniczości. Obejmuje on obliczenia na współrzędnych biegunowych powtarzanych dla każdego piksela w sąsiedztwie kluczowego punktu. Ich złożoność wyklucza możliwość zastosowania ich do ekstrakcji cech w ramach klasycznej procedury detekcyjnej. W szcze-

gólności, wiele momentów (Mukundan i Ramakrishnan, 1998), często używanych w zadaniach rozpoznawania, jest wykluczana z zadań detekcji ze względu na to, że czas ich wyznaczania jest liniowo zależny od liczby pikseli. Podobnie, podejścia oparte na głębokim uczeniu (ang. *deep learning*) nie mogą być bezpośrednio zastosowane do detekcji i wymagają dodatkowych etapów przed ich wykorzystaniem — stosowane są sieci oparte na proponowaniu regionów (ang. *region proposal*), co pozwala na wykrywanie granic obiektów (S. Ren i in., 2017).

Istnieje kilka przestrzeni cech (czy deskryptorów), przy których można uniknąć wspomnianych trudności, ze względu na pojawienie się w ostatnich dwóch dekadach technik pozwalających na ich szybkie (stałoczasowe) wyznaczanie. Cechy Haara (HF) (Viola i Jones, 2001), lokalne wzorce binarne (ang. *local binary patterns*, w skrócie LBP) (Acasandrei i Barriga, 2014), histogram zorientowanych gradientów (ang. *Histogram of Oriented Gradients*, w skrócie HOG) (Dalal i Triggs, 2005), czy momenty Fouriera (FM) (Kłęsk, 2017) są tego przykładem. Kluczowym trikiem pozwalających na ich stałoczasowe wyznaczanie są *obrazy całkowite*. Są to pomocnicze tablice przechowujące skumulowane jasności pikseli lub inne wyrażenia związane z pikselami. Mając je przygotowane przed skanowaniem obrazu oknem przesuwным, można szybko obliczać potrzebne sumy dzięki tzw. operatorowi przyrostu. Każdy przyrost wymaga dwóch dodawań i jednego odejmowania wykorzystując cztery wartości odczytane z obrazu całkowitego.

W literaturze można odnaleźć kilka pomysłów poprawiających wydajność procedury detekcyjnej np.: niezależne od skali rozszerzenie cech HOG (ang. *single-scale HOG*) w pracy (Costea i Nedevschi, 2014), znormalizowane różnice pikseli (ang. *normalized pixel differences*) w artykule (Liao, Jain i S. Z. Li, 2016), czy połączenie cech Haara, LBP oraz cech HOG w pracy (H. Ren i Z. Li, 2015). Jednak wszystkie te pomysły są wciąż wspomagane przez podstawowe obrazy całkowite.

Niniejsza praca skupia się na momentach Zernike’a i możliwości przyspieszenia wyznaczania ich. Zasadniczy pomysł przedstawiony w pracy polega na przygotowaniu zestawu specjalnych zespolonych obrazów całkowitych oraz użyciu algorytmu, który pozwala na wyznaczanie momentów Zernike’a w stałym czasie, niezależnie od położenia i liczby pikseli w oknie. Dzięki temu, momenty Zernike’a mogą zostać zastosowane jako cechy w zadaniach detekcji, gdzie mamy do czynienia z procedurą detekcyjną opartą na oknie przesuwным, które może występować w wielu rozmiarach.

Pierwszy rozdział pracy zawiera wstęp teoretyczny, w którym opisano krótko podstawowe zagadnienia, takie jak: procedura detekcyjna, obrazy całkowite czy bo-

ostnię. W kolejnym rozdziale przedstawione zostały momenty Zernike’a oraz ich własności. Rozdział trzeci jest najważniejszą częścią pracy i opisuje sposób szybkiego wyznaczania momentów Zernike’a przy pomocy zaproponowanych obrazów całkowych. Dodatkowo opisany został wariant podstawowego pomysłu, dzięki któremu można zwiększyć liczbę cech. Kolejny rozdział zawiera opis przeprowadzonych wstępnych eksperymentów na dwóch zbiorach danych. Kolejne dwa rozdziały opisują techniki, których celem jest poprawa jakości detekcji. Pierwszy z nich opisuje metodę podziałów obrazów całkowych, która redukuje błędy numeryczne powstałe przy użyciu zaproponowanego w rozdziale trzecim algorytmu. Drugi natomiast, opisuje sposób na znaczne zwiększenie liczby cech bazujących na odpowiednich iloczynach momentów Zernike’a. Rozdział siódmy przedstawia pełne eksperymenty, które pozwalają porównać zaproponowane warianty algorytmu. Na koniec, ósmy rozdział zawiera podsumowanie pracy.

Cel i teza pracy

Celem pracy jest opracowanie algorytmu umożliwiającego szybkie (stałoczasowe) obliczanie momentów Zernike’a w ramach procedury detekcyjnej, uniezależniając się od pozycji okna i liczby pikseli w oknie.

Motywacja:

Momenty Zernika są zdefiniowane oryginalnie w układzie biegunowym. Pracownik badawczym przyświecało pytanie — czy możliwe jest skonstruowanie odpowiedniego zestawu obrazów całkowych w *jednym* globalnym (dla całego obrazu) układzie współrzędnych kartezjańskich, które pozwoli na późniejsze obliczanie momentów Zernike’a, zdefiniowanych w wielu lokalnych układach współrzędnych biegunowych, uniezależniając się od pozycji poszczególnych okien oraz liczby pikseli w nich zawartych.

Teza badawcza:

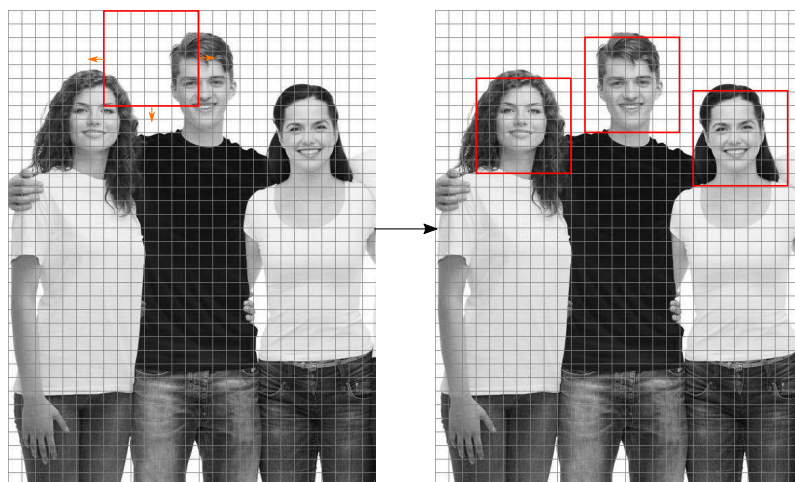
Po przygotowaniu odpowiedniego zestawu zespolonych obrazów całkowych przed procedurą detekcyjną, momenty Zernike’a dla każdego kwadratowego okna obrazu mogą zostać wyznaczone w czasie stałym — $O(1)$ — niezależnie od liczby pikseli w oknie.

Rozdział 1

Wstęp teoretyczny

1.1 Zadanie detekcji i procedura detekcyjna

Zadanie detekcji obiektów na obrazie polega na odnalezieniu poszukiwanych obiektów. Mogą być to twarze (jak na Rys. 1.1), znaki drogowe, itd. Taki proces wymaga przejrzenia i wykonania obliczeń na ogromnej liczbie okien, które dodatkowo występują w różnych rozmiarach, tak aby możliwe było wykrycie obiektów występujących na obrazie w różnych skalach. Taki proces wiąże się z analizowaniem



Rysunek 1.1: Ilustracja zadania detekcji twarzy na obrazach. (Źródło: opracowanie własne)

liczby okien rzędu 10^5 lub 10^6 . Wybór cech jest więc kluczowy, a liczba analizowanych okien skłania do wybierania tych prostych, których wyznaczenie będzie szybsze i prostsze obliczeniowo. Bardziej skomplikowane cechy są najczęściej wybierane tylko wtedy, gdy są one wyznaczone dla pewnego podzbioru okien uzyskanych po wstępnym przetwarzaniu oryginalnego obrazu. Szkic procedury detekcyjnej został przedstawiony w Algorytmie 1. W algorytmie tym f oznacza obraz wejściowy o roz-

miarach $n_x \times n_y$ (odpowiednio: szerokość \times wysokość), w_{\min} oznacza początkowy rozmiar okna detekcyjnego, s_{\max} oznacza liczbę analizowanych skal okna detekcyjnego, λ oznacza współczynnik przemieszczania okna, a η to współczynnik przyrostu okna.

Algorytm 1 Schemat procedury detekcyjnej.

```

procedure DETECTOBJECT( $f, w_{\min}, s_{\max}, \lambda, \eta$ )
  for  $s := 0, \dots, s_{\max} - 1$  do
    Oblicz rozmiar okna  $w := \text{round}(\eta^s w_{\min})$ .
    Oblicz wielkości przeskoków okna w poziomie i pionie:
       $d_k := \text{round}(\lambda \cdot w), d_j := \text{round}(\lambda \cdot w)$ .
     $k := 0$ 
    while  $k + w - 1 \leq n_x - 1$  do
       $j := 0$ 
      while  $j + w - 1 \leq n_y - 1$  do
        Wyekstrahuj wektor cech  $\phi$  dla okna o rozmiarze  $w \times w$ 
        zaczepionego w punkcie  $(j, k)$ .
        Oblicz odpowiedź klasyfikatora  $H(\phi)$ .
        if  $H(\phi) = +$  then
          Zapamiętaj badane okno jako zawierające poszukiwany obiekt.
        end if
         $j := j + d_j$ 
      end while
       $k := k + d_k$ 
    end while
  end for
  Zwróć wykryte okna pozytywne.
end procedure

```

Dla przykładu, jeśli założymy, że analizowany będzie obraz o wymiarze 640×480 z wykorzystaniem okna przesuwonego w ośmiu skalach, poczynając od okna 48×48 i zwiększając je o 20% ($\eta = 1.2$), stosując dokładne skanowanie obrazu ($d_k = d_j = 1$), konieczne będzie przeanalizowanie łącznie 1 669 456 pozycji okna. Jeśli natomiast zdecydujemy się na pominięcie połowy okien w każdym wymiarze (tj. $d_k = d_j = 2$), przetwarzanych okien będzie już prawie cztery razy mniej, dokładnie 418 298.

Niniejsza praca skupia się na cechach obrotowo niezmienniczych i to właśnie takie cechy będą wyznaczone w procedurze detekcyjnej. Cechami obrotowo niezmienniczymi nazywa się cechy, których wartości pozostają niezmiennie niezależnie od tego,

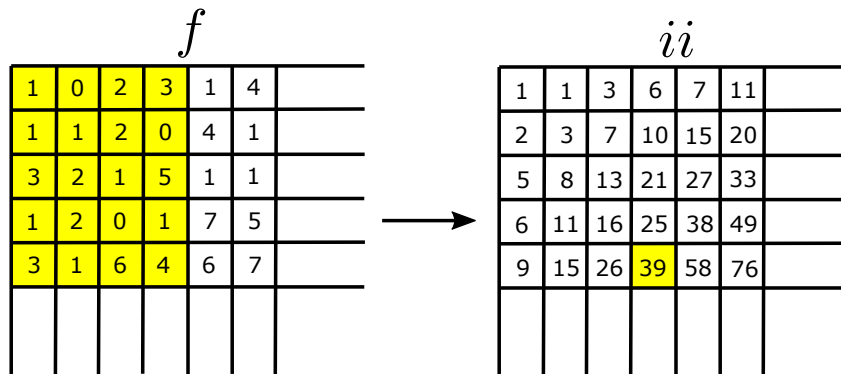
w jakim obrocie występuje obiekt w oknie.

1.2 Podstawowy obraz całkowy

Od tej pory w tekście pracy f będzie używane jako oznaczenie funkcji obrazu, zaś ii (od ang. *integral image*) jako oznaczenie obrazów całkowych (różnego typu). Obraz całkowy jest tego samego rozmiaru co obraz oryginalny (wejściowy), a w wariancie podstawowym jego wartość w punkcie (j, k) to suma jasności pikseli od początku obrazu aż do danego punktu:

$$ii(j, k) = \sum_{1 \leq l \leq j} \sum_{1 \leq m \leq k} f(l, m). \quad (1.1)$$

Rys. 1.2 ilustruje sposób tworzenia obrazu całkowego.



Rysunek 1.2: Wyznaczanie obrazu całkowego ii dla fragmentu obrazu f . (Źródło: opracowanie własne)

Nawiązując do popularnych w ostatnich latach cech Haara, warto zaznaczyć, że ich szybkie wyznaczanie nie jest powiązane bezpośrednio z samą naturą tych cech¹, a właśnie z obrazem całkowym (Crow, 1984; Viola i Jones, 2001).

Obraz całkowy może zostać wyznaczony podczas jednego przejścia przez obraz, wykorzystując wzory rekurencyjne:

$$s(j, k) = s(j, k - 1) + f(j, k), \quad (1.2)$$

$$ii(j, k) = ii(j - 1, k) + s(j, k), \quad (1.3)$$

gdzie $s(j, k)$ to skumulowana suma w wierszu, $s(j, -1) = 0$, a $ii(-1, k) = 0$ (Viola i Jones, 2001).

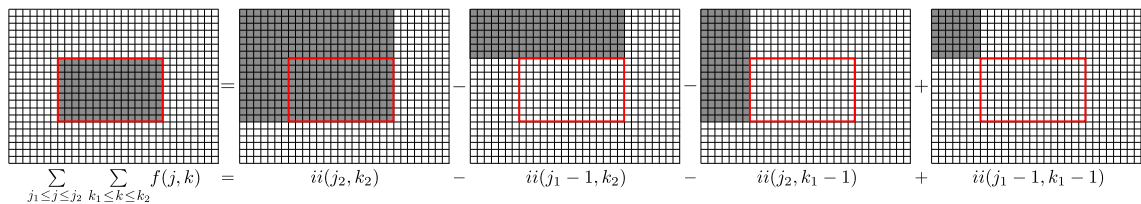
¹Cechy Haara to różnice między średnimi jasnościami pikseli wewnątrz prostokątnych fragmentów obrazów. Np. dla twarzy: linia oczu kontra czoło, linia nosa kontra policzki, itd.

Istnieje kilka modyfikacji podstawowego pomysłu. Np. obraz całkowy kwadratów $ii(j, k) = \sum_{1 \leq l \leq j} \sum_{1 \leq m \leq k} f^2(l, m)$ jest przydatny przy wyznaczaniu wariancji. Z kolei kumulanty tak zwanych macierzy głośów (Said, Atri i Tourki, 2011) pozwalają na szybką ekstrakcję cech deskryptora HOG (Dalal i Triggs, 2005; C. Huang i J. Huang, 2017). Przy pomocy odpowiednio skonstruowanych obrazów całkowych można także przyspieszyć wyznaczanie momentów Fouriera (Kłęsk, 2017). Inne propozycje wykorzystania obrazów całkowych występują w literaturze raczej rzadko.

Wracając do cech Haara, gdy taki obraz całkowy zostanie przygotowany, wtedy suma jasności z dowolnego okna obrazu może zostać wyznaczona w stałym czasie — $O(1)$ — dzięki operatorowi przyrostu obrazu całkowego, który pozwala właśnie na obliczenie sumy jasności pikseli na dowolnym obszarze prostokątnym rozciągniętym od punktu (j_1, k_1) do (j_2, k_2) . Wyznaczenie przyrostu, na podstawie posiadanego obrazu całkowego, wymaga trzech operacji: dwóch odejmowań i jednego dodawania:

$$\sum_{j_1 \leq j \leq j_2} \sum_{k_1 \leq k \leq k_2} f(j, k) = ii(j_2, k_2) - ii(j_1 - 1, k_2) - ii(j_2, k_1 - 1) + ii(j_1 - 1, k_1 - 1). \quad (1.4)$$

Ilustracja działania operatora przyrostu została przedstawiona na Rys. 1.3.



Rysunek 1.3: Wyznaczanie przyrostu obrazu całkowego. (Źródło: opracowanie własne)

1.3 Boosting

Boosting to podejście, które bazuje na połączeniu działania wielu „słabych” klasyfikatorów. Po raz pierwszy pomysł został przedstawiony w latach 90-tych XX wieku (Schapire, 1990). Boosting stosuje ten sam algorytm uczący wielokrotnie na reważonym zbiorze danych, celem poprawy dokładności tego algorytmu. Poniżej opisano dwa podstawowe algorytmy boostingowe: Discrete AdaBoost oraz Real AdaBoost. Drugi z algorytmów został wykorzystany w niniejszej pracy.

1.3.1 Discrete AdaBoost

Algorytm Discrete AdaBoost przyjmuje na wejściu zestaw, który stanowi zbiór uczący w postaci $(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_m, y_m)$, gdzie m to liczba przykładów w zbiorze uczącym. Każdy przykład uczący (\mathbf{x}_i, y_i) otrzymuje pewną wagę w_i . Początkowo

wszystkie wagi mają przypisywaną taką samą wartość, która jest później korygowana w kolejnych etapach w taki sposób, że wagi przykładów błędnie sklasyfikowanych są zwiększane (kładąc tym samym większy nacisk na nie). W algorytmie tym, każdy ze słabych klasyfikatorów zwraca jedną z dwóch odpowiedzi $\{-1, 1\}$. AdaBoost wywołuje po kolei każdy klasyfikator h_t w serii etapów dla $t = 1, \dots, T$. Jakość danego klasyfikatora jest określana poprzez błąd klasyfikacji ϵ_t :

$$\epsilon_t = \sum_{i=1}^m w_i [h_t(\mathbf{x}_i) \neq y_i], \quad (1.5)$$

gdzie $[\dots]$ to funkcja wskaźnikowa, która zwraca 1, gdy postawiony warunek jest spełniony, a 0 w przeciwnym razie. Od błędu zależy siła wpływu tego klasyfikatora na wynik klasyfikacji $\psi_t = \frac{1}{2} \ln \frac{1-\epsilon_t}{\epsilon_t}$. Warto zauważyć, że im błąd jest bliższy wartości $\frac{1}{2}$, tym mniejszy jest wpływ słabego klasyfikatora na zespół (Freund i Schapire, 1999). Poniżej przedstawiono pseudokod algorytmu uczącego AdaBoost.

Algorytm 2 Algorytm uczący klasyfikatora Discrete AdaBoost.

procedure ADABOOSTTRAIN((\mathbf{x}_i, y_i)) $\triangleright i = 1, \dots, m, y_i \in \{-1, +1\}$

Zainicjalizuj wszystkie wagi taką samą wartością: $w_i := \frac{1}{m}$, gdzie $i = 1, \dots, m$.

for $t := 1, \dots, T$ **do**

Naucz słaby klasyfikator $h_t(\mathbf{x}) \in \{-1, 1\}$, wykorzystując wagi w_1, \dots, w_m .

Oblicz błąd klasyfikacji $\epsilon_t := \sum_{i=1}^m w_i [h_t(\mathbf{x}_i) \neq y_i]$.

Oblicz wskaźnik siły wpływu klasyfikatora $\psi_t := \frac{1}{2} \ln \frac{1-\epsilon_t}{\epsilon_t}$.

Zaktualizuj wagi:

$Z_t := \sum_{i=1}^m w_i$ (współczynnik normalizacji wag),

$w_i := \frac{1}{Z_t} w_i e^{(-\psi_t y_i h_t(\mathbf{x}_i))}, \quad i = 1, \dots, m.$

end for

Zwróć zespół klasyfikatorów: h_1, \dots, h_T .

end procedure

Ostateczną odpowiedź klasyfikatora wyznacza się na podstawie sumy ważonych odpowiedzi słabych klasyfikatorów:

$$H(\mathbf{x}_i) = \text{sgn}\left(\sum_{t=1}^T \psi_t h_t(\mathbf{x}_i)\right). \quad (1.6)$$

1.3.2 RealBoost

W przeciwieństwie do Discrete AdaBoost, gdzie mieliśmy do czynienia z klasyfikatorami binarnymi, w algorytmie Real AdaBoost słabe klasyfikatory są rzeczywistoliczbowe. Można więc określić ten algorytm jako uogólnioną wersję algorytmu Discrete AdaBoost.

W algorytmie RealBoost każdy słaby klasyfikator wykorzystuje oszacowanie prawdopodobieństwa klasy pozytywnej pod warunkiem \mathbf{x} :

$$p_t(\mathbf{x}) = \widehat{P}_w(y = 1|\mathbf{x}) \in [0, 1], \quad (1.7)$$

zwracając odpowiedź równą połowie przekształcenia logit (Friedman, Hastie i Tibshirani, 2000):

$$h_t(\mathbf{x}) = \frac{1}{2} \log \left(\frac{p_t(\mathbf{x})}{1 - p_t(\mathbf{x})} \right). \quad (1.8)$$

Algorytm 3 przedstawia sposób działania algorytmu RealBoost.

Algorytm 3 Algorytm uczący klasyfikatora Real AdaBoost.

procedure REALADABOOSTTRAIN((\mathbf{x}_i, y_i)) $\triangleright i = 1, \dots, m, y_i \in \{-1, +1\}$
 Zainicjalizuj wszystkie wagi taką samą wartością: $w_i := \frac{1}{m}$, gdzie $i = 1, \dots, m$.
for $t := 1, \dots, T$ **do**
 Naucz słaby klasyfikator $h_t(\mathbf{x}) \in \mathbb{R}$, wykorzystując wagi w_1, \dots, w_m .
 Nauczony klasyfikator powinien odpowiadać zgodnie ze wzorem
 $h_t(\mathbf{x}) := \frac{1}{2} \log(p_t(\mathbf{x})/(1 - p_t(\mathbf{x})))$.
 Zaktualizuj wagi:
 $Z_t := \sum_{i=1}^m w_i$ (współczynnik normalizacji wag),
 $w_i := \frac{1}{Z_t} w_i e^{-y_i h_t(\mathbf{x}_i)}$, $i = 1, \dots, m$.
end for
 Zwóć zespół klasyfikatorów: h_1, \dots, h_T .
end procedure

Ostateczna odpowiedź klasyfikatora wyznaczana jest zgodnie ze wzorem:

$$H(\mathbf{x}_i) = \text{sgn} \left(\sum_{t=1}^T h_t(\mathbf{x}_i) \right). \quad (1.9)$$

1.4 Słabe klasyfikatory

Poniżej opisano wybrane klasyfikatory, które mogą zostać wykorzystane w algorytmach boostingowych. Klasyfikatory mają na celu utworzenie pewnych reguł, które pozwalają określić klasę dowolnej próbki na podstawie wartości jej atrybutów. Jakość klasyfikatora zależy w dużej mierze od wybranych atrybutów i informacji, jakie ze sobą niosą.

1.4.1 Drzewo decyzyjne

Drzewo decyzyjne (Quinlan, 1986) stanowi przykład algorytmu uczenia nadzorowanego, gdzie przy pomocy prostej struktury można dokonywać klasyfikacji przy-

kładów. W drzewie dane są wielokrotnie dzielone względem określonego atrybutu. Drzewo decyzyjne składa się z:

- węzłów — reprezentują testy bazujące na atrybutach i są połączone z kolejnymi węzłami; liczba krawędzi wychodzących z danego węzła jest równa liczbie możliwych wyników testu,
- krawędzi — reprezentują wynik testu i łączą ze sobą węzły, lub węzeł z liściem,
- liści — reprezentują odpowiedzi klasyfikatora.

Budując drzewo, należy wybrać atrybuty, które niosą ze sobą największą informację (pozwalają na jak najskuteczniejsze rozróżnienie klas na dany moment). W tym celu można stosować jedną z wielu miar, jak np. przyrost informacji, błąd klasyfikacji (Rokach i Maimon, 2005). W dalszej części pracy skupiono się na kryterium bazującym na indeksie Giniego (ang. *Gini Index*). Indeks ten może zostać użyty do określenia przydatności cech i miejsc podziałów zbioru względem odpowiedniej cechy. Mając zbiór D zawierający m próbek o możliwych C wartościach klasy, indeks Giniego dla zbioru D jest zdefiniowany jako:

$$\text{Gini}(D) = 1 - \sum_{c=1}^C p_c^2, \quad (1.10)$$

gdzie p_c to częstość występowania klasy c w ramach D .

Zakładając, że chcemy sprawdzić np. przydatność atrybutu binarnego A (o dwóch możliwych wartościach), zbiór można podzielić na dwa podzbiory zgodnie z wartościami tej cechy (np. nazwijmy je D_1 i D_2). Wtedy, indeks dla podzielonych danych, można zapisać jako:

$$\text{Gini}(D, A) = \frac{m_1}{m} \text{Gini}(D_1) + \frac{m_2}{m} \text{Gini}(D_2), \quad (1.11)$$

gdzie m_1 i m_2 to liczba przykładów w danym podzbiory. Cecha z najmniejszą wartością indeksu Giniego powinna zostać wybrana jako test w danym momencie budowy drzewa (Wu, Xin i Zheng, 2015).

Po zbudowaniu drzewa, klasyfikacja polega na wędrówce po nim, rozpoczynając od korzenia. Dla węzła przeprowadzany jest test i wybierana odpowiednia krawędź drzewa prowadząca do kolejnego węzła. Proces powtarza się tak długo, aż osiągnięty węzeł nie jest liściem. Wtedy, próbka otrzymuje klasę reprezentowaną przez liść. Warto zaznaczyć, że różne próbki mogą wykorzystywać inne zestawy atrybutów, w zależności od wyników testów na kolejnych węzłach.

Każde drzewo decyzyjne może zostać opisane przez parametr opisujący maksymalną możliwą głębokość drzewa, która wiąże się z liczbą liści. Parametr ten jest

istotny w momencie, gdy w związku ze złożonością zbioru danych nie jesteśmy w stanie wyodrębnić klas przy wykorzystaniu rozsądnej liczby atrybutów. Może wystąpić także sytuacja, że danych nie da się rozdzielić (np. bardzo zbliżone do siebie grupy próbek są różnych klas). Określając maksymalną głębokość drzewa można zapobiec sytuacji, w której nauczony klasyfikator zostałby przeuczony (zachowując przy tym możliwość generalizacji danych). Podejście takie jest często stosowane w kontekście nauki klasyfikatorów zespołowych, gdzie stosuje się wiele płytkich drzew.

1.4.2 Kosze z odpowiedzią rzeczywistoliczbową

W wielu algorytmach boostingowych popularnymi słabymi klasyfikatorami są tzw. decision stumps. Są to klasyfikatory równoważne najpłytszym drzewom decyzyjnym z dwoma liśćmi. Budowa takiego drzewa wiąże się z wybraniem jednego atrybutu oraz jednego progu na osi tego atrybutu. Lepszym podejściem (bazującym także na jednym atrybucie) są kosze, które pozwalają wspomnianą oś podzielić na więcej przedziałów.

Idea koszy została przedstawiona w pracy (Rasolzadeh, Petersson i Pettersson, 2006). W metodzie tej wartości danego atrybutu dzielone są na B równomiernych przedziałów, nazywanych koszami. Przynależność próbki \mathbf{x} do danego kosza $b \in \{1, \dots, B\}$ względem atrybutu j (który na zbiorze uczącym przyjmuje wartości z przedziału $[A_{j_{\min}}, A_{j_{\max}}]$) można określić na podstawie wzoru:

$$\text{bin}(\mathbf{x}; j) = \begin{cases} 1, & x_j \leq A_{j_{\min}}; \\ \left\lceil B \cdot \frac{x_j - A_{j_{\min}}}{A_{j_{\max}} - A_{j_{\min}}} \right\rceil, & A_{j_{\min}} < x_j \leq A_{j_{\max}}; \\ B, & A_{j_{\max}} < x_j. \end{cases} \quad (1.12)$$

Kosze mogą zostać zastosowane w algorytmach boostingowych, po uwzględnieniu wag próbek. Niech $W_c(b)$ oznacza sumę wag (prawdopodobieństw) przykładów klasy c przebywających w koszu o numerze b :

$$W_c(b) = \sum_{\substack{i \\ y_i=c \text{ i } \text{bin}(\mathbf{x}; j) \in b}} w_i. \quad (1.13)$$

Wtedy odpowiedź słabego klasyfikatora obliczona poprzez przekształcenie logit wynosi:

$$h(\mathbf{x}) = \frac{1}{2} \log \left(\frac{W_{-1}(\text{bin}(\mathbf{x}; j_t))}{W_{+1}(\text{bin}(\mathbf{x}; j_t))} \right), \quad (1.14)$$

gdzie j_t oznacza indeks najlepszej cechy wybranej w rundzie t .

1.4.3 Zastosowanie koszy w drzewach

Kosze mogą zostać także zastosowane w drzewach decyzyjnych (Klęsk, Godziuk i in., 2015). Dzięki zastosowaniu ich dla węzłów, możliwe jest ustalenie progów decyzyjnych, wybierając jeden z $B - 1$ progów określanych przez granice koszy. Dla każdej z cech badanych jest $B - 1$ możliwych progów decyzyjnych, przy wykorzystaniu indeksu Giniego. Ostatecznie dla danego węzła wybierana jest para (cecha, próg), która osiągnęła najmniejszą wartość używanej miary (np. indeksu Giniego). Dodatkowo, dla liści drzewa, modyfikacja ta, podobnie jak kosze z odpowiedzią rzeczywistoliczbową, zwraca odpowiedź w postaci połowy przekształcenia logit.

Zastosowanie koszy w drzewach decyzyjnych pozwala na uzyskanie dwóch korzyści. Po pierwsze, użycie koszy przyspiesza uczenie pojedynczego drzewa decyzyjnego poprzez rozważanie mniejszej liczby rozcięć — dużo mniejszej niż liczba przykładów. Po drugie, nie ma także konieczności sortowania próbek wzdłuż każdej cechy, co przy dużej liczbie przykładów mocno spowalniałoby algorytm.

Rozdział 2

Momenty Zernike’a

W tym rozdziale przedstawione zostaną podstawowe informacje dotyczące notacji matematycznej, a także momentów Zernike’a (dalej w pracy oznaczane poprzez anglojęzyczny skrót ZM).

W całej pracy używany będzie symbol f dla oznaczenia pewnej funkcji matematycznej, która w zależności od kontekstu może być funkcją dwóch zmiennych ciągłych lub też funkcją obrazu z dwiema zmiennymi dyskretnymi (indeksami pikseli). Nazwy argumentów będą jednoznacznie wskazywały właściwe rozumienie dla f i rozważanego układu współrzędnych:

$f(r, \theta)$ — współrzędne: ciągłe, biegunowe, lokalne (koło jednostkowe),

$f(x, y)$ — współrzędne: ciągłe, kartezjańskie, lokalne (koło jednostkowe),

$f(j, k)$ — współrzędne: dyskretnie, kartezjańskie, lokalne (okno)

lub globalne (okno obrazu) w zależności od kontekstu;

indeksy piksela: j — wiersz, k — kolumna.

W odniesieniu do liczb zespolonych i oznaczać będzie jednostkę urojoną ($i^2 = -1$), a pozioma kreska nad wyrażeniem będzie używana do oznaczenia sprzężenia liczby zespolonej: $\overline{a + ib} = a - ib$. Pary nawiasów $[\cdot]$, $[\cdot]$ oznaczać będą funkcje odpowiednio podłoga oraz sufit¹. Zapisy $\langle \cdot, \cdot \rangle$ i $\| \cdot \|$ będą używane do oznaczenia odpowiednio iloczynu skalarnego oraz normy. Znak $\#$ umieszczony przed zbiorem zwracać będzie jego rozmiar.

2.1 Wielomiany i momenty Zernike’a

Wielomiany Zernike’a stanowią bazę ortogonalnych funkcji nad kołem jednostkowym. Wielomian rzędu (p, q) jest definiowany jako iloczyn części radialnej oraz

¹Funkcje zaokrąglające liczby rzeczywiste do liczb całkowitych, odpowiednio w dół i w górę.

harmonicznej i może zostać zapisany w układzie współrzędnych biegunowych jako:

$$V_{p,q}(r, \theta) = R_{p,q}(r)F_q(\theta), \quad (2.1)$$

gdzie

$$R_{p,q}(r) = \sum_{s=0}^{(p-|q|)/2} \beta_{p,q,s} r^{p-2s} \quad (2.2)$$

oznacza wielomian radialny (czyli określony nad zmienną promienia r) ze współczynnikami:

$$\beta_{p,q,s} = \frac{(-1)^s (p-s)!}{s! \left(\frac{p+q}{2} - s\right)! \left(\frac{p-q}{2} - s\right)!}, \quad (2.3)$$

a $F_q(\theta) = e^{iq\theta}$ oznacza część harmoniczną (związaną z przekształceniem Fouriera). Różnica $p - |q|$ musi być parzysta oraz dodatkowo $p \geq |q|$.

Powyższa formuła wynika z procedury ortogonalizacji Grama-Schmidta przeprowadzonej dla następującego zbioru funkcji:

$$\{1, re^{i\theta}, r^2, r^2 e^{i2\theta}, r^3 e^{i\theta}, r^3 e^{i3\theta}, r^4, r^4 e^{i2\theta}, r^4 e^{i4\theta}, \dots\}. \quad (2.4)$$

Jak można zauważyć, potęga przy promieniu oraz indeks harmoniczny są jednocześnie parzyste lub nieparzyste. Dodatkowo, indeksy harmoniczne są nie większe niż potęgi radialne. W związku z faktem, że wielomiany Zernike'a są funkcjami o wartościach zespolonych, używany w tym przypadku w procedurze ortogonalizacji jest hermitowski iloczyn skalarny (na kole jednostkowym):

$$\langle g, h \rangle = \int_0^{2\pi} \int_0^1 g(r, \theta) \overline{h(r, \theta)} r dr d\theta, \quad (2.5)$$

a następnie stosowany jest warunek normalizacji, który narzuca, aby $R_{p,q}(1) = 1$.

Można sprawdzić, że kwadraty norm wielomianów Zernike'a wynoszą:

$$\|V_{p,q}\|^2 = \langle V_{p,q}, V_{p,q} \rangle = \int_0^{2\pi} \int_0^1 V_{p,q}(r, \theta) V_{p,-q}(r, \theta) r dr d\theta = 2\pi a_p, \quad (2.6)$$

gdzie $a_p = 1/(2(p+1))$; dlatego też własność ortogonalności dla wspomnianych wielomianów może zostać wyrażona następująco:

$$\langle V_{p,q}, V_{m,n} \rangle = \begin{cases} 0, & \text{dla } (p, q) \neq (m, n); \\ 2\pi a_p, & \text{dla } (p, q) = (m, n). \end{cases} \quad (2.7)$$

Rozważmy funkcję f zdefiniowaną na kole jednostkowym, której rozwinięcie z użyciem wielomianów Zernike'a to:

$$f(r, \theta) = \sum_{0 \leq p < \infty} \sum_{\substack{-p \leq q \leq p \\ p-|q| \text{ parzyste}}} M_{p,q} V_{p,q}(r, \theta). \quad (2.8)$$

Można pokazać, że optymalne zespolone współczynniki $M_{p,q}$ rozwinięcia (2.8) — *momenty* — mogą zostać zapisane jako:

$$\begin{aligned} M_{p,q} &= \frac{\langle f, V_{p,q} \rangle}{\|V_{p,q}\|^2} = \frac{1}{2\pi a_p} \int_0^{2\pi} \int_0^1 f(r, \theta) R_{p,q}(r) F_{-q}(\theta) r dr d\theta \\ &= \frac{1}{2\pi a_p} \int_0^{2\pi} \int_0^1 f(r, \theta) \sum_{s=0}^{(p-|q|)/2} \beta_{p,q,s} r^{p-2s} e^{-iq\theta} r dr d\theta. \end{aligned} \quad (2.9)$$

Momenty Zernike’a są niezmiennicze ze względu na skalowanie, a dodatkowo ich *moduły* są niezmiennicze ze względu na obrót. Praca skupia się w szczególności na drugiej z tych własności.

2.2 Niezmienniczość ze względu na obrót i skalowanie

Własność 1. *Moduły momentów Zernike’a są niezmiennicze ze względu na obrót.*

Dowód. Załóżmy, że g oznacza funkcję powstałą poprzez obrót funkcji f o dowolny kąt δ wokół środka układu współrzędnych, tj. $g(r, \theta) = f(r, \theta + \delta)$. Moduł momentu Zernike’a w dla funkcji g wynosi:

$$\begin{aligned} &\left| \frac{1}{2\pi a_p} \int_0^{2\pi} \int_0^1 g(r, \theta) \sum_{s=0}^{(p-|q|)/2} \beta_{p,q,s} r^{p-2s} e^{-iq\theta} r dr d\theta \right| \\ &= \left| \frac{1}{2\pi a_p} \int_0^{2\pi} \int_0^1 f(r, \theta + \delta) \sum_{s=0}^{(p-|q|)/2} \beta_{p,q,s} r^{p-2s} e^{-iq\theta} r dr d\theta \right|. \end{aligned}$$

Po podstawieniu $\gamma = \theta + \delta$ oraz $d\gamma = d\theta$ otrzymujemy:

$$\begin{aligned} &\left| \frac{1}{2\pi a_p} \int_0^{2\pi} \int_0^1 f(r, \gamma) \sum_{s=0}^{(p-|q|)/2} \beta_{p,q,s} r^{p-2s} e^{-iq(\gamma-\delta)} r dr d\gamma \right| \\ &= \left| e^{iq\delta} \frac{1}{2\pi a_p} \int_0^{2\pi} \int_0^1 f(r, \gamma) \sum_{s=0}^{(p-|q|)/2} \beta_{p,q,s} r^{p-2s} e^{-iq\gamma} r dr d\gamma \right| \\ &= \underbrace{|e^{iq\delta}|}_1 \left| \frac{1}{2\pi a_p} \int_0^{2\pi} \int_0^1 f(r, \gamma) \sum_{s=0}^{(p-|q|)/2} \beta_{p,q,s} r^{p-2s} e^{-iq\gamma} r dr d\gamma \right|. \end{aligned} \quad (2.10)$$

□

Własność 2. *Momenty Zernike’a są niezmiennicze ze względu na skalowanie.*

Dowód. Rozważmy funkcję g zdefiniowaną na kole jednostkowym o promieniu $R > 0$, $g: [0, R] \times [0, 2\pi] \rightarrow \mathbb{R}$. Załóżmy, że funkcja f powstała poprzez przeskalowanie funkcji g do koła jednostkowego w następujący sposób:

$$f(t, \theta) = g(tR, \theta), \quad t \in [0, 1],$$

gdzie t jest tymczasowym oznaczeniem zmiennej promienia (w celu uniknięcia kolizji oznaczeń z r , które pojawi się później). Pola powierzchni dziedzin obydwu funkcji są do siebie proporcjonalne z czynnikiem R^2 .

Momenty Zernike’a funkcji f mogą zostać zapisane jako:

$$\begin{aligned} M_{p,q} &= \frac{1}{2\pi a_p} \int_0^{2\pi} \int_0^1 f(t, \theta) R_{p,q}(t) e^{-iq\theta} t dt d\theta \\ &= \frac{1}{2\pi a_p} \int_0^{2\pi} \int_0^1 g(tR, \theta) \sum_{s=0}^{(p-|q|)/2} \beta_{p,q,s} t^{p-2s} e^{-iq\theta} t dt d\theta. \end{aligned}$$

Podstawienia $r = tR$ oraz $dr = R dt$, a także zmiana granic całkowania z $0 \leq t \leq 1$ na $0 \leq r \leq R$, prowadzą do wyrażenia:

$$\begin{aligned} \frac{1}{2\pi a_p} \int_0^{2\pi} \int_0^R g(r, \theta) \sum_{s=0}^{(p-|q|)/2} \beta_{p,q,s} \left(\frac{r}{R}\right)^{p-2s} e^{-iq\theta} \frac{r}{R} \frac{dr}{R} d\theta \\ = \frac{1}{R^2} \frac{1}{2\pi a_p} \int_0^{2\pi} \int_0^R g(r, \theta) R_{p,q} \left(\frac{r}{R}\right) e^{-iq\theta} r dr d\theta = \frac{1}{R^2} M_{p,q} R^2. \end{aligned}$$

Ostatnie przekształcenie jest prawdziwe, ponieważ argumenty wielomianu radialnego $R_{p,q}(r/R)$, który występuje pod całkowaniem, pozostają w przedziale $[0, 1]$, zaś całkowany obszar jest większy od oryginalnego zgodnie z proporcją określoną czynnikiem R^2 . \square

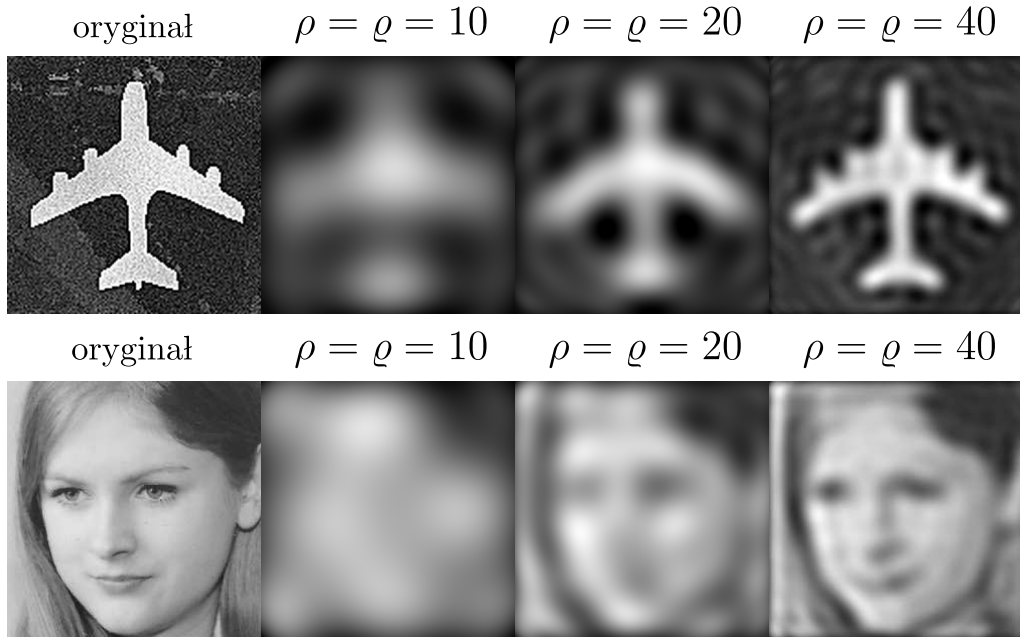
Warto także zauważyć, że dowód na niezmienniczość ortogonalnych momentów Fouriera–Mellina jest analogiczny do tego dla momentów Zernike’a, ponieważ różnice w ich matematycznych zapisach sprowadzają się jedynie do części wielomianowych i ich współczynników.

2.3 Aproksymacja funkcji poprzez częściową sumę rozwinięcia

W praktyce funkcja f (lub obraz) jest aproksymowana przez częściową sumę pełnego rozwinięcia (2.8). Niech ρ i ϱ (gdzie $\rho \geq \varrho$) oznaczają maksymalny rząd aproksymacji: odpowiednio wielomianowy i harmoniczny. Wtedy, częściowa suma może zostać zapisana w następujący sposób:

$$f(r, \theta) \approx \sum_{0 \leq p \leq \rho} \sum_{\substack{-\min\{p, \varrho\} \leq q \leq \min\{p, \varrho\} \\ p-|q| \text{ parzyste}}} M_{p,q} V_{p,q}(r, \theta). \quad (2.11)$$

Rys. 2.1 przedstawia przykłady rekonstrukcji obrazów przy pomocy wielomianów Zernike’a z użyciem wzoru (2.11).



Rysunek 2.1: Przykłady rekonstrukcji obrazów wykorzystującej wielomiany Zernike’a. (Źródło: opracowanie własne)

2.4 Związek momentów Zernike’a z ortogonalnymi momentami Fouriera–Mellina

Ortogonalny moment Fouriera–Mellina (dalej w pracy używany będzie skrót angielszczyzny OFMM dla oznaczenia tych momentów) rzędu (p, q) można zapisać w następujący sposób:

$$O_{p,q} = \frac{1}{2\pi a_p} \int_0^{2\pi} \int_0^1 f(r, \theta) \sum_{s=0}^p \alpha_{p,s} r^s e^{-iq\theta} r dr d\theta, \quad (2.12)$$

gdzie:

$$\alpha_{p,s} = (-1)^{p+s} \frac{(p+s+1)!}{s!(s+1)!(p-s)!}. \quad (2.13)$$

Można zauważyć podobieństwo pomiędzy wyrażeniami (2.9) i (2.12). Różnica między nimi to współczynniki radialne ($\alpha_{p,s}$ w miejscu $\beta_{p,q,s}$) oraz odwrotna kolejność sumowania składników potęgowych. Jednak kluczową różnicą, jaką można zaobserwować, jest fakt, że warunki wiążące indeksy p i q są poluzowane. W OFMM p i q nie muszą być równocześnie parzyste czy też nieparzyste, a także $|q|$ może być większe od p . Powodem tego jest fakt, że OFMM są generowane poprzez ortogonalizację Grama-Schmidta zastosowaną *niezależnie* dla części radialnej, to znaczy dla zbioru potęg $\{1, r, r^2, \dots\}$ nad kołem jednostkowym, a następnie ogólna radialno-harmoniczna baza jest konstruowana produktowo. W efekcie, baza zawiera także mieszane termy (parzysto-nieparzyste) na przykład $r^7 e^{i4\theta}$, $r^2 e^{i3\theta}$, itd. Pod tym

względem, OFMM można uznać za momenty bardziej ogólne niż ZM. Jednakże, jak przedstawiono w dalszej części pracy, w zadaniu detekcji ta własność jest przeszkodą w momencie konstruowania obrazów całkowych dla OFMM.

Z drugiej strony, jeśli mowa o możliwościach rozpoznawania obrazów z wykorzystaniem ZM i OFMM, w literaturze (Abu-Mostafa i Psaltis, 1984; Teh i Chin, 1988) pokazano zarówno analitycznie jak i eksperymentalnie, że OFMM mają pewną przewagę nad ZM. W szczególności, ZM cierpią na tłumienie informacji w centralnych częściach obrazu oraz szumy i drgania na jego granicach (tzw. *ringing effect*).

2.5 Porównanie momentów Zernike’a z ortogonalnymi momentami Fouriera–Mellina

2.5.1 Bazy ortogonalne i rozkład zer w wielomianach radialnych

Dla przypomnienia bazy ortogonalne Zernike’a oraz Fouriera–Mellina (nad kołem jednostkowym) są wyrażone odpowiednio jako:

$$V_{p,q}(r) = R_{p,q}(r)F_q(\theta) = \sum_{s=0}^{(p-|q|)/2} \beta_{p,q,s} r^{p-2s} e^{iq\theta}, \quad (2.14)$$

$$B_{p,q}(r) = Q_p(r)F_q(\theta) = \sum_{s=0}^p \alpha_{p,s} r^s e^{iq\theta}, \quad (2.15)$$

z odpowiadającymi im współczynnikami:

$$\beta_{p,q,s} = \frac{(-1)^s (p-s)!}{s! \left(\frac{p+q}{2} - s\right)! \left(\frac{p-q}{2} - s\right)!},$$

$$\alpha_{p,s} = \frac{(-1)^{p+s} (p+s+1)!}{(p-s)! s! (s+1)!}.$$

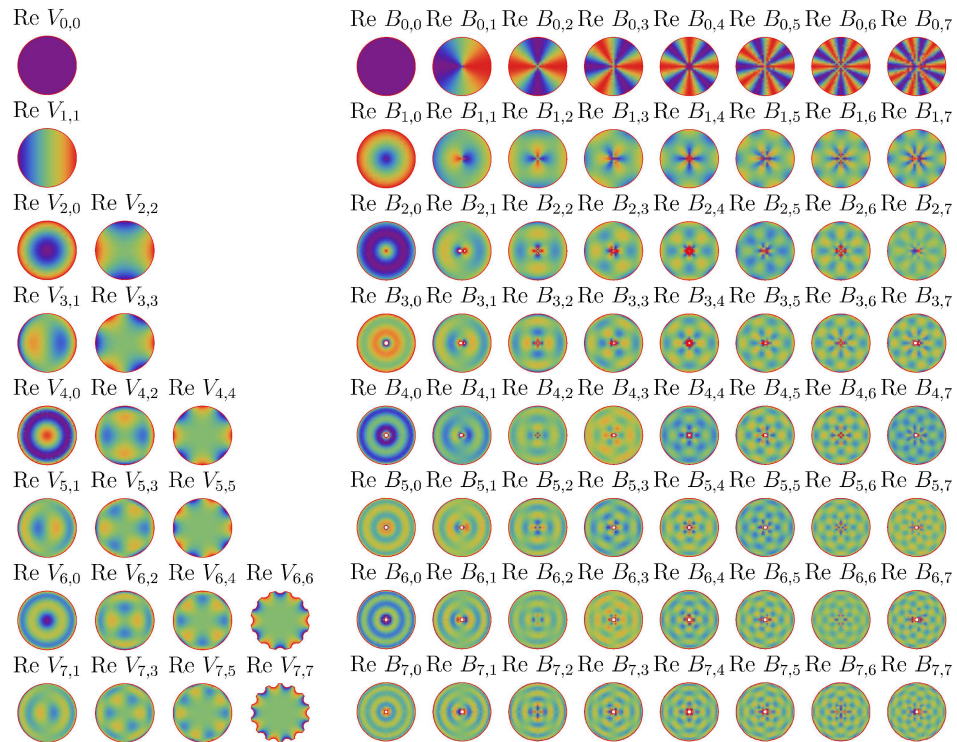
Początkowe wielomiany radialne w obu bazach są następujące:

$$\begin{aligned}
 R_{0,0}(r) &= 1, \\
 Q_0(r) &= 1, \\
 R_{1,1}(r) &= r, \\
 Q_1(r) &= -2 + 3r, \\
 R_{2,0}(r) &= -1 + 2r^2, \\
 R_{2,2}(r) &= r^2, \\
 Q_2(r) &= 3 - 12r + 10r^2, \\
 R_{3,1}(r) &= -2r + 3r^3, \\
 R_{3,3}(r) &= r^3, \\
 Q_3(r) &= -4 + 30r - 60r^2 + 35r^3, \\
 R_{4,0}(r) &= 1 - 6r^2 + 6r^4, \\
 R_{4,2}(r) &= -3r^2 + 4r^4, \\
 R_{4,4}(r) &= r^4, \\
 Q_4(r) &= 5 - 60r + 210r^2 - 280r^3 + 126r^4, \\
 R_{5,1}(r) &= 3r - 12r^3 + 10r^5, \\
 R_{5,3}(r) &= -4r^3 + 5r^5, \\
 R_{5,5}(r) &= r^5, \\
 Q_5(r) &= -6 + 105r - 560r^2 + 260r^3 - 1260r^4 + 462r^5, \\
 R_{6,0}(r) &= -1 + 12r^2 - 30r^4 + 20r^6, \\
 R_{6,2}(r) &= 6r^2 - 20r^4 + 15r^6, \\
 R_{6,4}(r) &= -5r^4 + 6r^6, \\
 R_{6,6}(r) &= r^6, \\
 Q_6(r) &= 7 - 168r + 1260r^2 - 4200r^3 + 6930r^4 - 5544r^5 + 1716r^6, \\
 R_{7,1}(r) &= -4r + 30r^3 - 60r^5 + 35r^7, \\
 R_{7,3}(r) &= 10r^3 - 30r^5 + 21r^7, \\
 R_{7,5}(r) &= -6r^5 + 7r^7, \\
 R_{7,7}(r) &= r^7, \\
 Q_7(r) &= -8 + 252r - 2520r^2 + 11550r^3 - 27720r^4 + 36036r^5 - 24024r^6 + 6435r^7, \\
 &\vdots
 \end{aligned}$$

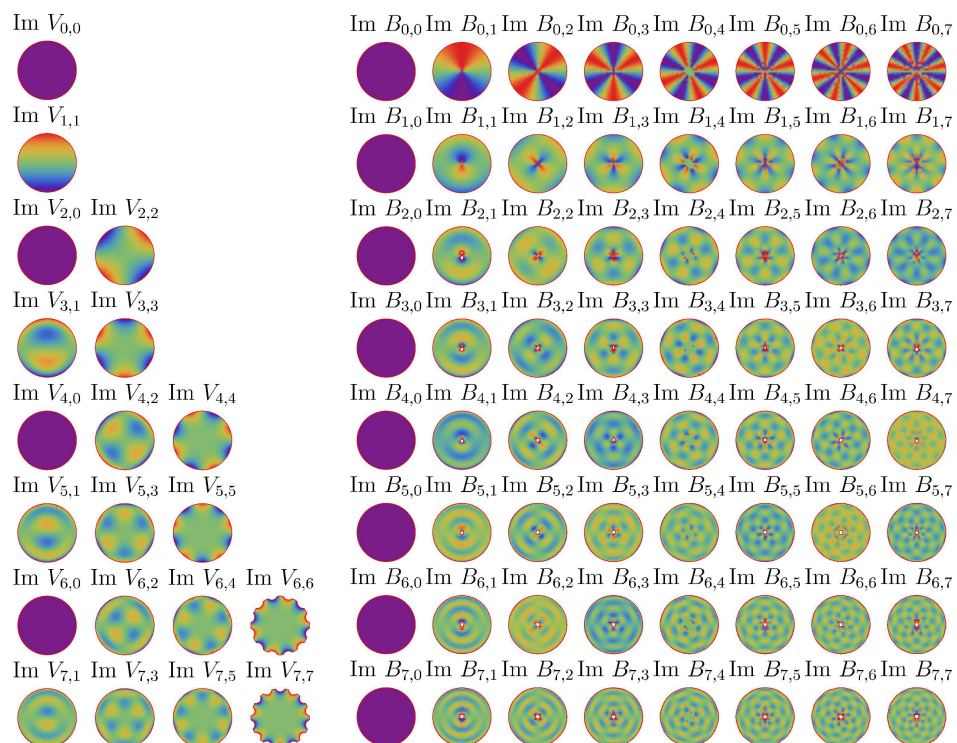
Rysunki 2.2 i 2.3 przedstawiają ich wykresy porównawcze z rozróżnieniem części rzeczywistej i urojonej.

Liczba zer w wielomianach odpowiada ich zdolności do opisywania cech funkcji lub obrazów o wysokiej częstotliwości, które mają być aproksymowane (Sheng i Shen, 1994). Porównując wielomiany radialne dla obu wersji baz Zernike’a i ortogonalnych baz Fouriera–Mellina (patrz także Rys. 2.4) można zaobserwować, że:

- (a) wielomiany $R_{p,q}$ są ograniczone do przedziału $[-1, 1]$;
- (b) wielomiany Q_p są nieograniczone;
- (c) wielomian $R_{p,q}(r)$ ma $(p-|q|)/2$ podwojonych zer dla $0 < r \leq 1$ (oprócz $R_{p,q}(0) = 0$);

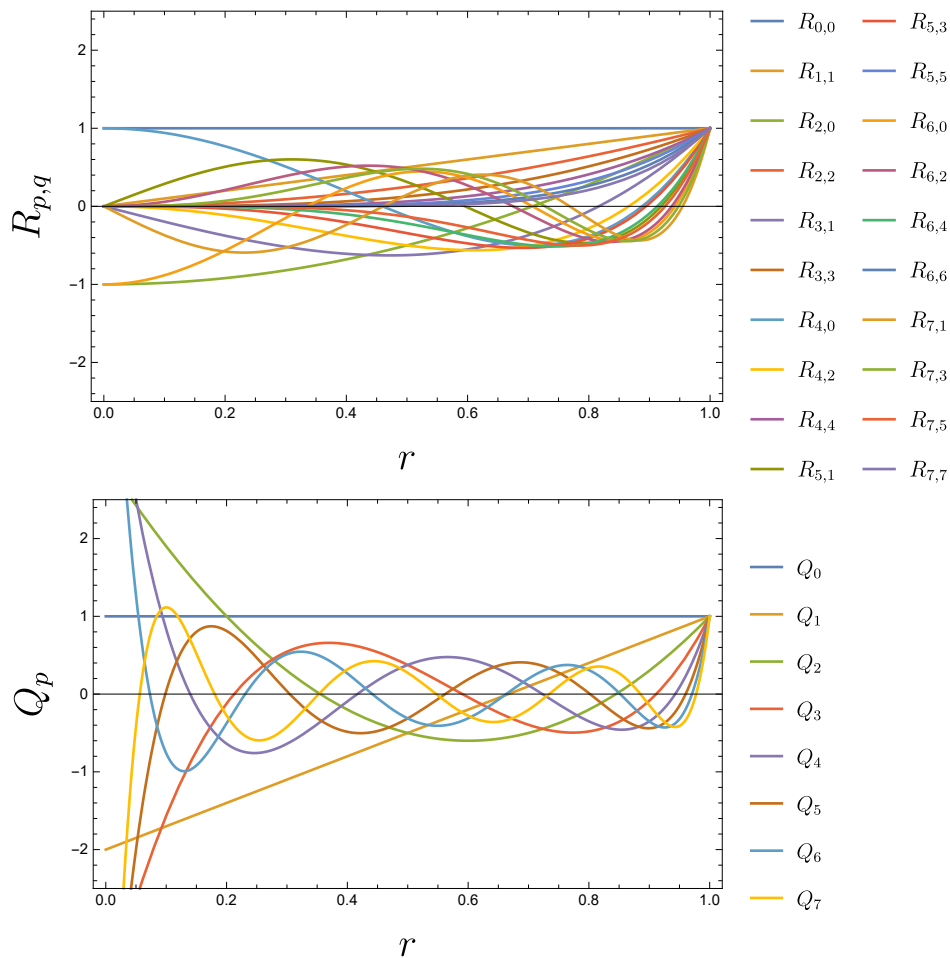


Rysunek 2.2: Części rzeczywiste ortogonalnych baz Zernike’a i Fouriera–Mellina zdefiniowane na kole jednostkowym. (Źródło: opracowanie własne)



Rysunek 2.3: Części urojone ortogonalnych baz Zernike’a i Fouriera–Mellina zdefiniowane na kole jednostkowym. (Źródło: opracowanie własne)

- (d) wielomian $Q_p(r)$ ma p zer (niezależnie od q);
- (e) aby mieć p zer używając pewnego wielomianu Zernike’a, tzn. taką samą liczbę zer jak w wielomianie Q_p , rząd wielomianu Zernike’a musiałby zostać podniesiony, tzn. należałoby użyć $R_{p+q}(r)$;
- (f) porównując wielomiany z tą samą liczbą zer, można zaobserwować, że zera w wielomianach Q są rozmieszczone prawie równomiernie, podczas gdy zera w wielomianach R są zagęszczone blisko punktu $r = 1$, tj. w kierunku granic obrazu, patrząc na koło jednostkowe (Sheng i Shen, 1994).



Rysunek 2.4: Wielomiany radialne baz Zernike’a oraz Fouriera–Mellina. (Źródło: opracowanie własne)

Ostatnia ze wspomnianych własności oznacza, że ZM cierpią na pewną utratę informacji — jej tłumienie (ang. *information suppression*) — w środkowej części obrazu (zostanie to dokładniej opisane w 2.5.2) i efekt drgań na jego granicach. Zostało także pokazane (Sheng i Shen, 1994), że ZM są gorsze od OFMM pod względem

błędu przy rekonstrukcji obrazu oraz stosunku informacji do szumu (*signal-to-noise ratio*). Z drugiej strony, eliminując z rozważań OFMM, badania przeprowadzone w pracy (Teh i Chin, 1988) pokazały, że ZM przewyższają inne funkcje momentów takie jak: zwykle momenty geometryczne, momenty Legendre’a, standardowe momenty Fouriera–Mellina i momenty zespolone (dalej w pracy określane poprzez skrót anglojęzyczny CM).

2.5.2 Tłumienie informacji

Pewną technikę do analizy tłumienia informacji tkwiącej w momentach zaproponowali Abu-Mostafa i Psaltis (1984). Poniżej zastosowano opisaną przez nich ideę.

Mając daną funkcję $f(r, \theta)$ zdefiniowaną nad kołem jednostkowym, rozważmy jej *harmoniczne kołowe rozwinięcie* dla każdego ustalonego promienia r — tj. rozważmy nieskończony zbiór zespolonych współczynników Fouriera $\{c_n(r)\}_{n=0, \pm 1, \pm 2, \dots}$, gdzie:

$$c_n(r) = \frac{1}{\|F_n\|^2} \langle f, F_n \rangle = \frac{1}{2\pi} \int_0^{2\pi} f(r, \theta) e^{-in\theta} d\theta. \quad (2.16)$$

Zauważmy, że zmieniając kolejność całkowania we wzorze (2.9) na ZM, momenty mogą zostać wyrażone w terminach rozwinięć kołowych w następujący sposób:

$$\begin{aligned} M_{p,q} &= \frac{1}{2\pi a_p} \int_0^1 \sum_{s=0}^{(p-|q|)/2} \beta_{p,q,s} r^{p-2s} \int_0^{2\pi} f(r, \theta) e^{-iq\theta} d\theta r dr \\ &= \frac{1}{a_p} \int_0^1 r R_{p,q}(r) c_q(r) dr. \end{aligned} \quad (2.17)$$

Co interesujące, we wzorze (2.17) można zaobserwować, że każdy moment $M_{p,q}$ zależy *wyłącznie* od pojedynczego współczynnika $c_q(r)$ (traktowanego jako funkcja promienia). Ta obserwacja wskazuje, które cechy obrazu wnoszą wkład do danego momentu Zernike’a.

Pierwotnie, Abu-Mostafa i Psaltis (1984) rozważali *zespolone momenty* (CM) zdefiniowane jako:

$$\begin{aligned} C_{p,q} &= \iint_{x^2+y^2 \leq 1} f(x, y) (x + iy)^p (x - iy)^q dx dy \\ &= \int_0^{2\pi} \int_0^1 f(r, \theta) r^{p+q} e^{i(p-q)\theta} r dr d\theta, \end{aligned}$$

które mogą zostać potraktowane jako „cegiełki” budujące momenty ZM lub OFMM — ZM i OFMM mogą zostać przedstawione jako liniowe kombinacje CM. Podstawienie $n = p - q$, $m = q$ prowadzi do zapisania CM analogicznie do (2.17):

$$C_{n+m,m} = 2\pi \int_0^1 r^{n+2m+1} c_{-n}(r) dr. \quad (2.18)$$

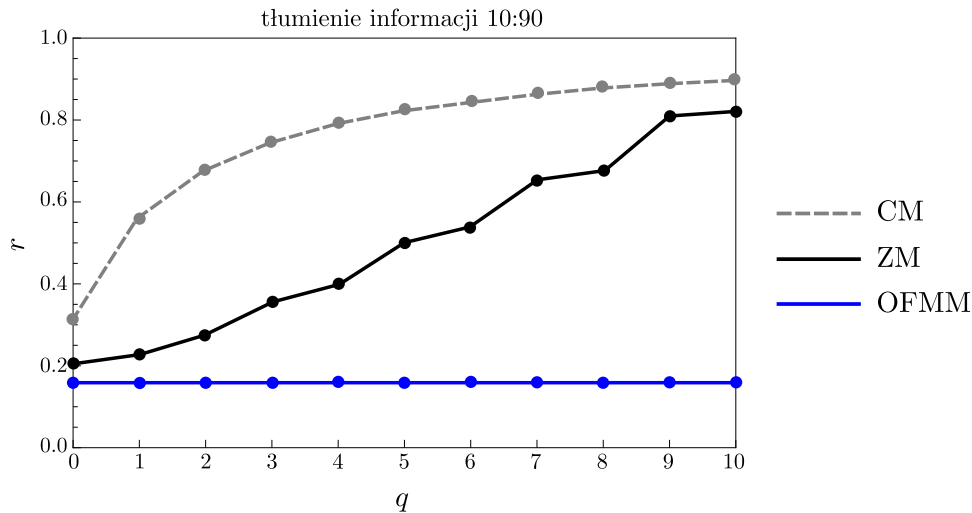
Jądro (ang. *kernel function*) we wzorze (2.18) stanowi funkcja r^{n+2m+1} . Można na nią patrzeć jak na funkcję wagującą, która ekstrahuje pewną cechę obrazu — współczynnik $c_{-n}(r)$ — z różną siłą w zależności od tego, gdzie dana cecha występuje wzdłuż zmiennej r . Można zaobserwować, że gdy wykładnik $n + 2m + 1$ staje się duży, to informacja o $c_{-n}(r)$ dla małych wartości r zostaje *stłumiona*, zaś informacja tkwiąca w części $c_{-n}(r)$ leżącej blisko $r = 1$ zdominowuje finalną wartość momentu $C_{n+m,m}$. Warto zauważyć, że problemem nie jest fakt, że pewien wybrany moment (ekstrahowany jako funkcja obrazu) tłumii pewną informację z centrum obrazu, zaś to, że *wszystkie* wyekstrahowane cechy tłumią *tę samą* informację (Abu-Mostafa i Psaltis, 1984).

Przeprowadzone zostanie teraz to samo rozumowanie dla ZM oraz ilościowa ocena tłumienia informacji. Zastosowana zostanie funkcja jądrowa $k_{p,q}(r) = rR_{p,q}(r)$ ze wzoru (2.17). W celu oceny stopnia tłumienia obserwowany będzie poniższy stosunek norm ℓ^1 dla funkcji jądrowych:

$$K_{p,q}(t) = \int_0^t |k_{p,q}(r)| dr / \int_0^1 |k_{p,q}(r)| dr \quad (2.19)$$

traktowany jako funkcja granicy całkowania t . Stosunek ten pozwala na uzyskanie oceny (znormalizowanej do przedziału $[0, 1]$) o tym, jak silnie informacja o cesze $c_q(r)$ w zakresie $r \in [0, t]$ jest tłumiona w finalnej wartości momentu $M_{p,q}$. Można także zastosować podejście odwrotne — narzucając pewien próg ξ , można rozwiązać równanie $K_{p,q}(t) = \xi$ ze względu na zmienną t . Dzięki temu uzyskujemy ocenę, jak mały jest przedział $(t, 1]$, który gromadzi część informacji równą $1 - \xi$ na temat cechy $c_q(r)$. Kolejny akapit przedstawia konkretny przykład liczbowy wraz z wykresem oceny tłumienia informacji.

Rozważmy skończoną przestrzeń cech, która bazuje na ZM i jest generowana poprzez $q = 0, \pm 1, \dots, \pm \varrho$ i $p = |q|, |q| + 2, \dots, \rho - (\rho - |q|) \bmod 2$. Warto zauważyć, że dla każdego ustalonego q istnieje *zbiór* momentów $M_{p,q}$ bazujących na tym samym współczynniku $c_q(r)$ w (2.17). Można założyć optymistyczną ocenę tłumienia znajdując dla ustalonego q minimum po zmiennej p spośród rozwiązań równania $K_{p,q}(t) = \xi$. Rys. 2.5 obrazuje wyniki otrzymane dla $\xi = 0.1$. Wykres można zinterpretować w następujący sposób. Dla każdego indeksu q wysokość odcinka pod krzywą reprezentuje najmniejszy przedział dla zmiennej r , który gromadzi 10% wagi funkcji jądrowej (opisującej cechę c_q), zaś odcinek ponad krzywą reprezentuje pozostały przedział gromadzący 90% wagi. Dla porównania pokazano analogiczne krzywe dla momentów CM i OFMM. Można łatwo zauważyć, że wraz ze wzrostem rzędu harmonicznego, ZM przechowują głównie informacje z brzegu obrazu. Problem ten jest znacznie bardziej poważny dla CM (ze względu na ich nieortogonalność), zaś nie występuje wcale w przypadku OFMM.



Rysunek 2.5: Ilustracja tłumienia informacji (z zastosowaniem proporcji 10%:90%) dla: momentów zespolonych (CM), momentów Zernike’a (ZM) i ortogonalnych momentów Fouriera–Mellina (OFMM). (Źródło: opracowanie własne)

2.6 Momenty Zernike’a w literaturze

Poniżej przedstawiono kilka wybranych zastosowań momentów Zernike’a opisanych w licznych pracach. Warto jednak zaznaczyć, że cytowane prace dotyczą głównie zadania rozpoznawania, a nie detekcji, a obliczenia przeprowadzane są jedynie na wstępnie wydzielonych fragmentach obrazu (bez wykorzystania obrazów całkowych). Kan i Srinath (2002) za pomocą momentów Zernike’a oraz ortogonalnych momentów Fouriera-Mellina klasyfikowali znaki alfanumeryczne. Iscan, Dokur i Ölmez (2010) zaprezentowali pracę, w której wykrywali guzy mózgow na podstawie obrazów z rezonansu magnetycznego w wykorzystaniem momentów Zernike’a. Mai i Ning (2015) wykorzystali momenty Zernike’a do wykrywania krawędzi. Ortogonalne momenty Fouriera-Mellina były z powodzeniem używane do rozpoznawania obiektów na obrazach z rozmyciami (Liu, Zhu i Q. Li, 2011), automatycznego rozpoznawania odcisków butów (Gueham i in., 2008), czy wykrywania twarzy (Terrillon i in., 2000). Malek, Azimifar i Boostani (2017) zaproponowali system określający wiek człowieka bazując na obrazach twarzy. Autorom udało się uzyskać dobre wyniki używając momentów Zernike’a oraz perceptronów. Yin, Meng i S. Li (2017) rozpoznawali symbole elektryczne ze szkiców inżynierskich. Zhou i in. (2016) zaprezentowali zastosowanie momentów Zernike’a do detekcji poruszających się obiektów. Co ciekawe, autorzy wyznaczali jedynie dwa momenty M_{31} , M_{33} i wskazali ich powiązanie z poruszającymi się obiektami. Można odnaleźć także podejścia, w których autorzy łączyli momenty Zernike’a z innymi typami cech. Xing i in. (2016) zaproponowali system do detekcji i rozpoznawania znaków drogowych bazując na mo-

mentach Zernike’a w połączeniu z techniką standaryzacji kolorów. Sornam, Kavitha i Shalini (2016) wykorzystali cechy bazujące na falkach oraz momentach Zernike’a w celu przygotowania automatycznej diagnostyki guzów bazując na obrazach z rezonansu magnetycznego. Liang i in. (2020) zaproponowali algorytm bazujący na normalizacji obrazu oraz momentach Zernike’a, który ma służyć do identyfikacji gwiazd na podstawie obrazów pozyskanych z teleskopu. Algorytm ten pozwala na przyporządkowanie gwiazd do odpowiednich pozycji w katalogu. K. Wang, H. Wang i J. Wang (2020) przedstawili sposób na dopasowywanie terenu bazując na połączeniu momentów Zernike’a i deskryptora HOG na danych pozyskanych z radaru z syntetyczną aperturą (ang. *Synthetic Aperture Radar*, w skrócie SAR) oraz radaru REM.

W literaturze można odnaleźć wiele prób, w których autorzy starają się przyspieszyć obliczanie momentów Zernike’a lub Fouriera-Mellina. Na przykład, Gu i in. (2002) uwzględniając zależności pomiędzy momentami Zernike’a wyrazili moment rzędu p poprzez momenty rzędu $p-2$ i $p-4$, zmniejszając tym samym łączną liczbę momentów, które trzeba wyznaczać. Papakostas i in. (2007b) zaproponowali rekurencyjną metodę dla ortogonalnych momentów Fouriera-Mellina, która pozwala na eliminację części obliczeń związanych z silnią. Jeszcze w tym samym roku, Ci sami autorzy Papakostas i in. (2007a) przedstawili metody, które pozwalają na całkowitą eliminację obliczeń związanych z silnią dla ortogonalnych momentów Fouriera-Mellina. Hosny, Shouman i Salam (2011) zaproponowali jeszcze szybszą metodę wyznaczania ortogonalnych momentów Fouriera-Mellina. W metodzie tej biegunowy układ współrzędnych zostaje odpowiednio podzielony na sektory, a w późniejszych obliczeniach możliwe jest wykorzystanie symetrii w nich występujących. Pozwala to jednocześnie na redukcję liczby wybranych do obliczeń punktów, jak i błędów numerycznych.

Rozdział 3

Stałoczasowe obliczanie momentów Zernike'a z wykorzystaniem obrazów całkowych

W niniejszym rozdziale przedstawione zostały przekształcenia, które umożliwią zastosowanie obrazów całkowych do wyznaczania momentów Zernike'a, a co za tym idzie ich stałoczasowe obliczanie, co stanowi sedno pracy.

3.1 Przekształcenia momentów Zernike'a służące wprowadzeniu obrazów całkowych

3.1.1 Odwrócona kolejność sumowania

Podrozdział ten zaczyna się od zapisania równoważnej postaci dla wielomianów Zernike'a z odwróconą kolejnością sumowania. Forma ta będzie przydatna do późniejszych rozważań. Bez straty ogólności, od tej pory rozważane będą jedynie przypadki gdzie $q \geq 0$, dzięki sprzężeniom wielomianów i momentów o przeciwnych indeksach harmonicznym: $V_{p,q} = \overline{V_{p,-q}}$, $M_{p,q} = \overline{M_{p,-q}}$.

Krokiem pomocniczym do dalszych przekształceń będzie wprowadzenie indeksów $2p + o$, $2q + o$, w miejsce poprzednich p, q , z dodatkową zmienną¹ $o \in \{0, 1\}$. Zapis ten wskazuje jawnie, czy moment jest parzysty, czy też nieparzysty ($o = 1$). Tym

¹litera o oznaczająca flagę nieparzystości ma się kojarzyć z angielskim słowem *odd*

samym, wzór wielomianu Zernike’a przyjmuje postać:

$$V_{2p+o,2q+o}(r, \theta) = \sum_{0 \leq s \leq p-q} \beta_{2p+o,2q+o,s} r^{2p+o-2s} e^{i(2q+o)\theta}, \quad (3.1)$$

gdzie $\beta_{2p+o,2q+o,s} = \frac{(-1)^s (2p+o-s)!}{s!(p+q+o-s)!(p-q-s)!}$.

Poniżej przedstawiony wzór (3.2) wyraża wielomian Zernike’a z odwrotną do oryginalnej kolejnością składników w sumie. Wzór ten otrzymano ze wzoru (3.1) poprzez podstawienie $s := p - s$ oraz liniowe przekształcenia na nierównościach stanowiących ograniczenia sumowania: pomnożenie przez -2 i dodanie $2p + o$ do obu nierówności.

$$\begin{aligned} V_{2p+o,2q+o}(r, \theta) &= \sum_{0 \leq p-s \leq p-q} \beta_{2p+o,2q+o,p-s} r^{2s+o} e^{i(2q+o)\theta} \\ &= \sum_{2q+o \leq 2s+o \leq 2p+o} \beta_{2p+o,2q+o,p-s} r^{2s+o} e^{i(2q+o)\theta}. \end{aligned} \quad (3.2)$$

Powyższy wzór będzie podstawą do dalszych wyprowadzeń zmierzających w kierunku stałoczasowych obliczeń momentów Zernike’a z wykorzystaniem obrazów całkowych.

3.1.2 Reprezentacja kartezjańska

Równoważnie do (2.9) momenty Zernike’a mogą zostać zapisane wykorzystując współrzędne kartezjańskie:

$$M_{p,q} = \frac{1}{2\pi a_p} \iint_{x^2+y^2 \leq 1} f(x, y) R_{p,q}(r(x, y)) F_{-q}(\theta(x, y)) dx dy, \quad (3.3)$$

gdzie promień i kąt (traktowane jako funkcje współrzędnych kartezjańskich x, y) mogą zostać przedstawione jako²:

$$r(x, y) = \sqrt{x^2 + y^2} = \sqrt{(x + iy)(x - iy)}, \quad (3.4)$$

$$\theta(x, y) = \operatorname{tg}^{-1} \frac{y}{x} = \frac{1}{2} i (\log(x - iy) - \log(x + iy)). \quad (3.5)$$

Dzięki zastosowaniu odwróconej kolejności reprezentacji (3.2), można wstawić (3.4) i (3.5) do wzoru (3.3), i sprowadzić zapis na momenty do następującej formy:

$$\begin{aligned} M_{2p+o,2q+o} &= \frac{1}{2\pi a_{2p+o}} \iint_{x^2+y^2 \leq 1} f(x, y) \sum_{2q+o \leq 2s+o \leq 2p+o} \beta_{2p+o,2q+o,p-s} \\ &\quad \cdot (x + iy)^{\frac{1}{2}(2s+o-2q-o)} \cdot (x - iy)^{\frac{1}{2}(2s+o+2q+o)} dx dy. \end{aligned} \quad (3.6)$$

²Tożsamość (3.5) działa analogicznie do programistycznych funkcji *atan2*, tzn. wyznacza arcus tangens z prawidłowym rozróżnianiem wszystkich ćwiartek układu współrzędnych.

Warto zwrócić uwagę, że wykładniki w (3.6) upraszczają się do nieujemnych liczb całkowitych i ostatecznie otrzymujemy:

$$M_{2p+o,2q+o} = \frac{1}{2\pi a_{2p+o}} \iint_{x^2+y^2 \leq 1} f(x,y) \sum_{2q+o \leq 2s+o \leq 2p+o} \beta_{2p+o,2q+o,p-s} \cdot (x+iy)^{s-q} \cdot (x-iy)^{s+q+o} dx dy. \quad (3.7)$$

Widać, że warunki sumowania $2q+o \leq 2s+o \leq 2p+o$ zapewniają, że $s-q \geq 0$. Obserwacja ta jest bardzo ważna ze względu na chęć późniejszego zastosowania obrazów całkowych do wsparcia obliczeń, co jest głównym celem niniejszej pracy.

W odróżnieniu od powyższych obserwacji, analogiczny wzór na OFMM przyjmuje postać:

$$O_{2p+o,2q+o} = \frac{1}{2\pi a_{2p+o}} \iint_{x^2+y^2 \leq 1} f(x,y) \sum_{0 \leq s \leq 2p+o} \alpha_{2p+o,s} \cdot (x+iy)^{\frac{1}{2}(s-2q-o)} \cdot (x-iy)^{\frac{1}{2}(s+2q+o)} dx dy. \quad (3.8)$$

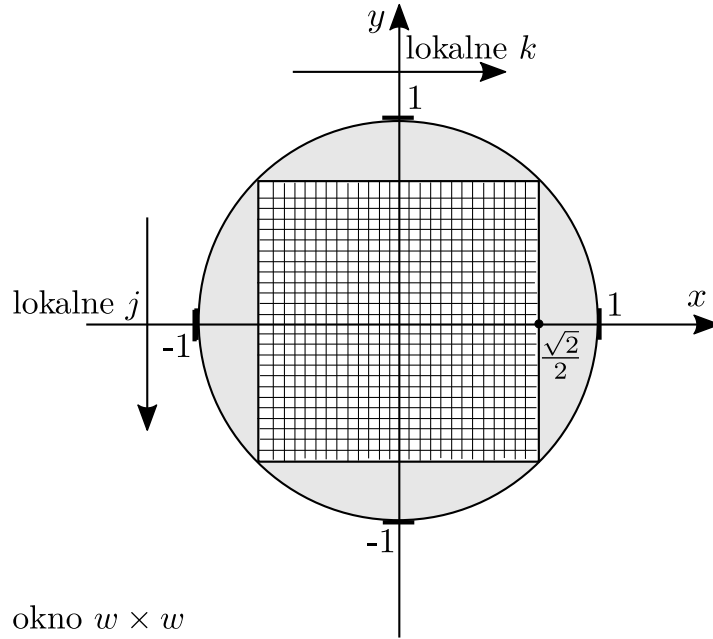
Warto zauważyć, że skoro indeks s startuje od zera i zmienia się o 1 (gdzie w (3.7) zmienia się o 2) to wykładników potęg nie można uprościć — mogą być one ujemne lub ułamkowe, dlatego też suma może zawierać ułamki lub pierwiastki kwadratowe.

Wybiegając nieco do przodu do kontekstu opracowywania obrazów całkowych, powyższe własności OFMM oznaczają, że punkty zaczepienia okna przesuwne (które są ukryte pod wartościami x, y) będą uwikłane w mianowniki lub pierwiastki kwadratowe we wzorze (3.8) i nie będzie możliwe, w ogólności, wyodrębnienie ich z wyrażenia. Z tego powodu, zapis oparty na obrazach całkowych nie jest możliwy dla OFMM. Powyższe argumenty staną się w pełni zrozumiałe w dalszej części rozdziału.

3.1.3 Wyznaczanie momentów Zernike’a dla fragmentu obrazu

W wielu zadaniach dotyczących przetwarzania obrazów, w szczególności detekcji, wygodniej jest wykonywać obliczenia na prostokątnym fragmencie obrazu niż na kole. Singh i Upneja (2012) zaproponowali zgrabny sposób, dedykowany początkowo dla OFMM, ale sprawdzający się także dla ZM, w którym kluczowa sztuczka polega na *wpisaniu* kwadratu³ o danym rozmiarze $w \times w$ (w pikselach) w koło jednostkowe. Sytuacja ta została przedstawiona na Rys. 3.1. Nad dopełnieniem kwadratu do koła „położone” są wartości zero, co matematycznie upraszcza całkowanie po kole do całkowania po kwadracie.

³Kwadrat ten w kontekście omawianego zadania — detekcji — reprezentuje okno przesuwne.



Rysunek 3.1: Sztuczka z pracy (Singh i Upneja, 2012) pozwalająca na obliczenie OFMM: kwadratowe okno obrazu wpisane w koło jednostkowe, wartości 0 leżą wewnątrz koła poza kwadratem. (Źródło: opracowanie własne)

Zastosowanie takiej sztuczki powoduje, że szerokość piksela odpowiada wartości⁴ $D = \sqrt{2}/w$, a jego pole powierzchni jest równe $D^2 = 2/w^2$. Iterując po indeksach pikseli, czyli dla $0 \leq j, k \leq w - 1$, można wygenerować współrzędne kartezjańskie odwzorowane na koło jednostkowe w następujący sposób:

$$x_k = \frac{2k - (w - 1)}{w\sqrt{2}}, \quad y_j = \frac{w - 1 - 2j}{w\sqrt{2}}. \quad (3.9)$$

Współrzędne te przyjmują kolejne wartości, poczynając od $-\sqrt{2}/2$ plus połowa szerokości piksela aż do $\sqrt{2}/2$ minus połowa szerokości piksela. Warto zauważyć, że parzyste w , zapewni, że współrzędne nigdy nie będą leżały na osiach układu tj. $x_k \neq 0$, $y_j \neq 0$. Pozwala to uniknąć kłopotów z funkcją tg^{-1} , które mogłyby wystąpić dla punktów leżących na osiach układu.

Należy wskazać, że w praktyce, rzadko oblicza się momenty poprzez dokładne całkowanie. W zadaniach przetwarzania obrazów stosuje się często aproksymację zerowego rzędu⁵, która do wyznaczania momentów jest wystarczająca:

⁴Wynika to z faktu, że koło jest jednostkowe. Warto zaznaczyć, że w różnych skalach okna przesuwne szerokość tego samego piksela będzie różna, tj. zależna od funkcji zmiennej w .

⁵Całka jest zastąpiona przez skończoną sumę, zakładającą funkcję kawałkami stałą nad każdym pikselem.

$$\begin{aligned}\widehat{M}_{2p+o,2q+o} &= \frac{1}{2\pi a_{2p+o}} \sum_{x_k^2+y_j^2 \leq 1} \sum f(x_k, y_j) R_{2p+o,2q+o}(r(x_k, y_j)) \cdot F_{-(2q+o)}(\theta(x_k, y_j)) D^2 \\ &= \frac{D^2}{2\pi a_{2p+o}} \sum_{\substack{0 \leq j \leq w-1 \\ 0 \leq k \leq w-1}} f(j, k) R_{2p+o,2q+o}(r(x_k, y_j)) F_{-(2q+o)}(\theta(x_k, y_j)) \approx M_{2p+o,2q+o}.\end{aligned}\tag{3.10}$$

Poprzez analogię do wzoru (3.7) można zapisać wersję wzoru (3.10) wyrażoną bezpośrednio we współrzędnych kartezjańskich:

$$\widehat{M}_{2p+o,2q+o} = \frac{D^2}{2\pi a_{2p+o}} \sum_{\substack{0 \leq j \leq w-1 \\ 0 \leq k \leq w-1}} f(j, k) \sum_{2q+o \leq 2s+o \leq 2p+o} \beta_{2p+o,2q+o,p-s} \cdot (x_k + iy_j)^{s-q} (x_k - iy_j)^{s+q+o}.\tag{3.11}$$

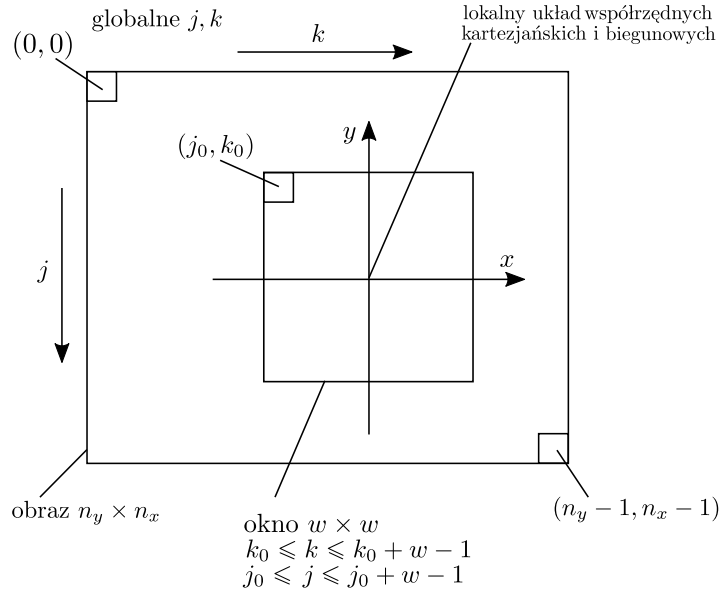
W kontekście złożoności obliczeniowej, formuła (3.11) jest liniowa ze względu na liczbę pikseli, a biorąc pod uwagę rząd momentu złożoność ta jest klasy $\Theta(w^2(p - q + 1))$.

Dalsza część pracy skupiać się będzie na wzorze (3.11) i zastosowaniu go do kluczowego celu, jakim jest detekcja obrotowo niezmiennicza bazująca na obrazach całkowych. Dla uproszczenia nazewnictwa, od tego momentu w pracy $\widehat{M}_{2p+o,2q+o}$ będą nazywane momentami Zernike’a, mimo że w rzeczywistości są one dyskretną aproksymacją dokładnych momentów $M_{2p+o,2q+o}$.

3.2 Zespolone obrazy całkowe i stałoczasowe obliczanie momentów Zernike’a podczas procedury detekcyjnej

Poniższa część pracy odchodzi od czysto matematycznego kontekstu momentów Zernike’a i skupia się na scenariuszu komputerowej procedury detekcyjnej.

Założmy, że obraz cyfrowy o rozmiarze $n_x \times n_y$ przemierzany jest oknem przesuwym o wymiarze $w \times w$, gdzie w jest parzyste (dla uproszczenia rozważony zostanie jedynie przypadek, gdy obraz jest przechodzony oknem jednego rozmiaru — jedna skala). Sytuacja ta została przedstawiona na Rys. 3.2. Niech (j, k) oznaczają globalne współrzędne piksela na obrazie. Dla każdego okna poddawanego analizie, niech jego punkt zaczepienia — górny lewy róg okna — będzie oznaczany jako (j_0, k_0) . W związku z tym, indeksy pikseli, które należą do danego okna to: $j_0 \leq j \leq j_0 + w - 1$, $k_0 \leq k \leq k_0 + w - 1$.



Rysunek 3.2: Ilustracja procedury detekcyjnej z wykorzystaniem okna przesuwanego. (Źródło: opracowanie własne)

Dodatkowo, wygodnym będzie wprowadzenie oznaczenia (j_c, k_c) dla tzw. *centralnego indeksu* okna:

$$j_c = \frac{1}{2}(2j_0 + w - 1), \quad k_c = \frac{1}{2}(2k_0 + w - 1). \quad (3.12)$$

Warto zauważyć, że ponieważ w jest parzyste, obie współrzędne j_c i k_c są ułamekami, tj. zawierają połówki.

Mając dany globalny indeks piksela (j, k) , lokalne kartezjańskie współrzędne odpowiadające mu, odwzorowane na koło jednostkowe z wpisanym w nie oknem, wynoszą:

$$x_k = \frac{2(k - k_0) - (w - 1)}{w\sqrt{2}} = \frac{\sqrt{2}}{w}(k - k_c), \quad (3.13)$$

$$y_j = \frac{(w - 1) - 2(j - j_0)}{w\sqrt{2}} = \frac{\sqrt{2}}{w}(j_c - j). \quad (3.14)$$

W poniższych akapitach przedstawiam kluczowe elementy potrzebne do stałoczasowego obliczania momentów Zernike’a w ramach procedury detekcyjnej. Rozpoczynam od wprowadzenia potrzebnych obrazów całkowych i zdefiniowania operatora przyrostu.

Niech $\{ii_{t,u}\}$ oznacza **zbiór zespolonych obrazów całkowych** skonstruowanych w następujący sposób:

$$ii_{t,u}(l, m) = \sum_{\substack{0 \leq j \leq l \\ 0 \leq k \leq m}} f(j, k)(k - ij)^t (k + ij)^u, \quad \begin{matrix} 0 \leq l \leq n_y - 1 \\ 0 \leq m \leq n_x - 1 \end{matrix}, \quad (3.15)$$

gdzie pary indeksów (t, u) , generujące zbiór, należą do:

$$\{(t, u): 0 \leq t \leq \lfloor \rho/2 \rfloor, 0 \leq u \leq \min(\rho - t, \lfloor (\rho + \varrho)/2 \rfloor)\}. \quad (3.16)$$

Indeksowanie zostanie dokładniej wytłumaczone w dalszej części pracy.

Dla dowolnego obrazu całkowego można zdefiniować operator przyrostu dla dowolnego prostokąta (okna) rozpiętego pomiędzy punktami (j_1, k_1) i (j_2, k_2) :

$$\Delta_{\substack{j_1, j_2 \\ k_1, k_2}}(ii_{t,u}) = ii_{t,u}(j_2, k_2) - ii_{t,u}(j_1-1, k_2) - ii_{t,u}(j_2, k_1-1) + ii_{t,u}(j_1-1, k_1-1). \quad (3.17)$$

Warto zauważyć, że operator ten zawiera trzy operacje na liczbach zespolonych: dwa odejmowania i jedno dodawanie.

Poniższa propozycja stanowi główny rezultat pracy.

Propozycja 1. *Załóżmy, że zestaw obrazów całkowych $\{ii_{t,u}\}$, zdefiniowanych zgodnie ze wzorem (3.15), został przygotowany przed rozpoczęciem procedury detekcyjnej. Wtedy, dla każdego kwadratowego okna obrazu każdy z momentów Zernike’a (3.11) może zostać obliczony w czasie stałym — $O(1)$ — niezależnie od liczby pikseli w oknie, w następujący sposób:*

$$\begin{aligned} \widehat{M}_{2p+o, 2q+o} &= \frac{1}{\pi a_{2p+o} w^2} \sum_{2q+o \leq 2s+o \leq 2p+o} \beta_{2p+o, 2q+o, p-s} \left(\frac{\sqrt{2}}{w}\right)^{2s+o} \\ &\cdot \sum_{t=0}^{s-q} \binom{s-q}{t} (-k_c + ij_c)^{s-q-t} \\ &\cdot \sum_{u=0}^{s+q+o} \binom{s+q+o}{u} (-k_c - ij_c)^{s+q+o-u} \Delta_{\substack{j_0, j_0+w-1 \\ k_0, k_0+w-1}}(ii_{t,u}), \end{aligned} \quad (3.18)$$

gdzie w oznacza szerokość, a (j_c, k_c) centralny indeks okna.

Dowód. Wyprowadzenie należy rozpocząć od wzoru (3.11). Dokonując podstawień $x_k = \frac{\sqrt{2}}{w}(k - k_c)$ i $y_j = \frac{\sqrt{2}}{w}(j_c - j)$, możliwe jest przejście z lokalnych współrzędnych piksela (na kole jednostkowym) do globalnych dla danego okna o środku w (j_c, k_c) .

Dodatkowo podstawmy $D^2 = 2/w^2$. W rezultacie otrzymujemy następujący wzór:

$$\begin{aligned} \widehat{M}_{2p+o, 2q+o} &= \frac{1}{2\pi a_{2p+o}} \frac{2}{w^2} \sum_{\substack{j_0 \leq j \leq j_0+w-1 \\ k_0 \leq k \leq k_0+w-1}} f(j, k) \\ &\cdot \sum_{2q+o \leq 2s+o \leq 2p+o} \beta_{2p+o, 2q+o, p-s} \left(\frac{\sqrt{2}}{w}(k - k_c) + i\frac{\sqrt{2}}{w}(j_c - j)\right)^{s-q} \\ &\cdot \left(\frac{\sqrt{2}}{w}(k - k_c) - i\frac{\sqrt{2}}{w}(j_c - j)\right)^{s+q+o} \\ &= \frac{1}{\pi a_{2p+o} w^2} \sum_{\substack{j_0 \leq j \leq j_0+w-1 \\ k_0 \leq k \leq k_0+w-1}} f(j, k) \sum_{2q+o \leq 2s+o \leq 2p+o} \beta_{2p+o, 2q+o, p-s} \\ &\cdot \left(\frac{\sqrt{2}}{w}\right)^{2s+o} (k - k_c + i(j_c - j))^{s-q} (k - k_c - i(j_c - j))^{s+q+o}. \end{aligned}$$

Następnie stosujemy wzór dwumianowy, rozróżniając dwie grupy wyrażeń: $-k_c \pm ij_c$ (niezależnych od globalnych współrzędnych piksela) oraz $k \mp ij$ (zależnych od nich). Ostatecznie, po zmianie kolejności sumowania, otrzymujemy:

$$\begin{aligned} \widehat{M}_{2p+o, 2q+o} &= \frac{1}{\pi a_{2p+o} w^2} \sum_{2q+o \leq 2s+o \leq 2p+o} \beta_{2p+o, 2q+o, p-s} \left(\frac{\sqrt{2}}{w} \right)^{2s+o} \\ &\quad \cdot \sum_{t=0}^{s-q} \binom{s-q}{t} (-k_c + ij_c)^{s-q-t} \\ &\quad \cdot \sum_{u=0}^{s+q+o} \binom{s+q+o}{u} (-k_c - ij_c)^{s+q+o-u} \\ &\quad \cdot \underbrace{\sum_{\substack{j_0 \leq j \leq j_0+w-1 \\ k_0 \leq k \leq k_0+w-1}} f(j, k) (k - ij)^t (k + ij)^u}_{\substack{\Delta \\ j_0, j_0+w-1 \\ k_0, k_0+w-1} (ii_{t,u})}. \end{aligned}$$

Dolna klamra w powyższym wzorze wskazuje, w jaki sposób kosztowne obliczeniowo sumowania po pikselach zostają zastąpione przez tanie — stałoczasowe — przyrosty obrazów całkowych. Tym samym całość obliczeń jest także złożoności $O(1)$ ze względu na liczbę pikseli w oknie obrazu. \square

Teraz można także wyjaśnić postać zapisanego wcześniej zbioru (3.16). Zawiera on tylko te pary indeksów (t, u) , które pojawiają się w wyprowadzonym wzorze (3.18) i wynikają bezpośrednio ze sposobu indeksowania sum.

Propozycja 1 została opublikowana w pracy (Bera, Klęsk i Sychel, 2019) w czasopiśmie *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

3.3 Własności

3.3.1 Zespolone sprzężenie obrazów całkowych

Własność 3. *Obrazy całkowe $ii_{t,u}$ oraz $ii_{u,t}$ są sprzężone we wszystkich punktach:*

$$\forall l, m \quad ii_{t,u}(l, m) = \overline{ii_{u,t}(l, m)}. \quad (3.19)$$

Dowód. Przyjmijmy $t \leq u$. Stosując podstawienie $d = u - t$, można zapisać $ii_{t,u}(l, m)$ jako:

$$\begin{aligned} ii_{t,u}(l, m) &= \sum_{j=0}^l \sum_{k=0}^m f(j, k) (k - ij)^t (k + ij)^{t+d} \\ &= \sum_{j=0}^l \sum_{k=0}^m f(j, k) (k^2 + j^2)^t (k + ij)^d. \end{aligned} \quad (3.20)$$

Korzystając ze wzoru dwumianowego, czynnik $(k + ij)^d$ możemy przedstawić w następujący sposób:

$$\sum_{\substack{0 \leq v \leq d \\ d-v \text{ parzyste}}} \binom{d}{v} k^v (ij)^{d-v} + \sum_{\substack{0 \leq v \leq d \\ d-v \text{ nieparzyste}}} \binom{d}{v} k^v (ij)^{d-v}. \quad (3.21)$$

Można zauważyć, że wynikiem pierwszej sumy jest liczba rzeczywista, ze względu na parzystą potęgę przy i , natomiast wynikiem drugiej sumy jest liczba zespolona, ze względu na nieparzystą potęgę przy i . Zatem, $\text{Im}(ii_{t,u}(l, m))$ jest równe:

$$\sum_{\substack{0 \leq j \leq l \\ 0 \leq k \leq m}} \sum f(j, k) (k^2 + j^2)^t \sum_{\substack{0 \leq v \leq d \\ d-v \text{ nieparzyste}}} \binom{d}{v} k^v (-1)^{(d-v-1)/2} j^{d-v}. \quad (3.22)$$

Łatwo sprawdzić, że dla zamienionych miejscami indeksów t, u , można przedstawić $ii_{u,t}(l, m)$ jako $\sum_{j=0}^l \sum_{k=0}^m f(j, k) (k^2 + j^2)^t (k - ij)^d$ i analogiczne rozwinięcie dwumianowe dla czynnika $(k - ij)^d$ prowadzi do $\text{Im}(ii_{u,t}(l, m)) = -\text{Im}(ii_{t,u}(l, m))$, podczas gdy części rzeczywiste pozostają równe w obu przypadkach. \square

Ważną konsekwencją algorytmiczną tej własności jest fakt, że potrzebnych jest mniej obrazów całkowych do realizacji głównej idei w Propozycji 1, jak opisano poniżej.

Konsekwencja 1. *Wystarczy przygotować obrazy całkowe $\{ii_{t,u}\}$ dla $t \leq u$. Dla $t > u$ zachodzi:*

$$\Delta_{\substack{j_0, j_0+w-1 \\ k_0, k_0+w-1}}(ii_{t,u}) = \overline{\Delta_{\substack{j_0, j_0+w-1 \\ k_0, k_0+w-1}}(ii_{u,t})}. \quad (3.23)$$

Stąd, jeśli w którymś momencie obliczeń, wzór (3.18) odwołuje się do obrazu całkowego $ii_{t,u}$ dla którego $t > u$, to należy sięgnąć po $ii_{u,t}$ i użyć sprzężenia jego przyrostu. Powoduje to dodanie jednej dodatkowej operacji (negacja części urojonej) w niektórych wyrazach, ale znacznie zmniejsza liczbę obrazów całkowych i czas na ich przygotowanie.

3.3.2 Dokładna liczba potrzebnych obrazów całkowych

Własność 4. *Korzystając z Konsekwencji 1, dokładna liczba obrazów całkowych potrzebna do obliczeń w ramach Propozycji 1 wynosi:*

$$\begin{aligned} \#\{ii_{t,u}\}_{t \leq u} &= \left\lfloor \frac{1}{2}\rho + 1 \right\rfloor^2 - \frac{1}{2} \left\lfloor \frac{1}{2}\rho + 1 \right\rfloor \left\lfloor \frac{1}{2}\rho \right\rfloor \\ &+ \begin{cases} \left\lfloor \frac{1}{2}\varrho \right\rfloor \left(\left\lfloor \frac{1}{2}\rho \right\rfloor - \frac{1}{2} \left\lfloor \frac{1}{2}\varrho \right\rfloor + \frac{1}{2} \right), & \text{dla parzystego } \rho; \\ \left\lfloor \frac{1}{2}(\varrho + 1) \right\rfloor \left(\left\lfloor \frac{1}{2}(\rho + 1) \right\rfloor - \frac{1}{2} \left\lfloor \frac{1}{2}(\varrho - 1) \right\rfloor \right), & \text{dla nieparzystego } \rho. \end{cases} \end{aligned} \quad (3.24)$$

Wyprowadzenie wzoru (3.24) bazuje na obserwacji, które pary (t, u) mogą wystąpić we wzorze (3.18). Dla przypomnienia, najbardziej zewnętrznymi ograniczeniami są $0 \leq p \leq \rho$ i $0 \leq 2q + o \leq \rho$. Gdy p oraz $2q + o$ są ustalone, pierwsza z sum ogranicza indeks s do $2q + o \leq 2s + o \leq 2p + o$. Wtedy, wewnętrzne sumy są zależne od bieżącego s , a tym samym indeksy (t, u) są ograniczone w następujący sposób: $0 \leq t \leq s - q$ i $0 \leq u \leq s + q + o$. Dodatkowo, wystarczy rozważyć tylko $t \leq u$ na mocy Konsekwencji 1.

Na początek rozważmy najmniejszy możliwy rząd harmoniczny $0 = 2q$ (parzysty). Stąd, $q = 0$ i $0 \leq 2s \leq \rho \Rightarrow 0 \leq s \leq \lfloor \frac{1}{2}\rho \rfloor$. Wtedy, liczba par (t, u) to:

$$\begin{aligned} \sum_{0 \leq t \leq \lfloor \frac{1}{2}\rho \rfloor} \sum_{t \leq u \leq \lfloor \frac{1}{2}\rho \rfloor} 1 &= \sum_{0 \leq t \leq \lfloor \frac{1}{2}\rho \rfloor} \left(\lfloor \frac{1}{2}\rho \rfloor - t + 1 \right) \\ &= \left\lfloor \frac{1}{2}\rho + 1 \right\rfloor^2 - \sum_{0 \leq t \leq \lfloor \frac{1}{2}\rho \rfloor} t = \left\lfloor \frac{1}{2}\rho + 1 \right\rfloor^2 - \frac{1}{2} \left\lfloor \frac{1}{2}\rho \right\rfloor \left\lfloor \frac{1}{2}\rho + 1 \right\rfloor. \end{aligned} \quad (3.25)$$

Warto zauważyć, że jest to pierwsza suma we wzorze (3.24). Następnie należy przeanalizować kolejne rzędy harmoniczne $1, 2, \dots$ i zobaczyć które dodatkowe pary (t, u) pojawią się (poza tymi odkrytymi do tej pory). Po przeanalizowaniu czterech przypadków, można będzie zauważyć pewną prawidłowość i zapisać ogólny wzór.

Przypadek $1 = 2q + o$ oznacza, że $q = 0, o = 1$ i ogranicza s do: $1 \leq 2s + 1 \leq \rho \Rightarrow 0 \leq s \leq \lfloor \frac{1}{2}(\rho - 1) \rfloor$. Dodatkowe pary (t, u) mogą wystąpić dla nowego maksymalnego u , to znaczy dla $u = \lfloor \frac{1}{2}(\rho - 1) \rfloor + q + o = \lfloor \frac{1}{2}(\rho - 1) \rfloor + 0 + 1$, pod warunkiem, że ρ jest nieparzyste. W przeciwnym razie, maksymalne u pozostaje takie samo jak w poprzednim przypadku $0 = 2q$. Stąd, liczba nowych obrazów całkowych (dla nieparzystego ρ) jest równa:

$$\sum_{0 \leq t \leq \lfloor \frac{1}{2}(\rho - 1) \rfloor - \underbrace{0}_q} 1 = \left\lfloor \frac{1}{2}(\rho - 1) \right\rfloor + 1 = \left\lfloor \frac{1}{2}\rho + \frac{1}{2} \right\rfloor. \quad (3.26)$$

Przypadek $2 = 2q + o$ oznacza, że $q = 1, o = 0$ i ogranicza s do: $2 \leq 2s \leq \rho \Rightarrow 1 \leq s \leq \lfloor \frac{1}{2}\rho \rfloor$. Dodatkowe pary (t, u) pojawiają się dla nowego maksymalnego u , to znaczy $u = \lfloor \frac{1}{2}(\rho - 1) \rfloor + q + o = \lfloor \frac{1}{2}(\rho - 1) \rfloor + 1 + 0$, pod warunkiem, że ρ jest parzyste. Stąd, liczba nowych obrazów całkowych (dla parzystego ρ) jest równa:

$$\sum_{0 \leq t \leq \lfloor \frac{1}{2}\rho \rfloor - \underbrace{1}_q} 1 = \left\lfloor \frac{1}{2}\rho \right\rfloor. \quad (3.27)$$

Przypadek $3 = 2q + o$ oznacza, że $q = 1, o = 1$ i ogranicza s do: $3 \leq 2s + 1 \leq \rho \Rightarrow 1 \leq s \leq \lfloor \frac{1}{2}(\rho - 1) \rfloor$. Dodatkowe pary (t, u) pojawiają się dla nowego maksymalnego u , to znaczy $u = \lfloor \frac{1}{2}(\rho - 1) \rfloor + q + o = \lfloor \frac{1}{2}(\rho - 1) \rfloor + 1 + 1$, pod warunkiem, że ρ jest nieparzyste.

Stąd, liczba nowych obrazów całkowych (dla nieparzystego ρ) jest równa:

$$\sum_{0 \leq t \leq \lfloor \frac{1}{2}(\rho-1) \rfloor - \underbrace{1}_q} 1 = \left\lfloor \frac{1}{2}\rho - \frac{1}{2} \right\rfloor. \quad (3.28)$$

Przypadek $4 = 2q + o$ oznacza, że $q = 2$, $o = 0$ i ogranicza s do: $4 \leq 2s \leq \rho \Rightarrow 2 \leq s \leq \lfloor \frac{1}{2}\rho \rfloor$. Dodatkowe pary (t, u) pojawiają się dla nowego maksymalnego u , to znaczy $u = \lfloor \frac{1}{2}(\rho - 1) \rfloor + q + o = \lfloor \frac{1}{2}(\rho - 1) \rfloor + 2 + 0$, pod warunkiem, że ρ jest parzyste. Stąd, liczba nowych obrazów całkowych (dla parzystego ρ) jest równa:

$$\sum_{0 \leq t \leq \lfloor \frac{1}{2}\rho \rfloor - \underbrace{2}_q} 1 = \left\lfloor \frac{1}{2}\rho - 1 \right\rfloor. \quad (3.29)$$

Na koniec, można zapisać ostateczną formułę. Gdy, ρ jest parzyste, liczba dodatkowych obrazów całkowych wynosi:

$$\begin{aligned} \sum_{2 \leq 2q \leq \varrho} \sum_{0 \leq t \leq \lfloor \frac{1}{2}\rho \rfloor - q} 1 &= \sum_{1 \leq q \leq \lfloor \frac{1}{2}\varrho \rfloor} \left(\left\lfloor \frac{1}{2}\rho \right\rfloor - q + 1 \right) \\ &= \left\lfloor \frac{1}{2}\varrho \right\rfloor \left(\left\lfloor \frac{1}{2}\rho \right\rfloor - \frac{1}{2} \left\lfloor \frac{1}{2}\varrho \right\rfloor + \frac{1}{2} \right) \end{aligned} \quad (3.30)$$

natomiast, gdy ρ jest nieparzyste, liczba dodatkowych obrazów całkowych wynosi:

$$\begin{aligned} \sum_{2 \leq 2q+1 \leq \varrho} \sum_{0 \leq t \leq \lfloor \frac{1}{2}(\rho-1) \rfloor - q} 1 &= \sum_{1 \leq q \leq \lfloor \frac{1}{2}(\varrho-1) \rfloor} \left(\left\lfloor \frac{1}{2}(\rho-1) \right\rfloor - q + 1 \right) \\ &= \left\lfloor \frac{1}{2}(\varrho+1) \right\rfloor \left(\left\lfloor \frac{1}{2}(\rho+1) \right\rfloor - \frac{1}{2} \left\lfloor \frac{1}{2}(\varrho-1) \right\rfloor \right). \end{aligned} \quad (3.31)$$

Łącząc podstawowy wzór (3.25) z przypadkami (3.30), (3.31) możliwe jest zapisanie ostatecznej formuły (3.24).

Łatwo sprawdzić, że gdy funkcje zaokrąglania w dół (podłoga) w (3.24) zostałyby zaniedbane, to liczba obrazów całkowych jest w przybliżeniu proporcjonalna do $\frac{1}{8}\rho^2 + \frac{1}{4}\rho\varrho - \frac{1}{8}\varrho^2$. Tablica 3.1 prezentuje liczbę potrzebnych obrazów całkowych dla początkowych wartości ρ i ϱ .

3.3.3 Liczba operacji przyrostu

Jak już wspomniano, obliczanie momentu zgodnie z Propozycją 1 wymaga stałej liczby operacji, niezależnie od liczby pikseli. Niemniej, dobrze byłoby wiedzieć, że ta stała liczba operacji jest rozsądnie nieduża, co zostanie sprawdzone w tym punkcie.

Uprośmy analizę do zliczania jedynie wywołań operatora Δ znajdujących się na najbardziej wewnętrznym poziomie sumowań we wzorze (3.18). Liczba tych wy-

Tablica 3.1: Liczba obrazów całkowych zgodnie ze wzorem (3.24).

$\rho \backslash \varrho$	0	1	2	3	4	5	6	7	8	9	10	11	12
0	1												
1	1	2											
2	3	3	4										
3	3	5	5	6									
4	6	6	8	8	9								
5	6	9	9	11	11	12							
6	10	12	13	13	15	15	16						
7	10	14	14	17	17	19	19	20					
8	15	15	19	19	22	22	24	24	25				
9	15	20	20	24	24	27	27	29	29	30			
10	21	21	26	26	30	30	33	33	35	35	36		
11	21	27	27	32	32	36	36	39	39	41	41	42	
12	28	28	34	34	39	39	43	43	46	46	48	48	49

wołań wynosi:

$$\begin{aligned}
 & \sum_{q \leq s \leq p} (s - q + 1)(s + q + o + 1) \\
 &= \begin{cases} \frac{1}{3}p^3 - pq^2 + \frac{2}{3}q^3 + \frac{3}{2}p^2 - \frac{3}{2}q^2 + \frac{13}{6}p - \frac{1}{6}q + 1, & \text{dla } o = 0; \\ \frac{1}{3}p^3 - pq^2 + \frac{2}{3}q^3 + 2p^2 - pq - q^2 + \frac{11}{3}p - \frac{5}{3}q + 2, & \text{dla } o = 1. \end{cases} \quad (3.32)
 \end{aligned}$$

Trzy najważniejsze termny wskazują na to, że liczba operacji jest rzędu $\frac{1}{3}p^3 - pq^2 + \frac{2}{3}q^3 + \Theta(p^2, q^2, pq)$.

Tablica 3.2 prezentuje przykłady dokładnej liczby operatorów przyrostu dla różnych nastaw $2p+o$ i $2q+o$. Liczby pokazane w tablicy warto spróbować skonfrontować z obliczeniami opartymi na wzorze definicyjnym (3.11). Dla przypomnienia jego złożoność to $\Theta(w^2(p - q + 1))$. Dla przykładu, rozważmy okno detekcyjne o wymiarze $w = 64$ oraz $2p + o = 10$ i $2q + o = 4$ jako założone rzędy. Wtedy, liczba sumowań we wzorze (3.11) wynosi w przybliżeniu $\approx 1.6 \cdot 10^4$. Propozycja 1 jest wyraźnie szybsza.

Należy być świadomym, że oprócz operatorów przyrostu istnieją inne pomocnicze operacje we wzorze (3.18), mianowicie: współczynniki dwumianowe, potęgi wyrażeń $-k_c \pm ij_c$, mnożenia (zespolone oraz rzeczywistoliczbowe) oraz dodawania. Warto zauważyć, że wiele z nich jest mniej ważna, ponieważ występują w zewnętrznych sumach i w związku z tym są wywoływane rzadziej. Co więcej, jak zostanie pokazane w dalszej części pracy, wiele z nich da się przyspieszyć poprzez *tablicowanie*.

Tablica 3.2: Liczba operacji przyrostu Δ biorących udział w obliczeniach wartości wzoru (3.18).

$2q + o$	$2p + o$	0	1	2	3	4	5	6	7	8	9	10	11	12
0	0	1												
1	1		2											
2	2	5		3										
3	3		8		4									
4	4	14		11		5								
5	5		20		14		6							
6	6	30		26		17		7						
7	7		40		32		20		8					
8	8	55		50		38		23		9				
9	9		70		60		44		26		10			
10	10	91		85		70		50		29		11		
11	11		112		100		80		56		32		12	
12	12	140		133		115		90		62		35		13

3.3.4 Liczba cech

Rozważmy zbiór cech, oznaczonych jako $\phi_{2p+o,2q+o}$, zawierający moduły momentów określonych wzorem (3.11):

$$\left\{ \phi_{2p+o,2q+o} : 0 \leq 2p+o \leq \rho, 0 \leq 2q+o \leq \min(2p+o, \varrho) \right\}, \quad (3.33)$$

gdzie $\phi_{2p+o,2q+o} = |\widehat{M}_{2p+o,2q+o}|$. Można sprawdzić (przez uważne zliczanie), że całkowita liczba takich cech jest równa:

$$\begin{aligned} \#\left\{ \phi_{2p+o,2q+o} \right\} &= (2\lfloor \rho/2 \rfloor - \lfloor \varrho/2 \rfloor + 3) \lfloor \varrho/2 \rfloor + (1 - \varrho \bmod 2) \\ &\quad \cdot (\lfloor \rho/2 \rfloor - \lfloor \varrho/2 \rfloor + 1) - (1 - \rho \bmod 2) \lfloor \varrho/2 \rfloor. \end{aligned} \quad (3.34)$$

3.4 Wariant pomysłu podstawowego

Jedną z intencji Violi i Jonesa, w ich podejściu bazującym na cechach Haara (Viola i Jones, 2001), było wygenerowanie dużej liczby cech np. rzędu 10^4 , 10^5 . Postępując w ten sposób, oczekuje się, że algorytm uczący będzie w stanie dzięki mnogości cech wybrać taki ich podzbiór, który dobrze opisuje poszukiwane na obrazach obiekty.

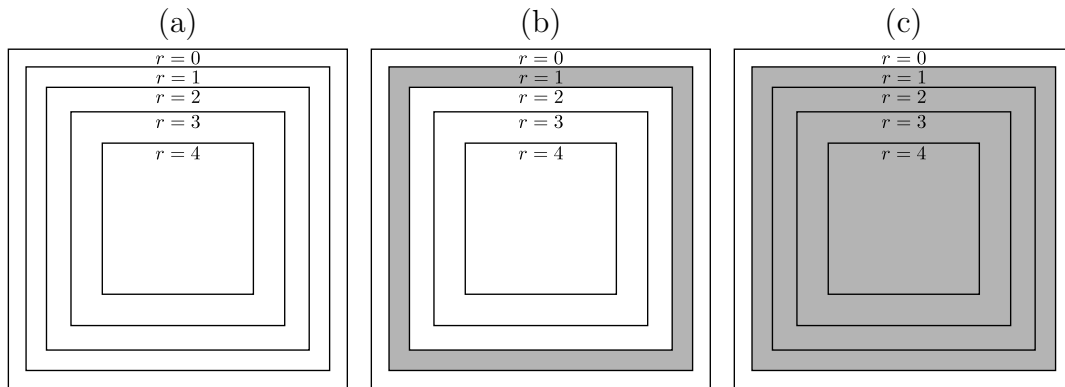
Patrząc na wzór (3.34), można zauważyć, że liczba cech w proponowanym dotychczas podejściu jest stosunkowo niewielka tj. dla $\rho = \varrho = 10$ otrzymuje się 36 cech.

Dlatego, podążając za motywacją Violi i Jonesa, poniżej zaproponowana została pewna odmiana głównej idei z Propozycji 1. Dzięki niej można wygenerować więcej cech poprzez podział analizowanego okna na określoną liczbę współśrodkowych kwadratów, które będą nazywane dalej *pierścieniami*. Cechy mogą zostać wtedy wykstrahowane z pierścieni, zamiast z całego okna.

Niech parametr R oznacza przyjętą liczbę pierścieni. Rys. 3.3 ilustruje pomysł dla $R = 5$ w dwóch możliwych wariantach:

- wariacie naturalnym (z dziurami), gdzie dziedziny całkowania (sumowania) wzajemnie się wykluczają — pierścienie zawierają ‘dziury’,
- wariacie bez dziur, gdzie kolejne dziedziny są podzbiorami poprzednich.

Dla ustalonego indeksu pierścienia $r \in \{0, 1, \dots, R - 1\}$, można rozróżnić dwa warianty wprowadzając flagę h , gdzie $h = 1$ określa pierwszy wariant (istnieje ‘dziura’) i $h = 0$ drugi (‘dziura’ nie istnieje). Warto zauważyć, że dla najmniejszego indeksu pierścienia $r = R - 1$, jedynie przypadek dla $h = 0$ jest możliwy.



Rysunek 3.3: Koncept pierścieni: (a) podział okna dla $R = 5$, (b) przykład dziedziny całkowania dla $r = 1$ i $h = 1$ (wariant z ‘dziurą’), (c) przykład dziedziny całkowania dla $r = 1$ i $h = 0$ (wariant bez ‘dziury’). (Źródło: opracowanie własne)

Aby utrzymać dziedziny całkowania w przybliżeniu równe co do rozmiaru dla przypadku $h = 1$, należy wymusić następujące warunki. Załóżmy, że szerokość najbardziej zewnętrznego pierścienia w_0 jest równa szerokości okna: $w_0 = w$. Idealne szerokości w_r ($r = 1, \dots, R - 1$) pozostałych pierścieni mogą zostać tak wybrane, aby pola powierzchni pierścieni, nad którymi całkujemy (szare na Rys. 3.3b) były równe — czyli spełniające warunek:

$$w_r^2 - w_{r+1}^2 = \frac{1}{R} w^2. \quad (3.35)$$

Oznacza to, że wzór na w_r będzie następujący:

$$w_r = w \sqrt{1 - r/R}. \quad (3.36)$$

Oczywiście, idealne szerokości nie mogą zostać użyte. Po pierwsze, mamy do czynienia z dyskretną siatką pikseli. Po drugie, wymagane są parzyste szerokości, co zostało opisane już wcześniej w pracy. W związku z tym, ostateczne szerokości pierścieni są zaokrąglane w górę do najbliższej liczby całkowitej:

$$v_r = \text{round}(w_r) + (\text{round}(w_r) \bmod 2). \quad (3.37)$$

Aby móc obliczyć operator przyrostu dla pierścieni, potrzebne są współrzędne lewego górnego wierzchołka każdego pierścienia (j_r, k_r) . Mogą one zostać przedstawione na dwa sposoby:

$$\begin{aligned} j_r &= j_0 + \frac{1}{2}(w - v_r) = j_c + \frac{1}{2} - \frac{1}{2}v_r, \\ k_r &= k_0 + \frac{1}{2}(w - v_r) = k_c + \frac{1}{2} - \frac{1}{2}v_r. \end{aligned} \quad (3.38)$$

Możliwe jest teraz zdefiniowanie nowego zbioru cech:

$$\left\{ \phi_{2p+o, 2q+o, r, h}: 0 \leq 2p+o \leq \rho, 0 \leq 2q+o \leq \min(2p+o, \varrho), 0 \leq r < R, h \in \{0, 1\} \right\}, \quad (3.39)$$

gdzie $\phi_{2p+o, 2q+o, r, h} = \left| \widehat{M}_{2p+o, 2q+o, r, h} \right|$ i

$$\begin{aligned} \widehat{M}_{2p+o, 2q+o, r, h} &= \frac{D^2}{2\pi a_p} \left(\sum_{\substack{j_r \leq j \leq j_r + v_r - 1 \\ k_r \leq k \leq k_r + v_r - 1}} f(j, k) \sum_{2q+o \leq 2s+o \leq 2p+o} \beta_{2p+o, 2q+o, p-s} \right. \\ &\quad \cdot (x_k + iy_j)^{s-q} (x_k - iy_j)^{s+q+o} - h \cdot \sum_{\substack{j_{r+1} \leq j \leq j_{r+1} + v_{r+1} - 1 \\ k_{r+1} \leq k \leq k_{r+1} + v_{r+1} - 1}} f(j, k) \\ &\quad \left. \cdot \sum_{2q+o \leq 2s+o \leq 2p+o} \beta_{2p+o, 2q+o, p-s} (x_k + iy_j)^{s-q} (x_k - iy_j)^{s+q+o} \right). \end{aligned}$$

Liczbę takich cech można powiązać ze wzorem (3.34), to znaczy z liczbą cech bez pierścieni, i wynosi ona:

$$(2R - 1) \cdot \#\left\{ \phi_{2p+o, 2q+o} \right\}. \quad (3.40)$$

Warto zauważyć dwie wady proponowanych zmian. Po pierwsze, bardziej odpowiednie byłyby kołowe, a nie kwadratowe fragmenty podziałowe. Informacje zawarte w kwadratowych fragmentach są niezmiennicze tylko dla obrotów o kąt 90° , 180° , i 270° . Po drugie, włączenie wariantu bez ‘dziur’ ($h = 0$) wprowadza pewną nadmiarowość wśród cech, mianowicie:

$$M_{2p+o, 2q+o, r, 1} = M_{2p+o, 2q+o, r, 0} - M_{2p+o, 2q+o, r+1, 0}, \quad (3.41)$$

co ma wpływ na korelację cech $\phi_{2p+o, 2q+o, r, 1}$ z cechami $\phi_{2p+o, 2q+o, r, 0}$ oraz $\phi_{2p+o, 2q+o, r+1, 0}$. Dlatego, niektóre cechy można obliczyć bazując na innych. Warto jednak przypomnieć, że główną motywacją jest generowanie większej liczby cech i zapewnienie algorytmowi uczącemu więcej informacji.

3.5 Przyspieszenie obliczeń poprzez tablicowanie

W tej sekcji skupiono się na implementacyjnych aspektach Propozycji 1 i wskazaniu miejsc, w których obliczenia mogą zostać przyspieszone.

3.5.1 Tablica dla wyrażeń zawierających indeks centralny

W celu zredukowania liczby obliczeń we wzorze (3.18) dotyczących potęg wyrażeń zespolonych powiązanych z *indeksem centralnym*, można przygotować tablicę z gotowymi wartościami tychże potęg. Tablica ta będzie oznaczana jako L^c (od ang. *look up table*). Po jej odpowiednim przygotowaniu, tablica może być wczytywana (np. z pliku binarnego) w chwili uruchamiania programu, a później używana podczas obliczeń. Poszczególne elementy tablicy L^c są określone następująco:

$$L^c[k_c - 0.5, j_c - 0.5, e_1, e_2] := (-k_c + ij_c)^{e_1} (-k_c - ij_c)^{e_2}, \quad (3.42)$$

gdzie naturalne wykładniki przyjmują wartości z zakresów:

$$0 \leq e_1 \leq \lfloor (\rho + \varrho)/2 \rfloor, \quad 0 \leq e_2 \leq \lfloor (\rho + \varrho)/2 \rfloor + 1,$$

a indeksy centralne są przygotowane dla pewnego z góry założonego maksymalnego rozmiaru obrazu $n_{x,\max} \times n_{y,\max}$:

$$0 \leq k_c - 1/2 \leq n_{x,\max} - 2, \quad 0 \leq j_c - 1/2 \leq n_{y,\max} - 2.$$

Podczas odwoływania się do elementów tablicy L^c należy wziąć pod uwagę korektę o -0.5 , która związana jest z faktem, że indeks centralny zawiera połówki i reprezentuje punkt styku czterech pikseli.

W czasie wykonania programu, rzeczywista implementacja odpytuje tablicę o gotowe wartości dla następujących indeksów:

$$L^c[k_c - 0.5, j_c - 0.5, s - q - t, s + q + o - u],$$

zgodnie z wnętrzem wzoru (3.18).

3.5.2 Tablica na wyniki zwracane przez operator przyrostu

Warto zauważyć, że dla ustalonej pozycji okna, przy obliczaniu różnych momentów, występuje wiele tych samych wyników operatora przyrostu (tj. wyniki dla tych samych par t, u). Wyniki te mogą być zapamiętywane i używane ponownie. Dla

tych potrzeb, można przygotować tablicę nazwaną L^Δ , z elementami zdefiniowanymi w następujący sposób:

$$L^\Delta[r, h, t, u] := \Delta_{\substack{j_r, j_r+v_r-1 \\ k_r, k_r+v_r-1}}(ii_{t,u}) - h \cdot \Delta_{\substack{j_{r+1}, j_{r+1}+v_{r+1}-1 \\ k_{r+1}, k_{r+1}+v_{r+1}-1}}(ii_{t,u}), \quad (3.43)$$

gdzie: $0 \leq r \leq R-1$, $h \in \{0, 1\}$, $0 \leq t \leq \lfloor \rho/2 \rfloor$, $0 \leq u \leq \lfloor (\rho + \varrho)/2 \rfloor$.

Elementy tej tablicy *nie* są obliczane przed procedurą detekcyjną. Mogą być zainicjalizowane zerami zespolonymi i będą wypełniane stopniowo podczas procedury ekstrakcji cech dla ustalonej pozycji okna przesuwne.

3.5.3 Algorytm ekstrakcji cech wykorzystujący tablicowanie

Oprócz dwóch opisanych wcześniej tablic L^c , L^Δ można wprowadzić jeszcze trzy dodatkowe przechowujące: współczynniki dwumianowe, współczynniki wielomianów Zernike’a oraz stałe normalizujące w następujący sposób:

$$\begin{aligned} L^b[n, k] &:= \binom{n}{k}, \\ L^\beta[p, q, s] &:= \beta_{p,q,s} = \frac{(-1)^s (p-s)!}{s!((p+q)/2-s)!((p-q)/2-s)!}, \\ L^a[p] &:= a_p = 1/(2(p+1)), \end{aligned} \quad (3.44)$$

gdzie: $0 \leq n \leq \lfloor (\rho + \varrho)/2 \rfloor + 1$, $0 \leq k \leq n$, $0 \leq p \leq \rho$, $0 \leq q \leq \min(p, \varrho)$, $p-q$ jest parzyste, $0 \leq s \leq (p-q)/2$. Te wstępnie przygotowane tablice mogą zostać załadowane w momencie uruchamiania programu.

Założmy, że $\mathcal{I} = \{(2p+o, 2q+o, r, h)\}$ reprezentuje dowolny podzbiór indeksów cech bazujących na momentach Zernike’a zdefiniowanych zgodnie z zapisem (3.39). Może to być pełen zbiór, odpowiedni gdy chce się obliczać wszystkie cechy podczas nauki detektora, lub mniejszy podzbiór, gdy używane są jedynie niektóre cechy wybrane przez algorytm uczący (tj. boosting).

Algorytmy 4 oraz 5 prezentują pseudokody procedury ekstrakcji cech zdefiniowanych przez zbiór \mathcal{I} dla określonego okna o rozmiarze $w \times w$ zaczepionego w punkcie (j_0, k_0) . Algorytmy te wykorzystują Propozycję 1, Własność 3 (sprzężenie zespolone obrazów całkowych) oraz omówione wcześniej tablice pozwalające na redukcję obliczeń.

Warto wyjaśnić trzy szczegóły występujące w przedstawionych algorytmach. Pierwszy dotyczy wielomianowych i harmonicznnych indeksów stojących przy momentach. Do ich reprezentacji najpierw używane są zmienne p_{expr} , q_{expr} — odzwierciedlają one lepiej warunki implementacyjne. Później są one zamieniane na $2p+o$ oraz $2q+o$. Drugi szczegół dotyczy pomocniczych sekwencji (j_r) , (k_r) , (v_r) związanych

Algorytm 4 Ekstrakcja cech opartych na ZM z użyciem tablic.

```

procedure EXTRACTFEATURES( $\{ii_{t,u}\}, \mathcal{I}, j_0, k_0, w$ )
    Przygotuj tablicę  $L^\Delta$  wypełnioną zerami zespolonymi.
    Przygotuj trójki  $(j_r), (k_r), (v_r)$  opisujące pierścienie dla danego okna
        zgodnie z (3.38), (3.37).
    for  $(p_{\text{expr}}, q_{\text{expr}}, r, h) \in \mathcal{I}$  do
        Oblicz i zapamiętaj  $\phi_{p_{\text{expr}}, q_{\text{expr}}, r, h}$  poprzez wywołanie
            SINGLEFEATURE( $\{ii_{t,u}\}, p_{\text{expr}}, q_{\text{expr}}, r, h, L^\Delta, (j_r), (k_r), (v_r)$ ).
    end for
    return  $\{\phi_{p_{\text{expr}}, q_{\text{expr}}, r, h}\}$ 
end procedure

```

z pierścieniami dla danej pozycji okna. Dla jasności zapisu sekwencje te są obliczane jawnie na poziomie zewnętrznej procedury, chociaż w rzeczywistości mogą być wstępnie wyliczone jeszcze przed procedurą detekcyjną, gdy wiadomo jakie szerokości w okna przesuwanego (tj. jak wiele skal) będą brały udział w procedurze. Trzeci szczegół dotyczy tablic. Jedynie tablica L^Δ (wyniki operatora przyrostu) jest tworzona i przekazywana jako argument, z kolei można założyć, że tablice L^c, L^β, L^a zostały przygotowane wcześniej i są dostępne globalnie.

Algorytm 5 Ekstrakcja pojedynczej cechy opartej na ZM z użyciem tablic.

```

procedure SINGLEFEATURE( $\{ii_{t,u}\}$ ,  $p_{\text{expr}}$ ,  $q_{\text{expr}}$ ,  $r$ ,  $h$ ,  $L^\Delta$ ,  $(j_r)$ ,  $(k_r)$ ,  $(v_r)$ )
   $o := p_{\text{expr}} \bmod 2$ ,  $p := (p_{\text{expr}} - o)/2$ ,  $q := (q_{\text{expr}} - o)/2$ 
   $\widehat{M}_{2p+o, 2q+o, r, h} := 0.0 + i 0.0$ 
  for  $s := q, \dots, p$  do
     $s_t := 0.0 + i 0.0$ 
    for  $t := 0, \dots, s - q$  do
       $s_u := 0.0 + i 0.0$ 
      for  $u := 0, \dots, s + q + o$  do
        if  $L^\Delta[r, h, t, u] = 0.0 + i 0.0$  then
          if  $t \leq u$  then
             $\delta := \Delta_{\substack{j_r, j_r+v_r-1 \\ k_r, k_r+v_r-1}}(ii_{t,u})$ 
            if  $h = 1$  then
               $\delta := \delta - \Delta_{\substack{j_{r+1}, j_{r+1}+v_{r+1}-1 \\ k_{r+1}, k_{r+1}+v_{r+1}-1}}(ii_{t,u})$ 
            end if
          else
             $\delta := \Delta_{\substack{j_r, j_r+v_r-1 \\ k_r, k_r+v_r-1}}(ii_{u,t})$ 
            if  $h = 1$  then
               $\delta := \delta - \Delta_{\substack{j_{r+1}, j_{r+1}+v_{r+1}-1 \\ k_{r+1}, k_{r+1}+v_{r+1}-1}}(ii_{u,t})$ 
            end if
             $\delta := \bar{\delta}$ 
          end if
           $L^\Delta[r, h, t, u] := \delta$ 
        end if
         $s_u := s_u + L^b[s + q + o, u]$ 
           $\cdot L^c[k_c - 0.5, j_c - 0.5, s - q - t, s + q + o - u]$ 
           $\cdot L^\Delta[r, h, t, u]$ 
      end for
       $s_t := s_t + L^b[s - q, t] \cdot s_u$ 
    end for
     $\widehat{M}_{2p+o, 2q+o, r, h} := \widehat{M}_{2p+o, 2q+o, r, h}$ 
       $+ L^\beta[2p + o, 2q + o, p - s] \cdot (\sqrt{2}/w)^{2s+o} \cdot s_t$ 
  end for
   $\widehat{M}_{2p+o, 2q+o, r, h} := \widehat{M}_{2p+o, 2q+o, r, h} / (\pi \cdot L^a[2p + o] \cdot w^2)$ 
  return  $|\widehat{M}_{2p+o, 2q+o, r, h}|$ 
end procedure

```

Rozdział 4

Rezultaty pierwszych eksperymentów

Rozdział przedstawia pierwsze eksperymenty przeprowadzone na dwóch zbiorach danych.

We wszystkich przedstawionych poniżej eksperymentach wykorzystany zostanie algorytm uczący *RealBoost* (RB), który uczy zespół słabych klasyfikatorów — płytkie *drzewa decyzyjne* (RB+DT) (Bera, Klęsk i Sychel, 2019; Friedman, Hastie i Tibshirani, 2000; Rasolzadeh, Petersson i Pettersson, 2006). Rozmiar zespołu oznaczany będzie poprzez T . W opisanych eksperymentach $T = 512$, liczba koszy $B = 64$, a głębokość drzew będzie równa 6, co przekłada się maksymalnie na 32 terminale (liście) w drzewie (opis parametrów można znaleźć w Rozdziale 1). Wykorzystana zostanie podstawowa przestrzeń cech bazująca na modułach momentów Zernike’a (ZM-M) (3.18), a liczba cech będzie zapisywana w nawiasach. Stąd zapis np. ZM-M (375) określać będzie zbiór modułów momentów Zernike’a zawierający 375 cech.

Warto nadmienić, że w eksperymentach użyto indeksu Jaccarda (stosunek powierzchni części wspólnej do powierzchni sumy dwóch okien) w dwóch miejscach:

- przy przetwarzaniu wykrytych okien w celu ich grupowania,
- przy sprawdzaniu poprawności wskazań detektora względem rzeczywistych położenia wyszukiwanego obiektu.

Najczęściej wynikiem działania detektora jest zestaw wielu okien z pozytywnymi wskazaniami skupionych wokół każdego celu. Wynika to z faktu, że ten sam obiekt może zostać wykryty wielokrotnie pomimo drobnych różnic w pozycji lub rozmiarze okna przesuwnego. Dzięki indeksowi Jaccarda można podjąć decyzję, które z tych okien powinny zostać ze sobą zgrupowane. W tym celu przygotowano krótką procedurę, która pozwala zdecydować, które z okien powinny zostać ze sobą połączone

— są to okna, których indeks Jaccarda jest największy i jednocześnie nie mniejszy niż założony, w poniższych eksperymentach przyjęto wartość 0.4.

W późniejszym czasie, pozytywne wskazania klasyfikatora są porównywane względem rzeczywistych położeń obiektów (zapisanych w plikach) — czyli względem tzw. *ground truth*. W poniższych eksperymentach przyjęto, że jako poprawne wskazania uznaje się te, których indeks wynosi przynajmniej 0.5. W przypadku niespełnienia tego warunku, okno uznawane jest za fałszywy alarm.

W poniższych eksperymentach przygotowano trzy zbiory danych:

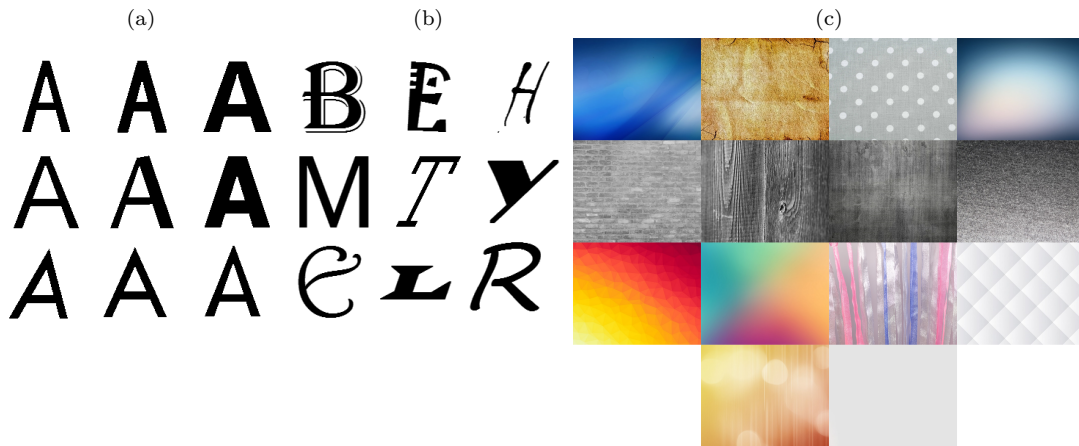
- dane uczące — zbiór zawierający gotowe okna z próbkami pozytywnymi oraz negatywnymi; zbiór służy do nauki klasyfikatorów,
- dane walidacyjne — zbiór zawierający obrazy, na których znajdują się różne obiekty, w tym te, które należy wykryć; zbiór służy do wyboru progów decyzyjnych dla przygotowanych klasyfikatorów zespołowych oraz wstępnego wytypowanie najlepszego z nich,
- dane testowe — zbiór zawierający obrazy, na których znajdują się różne obiekty, w tym te, które należy wykryć; zbiór służy do ostatecznego przetestowania jakości wybranego/wybranych klasyfikatorów.

4.1 Zbiór 1 — litera A

W pierwszym eksperymencie przygotowany został sztuczny zbiór danych zawierający wielkie litery z nowoczesnego alfabetu angielskiego bazujący na czcionkach komputerowych przygotowanych w pracy (de Campos, Babu i Varma, 2009). Podzbiór reprezentujący literę ‘A’ został ograniczony do kilku czcionek o podobnej charakterystyce i stanowił bazę do przygotowania próbek z pozytywnymi. Podzbiory z innymi literami zostały połączone i na ich podstawie powstały próbki negatywne. W celu zwiększenia liczby próbek w zbiorze uczącym oraz ich urozmaicenia do liter dodane zostały tła. Rys. 4.1 przedstawia materiał źródłowy użyty w eksperymencie.

Na podstawie pozyskanego materiału przygotowane zostały trzy zbiory danych:

- dane uczące — zbiór zawierający gotowe okna kwadratowe o wymiarach od 68×68 do 240×240 pikseli z literami na różnych tłach; litery występują w obrotach od pozycji bazowej w zakresie od -45° do 45° ,
- dane walidacyjne — zbiór zawierający obrazy o wymiarze 600×480 , na których, w sposób losowy, umieszczono okna z literami tak, aby nie nakładały się na siebie; litery występują w dowolnym obrocie,



Rysunek 4.1: Litery i tła użyte do tworzenia zbioru „Litera A”. Pozytywy: litery A (a), negatywy: pozostałe litery (b), tła (c). (Źródło: opracowanie własne)

- dane testowe — zbiór zawierający obrazy o wymiarze 600×480 , na których, w sposób losowy, umieszczono okna z literami tak, aby nie nakładały się na siebie; litery występują w dowolnym obrocie.

Tablica 4.1 zawiera dokładny opis przeprowadzonego eksperymentu.

Tablica 4.1: „Litera A”: nastawy eksperymentu.

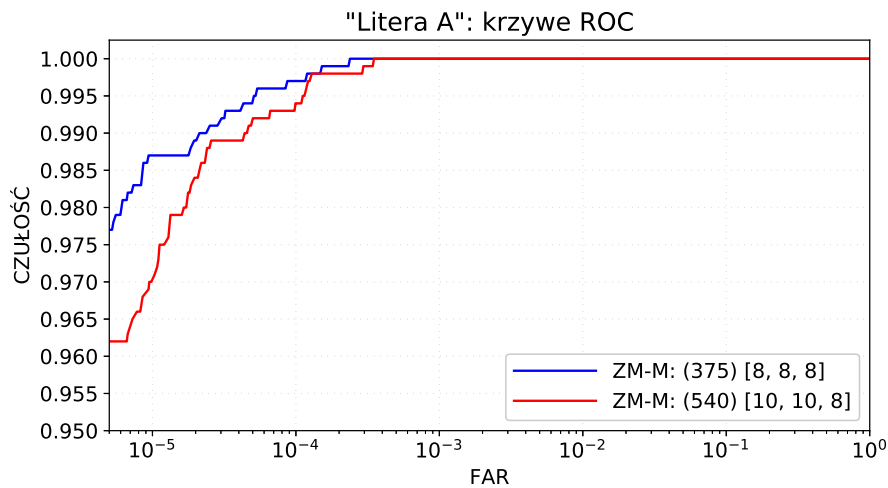
dane uczące		
liczba / parametr	wartość	dodatkowe informacje
liczba przykładów pozytywnych	20 384	okna zawierające literę 'A'
liczba przykładów negatywnych	323 564	okna z literami innymi niż 'A' oraz losowe próbki tła
rozmiar zbioru uczącego	343 948	łącznie pozytywne i negatywne przykłady
dane walidacyjne		
liczba obrazów	500	
liczba pozytywnych przykładów	1 000	okna zawierające literę 'A'
liczba negatywnych przykładów	9 375 000	okna bez litery 'A' (wynikająca z nastaw procedury detekcyjnej)
rozmiar zbioru walidacyjnego	9 376 000	łącznie pozytywnych i negatywnych przykładów
dane testowe		
liczba obrazów	200	
liczba pozytywnych przykładów	381	okna zawierające literę 'A'
liczba negatywnych przykładów	3 750 019	okna bez litery 'A' (wynikająca z nastaw procedury detekcyjnej)
rozmiar zbioru testowego	3 750 400	łącznie pozytywnych i negatywnych przykładów
procedura detekcyjna (skalowanie oknem przesuwным)		
rozdzielczość obrazów	600×480	wymuszona rozdzielczość
liczba skal	5	
współczynnik przyrostu okna	1.2	
rozmiar najmniejszego okna	100×100	
rozmiar największego okna	208×208	
współczynnik przemieszczania okna	0.05	

W omawianym eksperymencie nauczone zostały dwa klasyfikatory:

- dla 375 cech (moduły ZM),

- dla 540 cech (moduły ZM).

Każdy z nauczonych klasyfikatorów został następnie przetestowany na zbiorze walidacyjnym. Rys. 4.2 przedstawia krzywe ROC ze skalą logarymiczną dla osi FAR¹. Na podstawie wykresu można stwierdzić, że mniejsza liczba cech przełożyła się na lepsze wyniki, tj. krzywa niebieska leży powyżej krzywej czerwonej.



Rysunek 4.2: „Litera A”: krzywe ROC dla danych walidacyjnych. (Źródło: opracowanie własne)

Dla każdego z nauczonych klasyfikatorów, wybrany został próg klasyfikacji, dla którego osiągnięta została maksymalna dokładność. Tablica 4.2 przedstawia wartości trzech wskaźników: czułości, FAR oraz dokładności, dla wybranych progów. W tym wypadku ciężko mówić o wyraźnie lepszym klasyfikatorze. Minimalnie lepsze rezultaty uzyskano dla klasyfikatora nauczonego na zestawie 375 cech — zarówno jeśli chodzi o wyższą dokładność, jak i fakt wykrycia jak największej liczby egzemplarzy poszukiwanego obiektu.

Tablica 4.2: Wyniki dla zbioru walidacyjnego: „Litera A”.

przestrzeń cech (liczba cech)	czułość	FAR na obraz	FAR na okno [$\cdot 10^{-6}$]	dokładność na okno
ZM-M (375) [8,8,8]	.964	20/500	2.13	.999994027
ZM-M (540) [10,10,8]	.951	21/500	2.24	.999992534

Następnie, oba klasyfikatory zostały ostatecznie przetestowane na trzecim z przygotowanych zbiorów, tj. zbiorze testowym, który zawierał 200 obrazów, czyli łącznie 3750 400 okien. Wyniki eksperymentu zostały zaprezentowane w Tablicy 4.3.

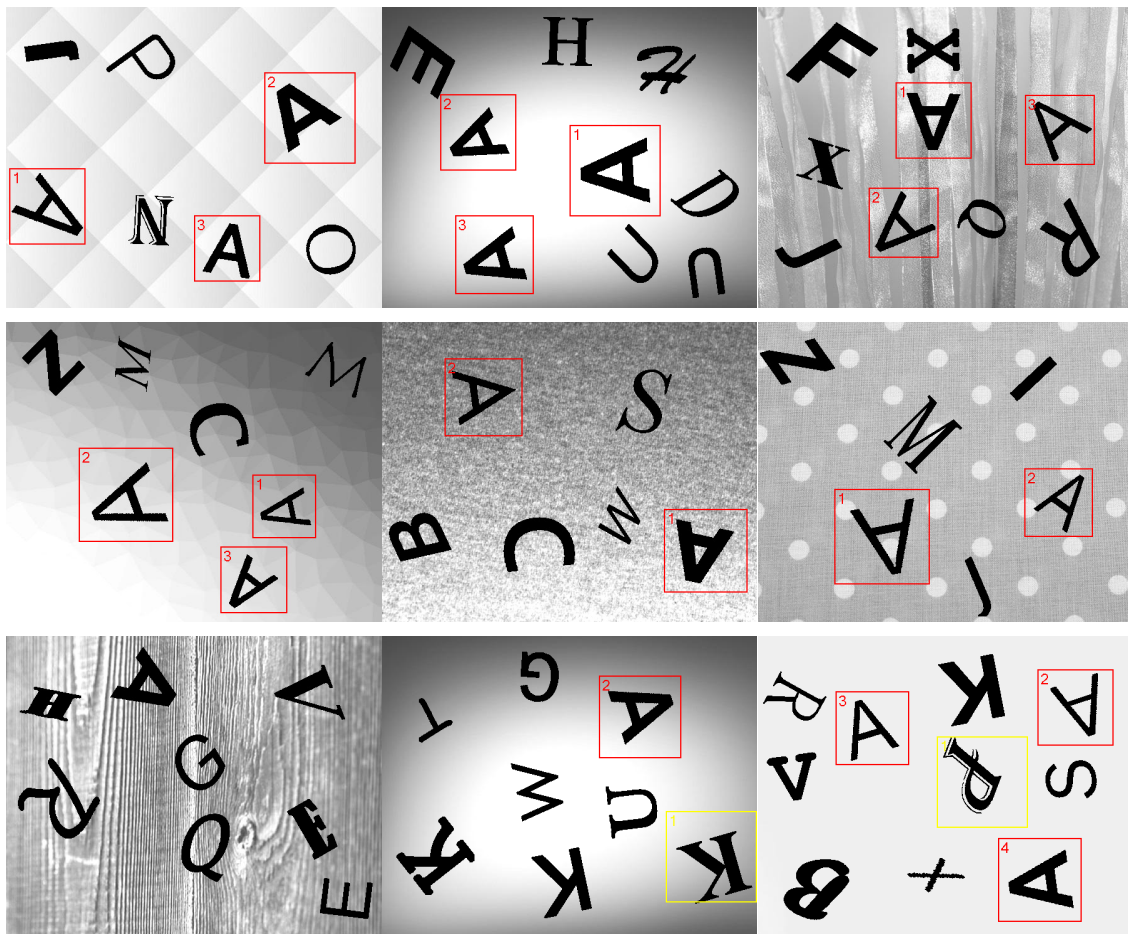
¹FAR ang. *false alarm rate*; wartość tego wskaźnika jest równoważna wartości 1 - specyficzność

Na podstawie tablicy można dokonać ostatecznego wyboru lepszego z klasyfikatorów. Jest nim ten bazujący na 375 cechach.

Tablica 4.3: Wyniki dla zbioru testowego: „Litera A”.

przeźrenie cech (liczba cech)	czułość	FAR na obraz	FAR na okno [$\cdot 10^{-6}$]	dokładność na okno
ZM-M (375) [8,8,8]	.979	8/200	2.13	.999995734
ZM-M (540) [10,10,8]	.961	12/200	3.20	.999992801

Rys. 4.3 przedstawia przykładowe wyniki detekcji.



Rysunek 4.3: „Litera A”: przykłady detekcji. Ostatni wiersz pokazuje brak wskazania, czyli przykład fałszywego negatywu (niewykryta litera A na pierwszym obrazie), oraz dwa obrazy z nieprawidłowymi wskazaniami, czyli przykładami fałszywych pozytywów (litera K oraz P wskazane jako szukana litera A). (Źródło: opracowanie własne)

4.2 Zbiór 2 — samoloty

W drugim z eksperymentów wykorzystany został zbiór DOTA przygotowany przez Xia i in. (2018a) bazujący na zdjęciach płyt lotnisk. Autorzy zebrali 2806 zdjęć o wysokiej rozdzielczości z różnych czujników i platform. Zawierają one obiekty występujące w wielu skalach, w różnej orientacji i rozmaitych kształtach. Do obrazów zostały dołączone adnotacje, wykonane przez ekspertów, opisujące 15 różnych klas obiektów. Autorzy przygotowali trzy podzbiory danych: uczący, walidacyjny oraz testowy. Dokładniejszy opis zbioru został przedstawiony w jednej z prac autorów Xia i in. (2018b). Dodatkowo autorzy przedstawiają zastosowanie przygotowanego zbioru do zadania detekcji (Ding i in., 2019).

W przedstawionym eksperymencie skupiono się na wykrywaniu samolotów. Ze względu na dużą rozdzielczość (np. 12029×5014 , co daje ponad 60 milionów pikseli, czy też 13383×4287 , co daje ponad 57 milionów pikseli w obrazie), a co za tym idzie zwiększenie czasu chociażby wczytywania, obrazy w każdym z trzech podzbiorów zostały pocięte na mniejsze fragmenty. W związku z tym, że sposób oznaczenia obiektów nie był do końca spójny — różna wielkość ramek wokół samolotów — samoloty zostały ponownie pooznaczone tak, aby zminimalizować wielkość ramki wokół nich.

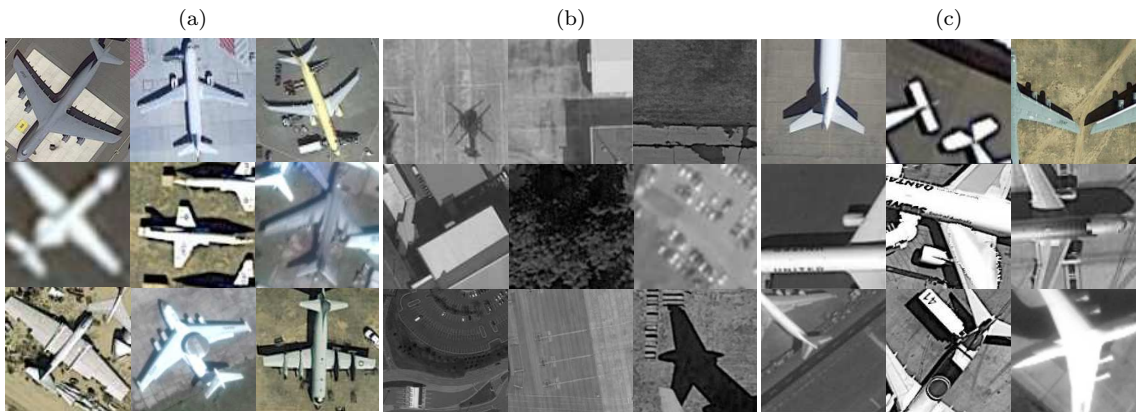
Na podstawie pozyskanego materiału przygotowane zostały trzy zbiory danych:

- dane uczące — zbiór zawierający gotowe okna kwadratowe o wymiarach od 42×42 do 200×200 pikseli z wyciętymi z obrazów próbkami zawierającymi samoloty; samoloty występują w dowolnym obrocie,
- dane walidacyjne — zbiór zawierający obrazy o różnych wymiarach z różnymi obiektami, w tym samolotami; obiekty występują w dowolnym obrocie,
- dane testowe — zbiór zawierający obrazy o różnych wymiarach, na których znajdują się różne obiekty w dowolnym obrocie, w tym samoloty.

Rys. 4.4 przedstawia przykładowe próbki ze zbioru uczącego użyte w eksperymencie. Tablica. 4.4 zawiera dokładny opis przeprowadzonego eksperymentu.

W omawianym eksperymencie nauczone zostały trzy klasyfikatory:

- dla 375 cech (moduły ZM),
- dla 450 cech (moduły ZM),
- dla 540 cech (moduły ZM).



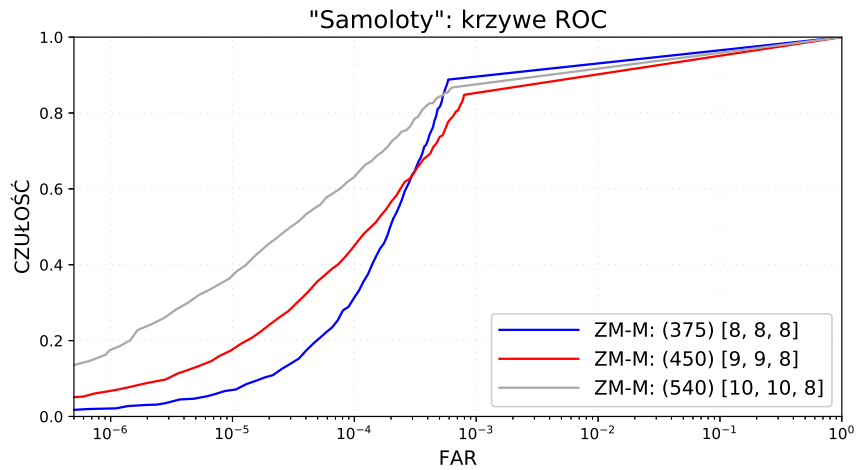
Rysunek 4.4: Przykłady próbek ze zbioru uczącego. Pozytywy: samoloty (a), negatywy: (b), „trudne” negatywy: (c). (Źródło: opracowanie własne)

Tablica 4.4: „Samoloty”: nastawy eksperymentu.

dane uczące		
liczba / parametr	wartość	dodatkowe informacje
liczba przykładów pozytywnych	6 916	okna zawierające samolot
liczba przykładów negatywnych	376 062	okna z obiektami innymi niż samolot oraz losowe próbki tła
rozmiar zbioru uczącego	382 978	łącznie pozytywne i negatywne przykłady
dane walidacyjne		
liczba obrazów	327	
liczba pozytywnych przykładów	1 027	okna zawierające samolot
liczby negatywnych przykładów	45 018 654	okna bez samolotów (wynikające z nastaw procedury detekcyjnej)
rozmiar zbioru walidacyjnego	45 019 681	łącznie pozytywnych i negatywnych przykładów
dane testowe		
liczba obrazów	513	
liczba pozytywnych przykładów	1 267	okna zawierające samolot
liczby negatywnych przykładów	71 052 698	okna bez samolotów (wynikające z nastaw procedury detekcyjnej)
rozmiar zbioru testowego	71 053 965	łącznie pozytywnych i negatywnych przykładów
procedura detekcyjna (skanowanie oknem przesuwным)		
wysokość obrazów	420	wymuszona wysokość
liczba skal	8	
współczynnik przyrostu okna	1.2	
rozmiar najmniejszego okna	42 × 42	
rozmiar największego okna	150 × 150	
współczynnik przemieszczania okna	0.05	

Każdy z nauczonych klasyfikatorów został następnie przetestowany na zbiorze walidacyjnym.

Rys. 4.5 przedstawia krzywe ROC ze skalą logarytmiczną dla osi FAR. Na wykresie widać wyraźnie, że żaden z klasyfikatorów nie osiąga dostatecznie zadowalających wyników dla bardziej skomplikowanego zbioru danych, jakim jest właśnie omawiany zbiór DOTA.



Rysunek 4.5: „Samoloty”: krzywe ROC dla danych walidacyjnych. (Źródło: opracowanie własne)

W związku z bardzo słabymi krzywymi ROC dalszą analizę omawianego zbioru na tym etapie można uznać za bezcelową. Słabe wyniki osiągnięte dla przyjętych cech (modułów momentów Zernike’a) skłoniły do rozważań, jaki jest powód tak słabych wyników. Kolejne dwa rozdziały omawiają i proponują rozwiązania pozwalające na polepszenie jakości klasyfikacji.

Rozdział 5

Redukcja błędów numerycznych

W poniższym rozdziale przedstawione zostało podejście, które pozwala na redukcję błędów numerycznych powstających przy obliczaniu momentów Zernike’a za pomocą obrazów całkowych. Metoda ta została opisana w artykule (Klęsk, Bera i Sychel, 2020).

5.1 Błędy numeryczne

Jak wiadomo, arytmetyka zmiennoprzecinkowa może prowadzić do powstawania błędów numerycznych, gdy liczby biorące udział w danym algorytmie różnią się znacznie co do rzędu wielkości (Goldberg, 1991; Rajaraman, 2016). Mnożenia oraz dzielenia są powszechnie uważane za operacje „bezpieczne”. W przeciwieństwie do nich, dodawania i odejmowania są niebezpieczne, ponieważ, gdy przeprowadzamy te operacje na liczbach o różnych rzędach wielkości, najmniej znaczące cyfry mantysy (tj. skrajne prawe) mogą zostać utracone, gdy dalekie od siebie wykładniki są wyrównywane w celu wykonania operacji. Gdy obliczanie momentów Zernike’a jest wspomagane obrazami całkowymi i przeprowadzane zgodnie z Propozycją 1 (poprzez Algorytm 4), takie sytuacje mogą powstawać w *dwóch* miejscach, przy czym w obydwu przypadkach są spowodowane przez jedną wspólną przyczynę, która zostanie wyjaśniona poniżej.

Na początek, warto zwrócić uwagę, że wspólną częścią wzoru definicyjnego dla ZM (3.11) oraz wzoru opartego na obrazach całkowych (3.18) są wyrażenia: $\frac{\sqrt{2}}{w}(k - k_c)$ (3.13) oraz $\frac{\sqrt{2}}{w}(j_c - j)$ (3.14). Przekształcając one globalne współrzędne piksela na lokalne koło jednostkowe. Gdy wzory te są podstawiane w miejsce x_k i y_j w ramach wzoru (3.11), to wewnątrz sumy pojawia się następujące podwyrażenie:

$$\dots \left(\frac{\sqrt{2}}{w} \right)^{2s+o} (k - ij - k_c + ij_c)^{s-q} (k + ij - k_c - ij_c)^{s+q+o}. \quad (5.1)$$

Teraz, aby móc wprowadzić obrazy całkowe, należy rozwinąć dwa wyrażenia dwumianowe, rozróżniając dwie grupy termów: $k \mp ij$ — zależnych od globalnego indeksu piksela, oraz $-k_c \pm ij_c$ — niezależnych od niego. Dzięki temu, współrzędne danego okna mogą zostać wyizolowane i otrzymuje się wzór (3.18). Niestety, powstają w związku z tym miejsca we wzorze, które można nazwać wrażliwymi numerycznie.

Pierwszym z nich są same obrazy całkowe, zdefiniowane w postaci (3.15). Warto zauważyć, że globalne indeksy pikseli j, k występujące w wyrażeniach potęgowych $(k - ij)^t(k + ij)^u$ mają wartości z zakresu: $0 \leq j \leq n_y - 1$ i $0 \leq k \leq n_x - 1$. Stąd, dla obrazu o rozmiarze np. 640×480 , elementy pod sumą wahają się pod względem swojego rzędu wielkości z grubsza od $10^{0(t+u)}$ do $10^{3(t+u)}$. Dla przykładu, założymy że $t + u = 10$ (co występuje np. gdy $\rho = \varrho = 10$) i przyjmijmy w przybliżeniu stałe wartości funkcji obrazu f . Wtedy obraz całkowy $ii_{t,u}$ musi przechowywać wartości z zakresu od 10^0 do 10^{30} . Oczywiście, błędy zaokrągleń zwiększają się w miarę jak suma w $ii_{t,u}$ podąża w kierunku dolnego prawego rogu obrazu.

Drugie miejsce wrażliwe numerycznie to wyrażenia: $(-k_c + ij_c)^{s-q-t}$ i $(-k_c - ij_c)^{s+q+o-u}$, zawierające tak zwane indeksy centralne — wzór (3.18). Ich iloczyny, przechowywane w tablicy L^c , mogą stać się bardzo duże w miarę jak obliczenia przesuwają się w kierunku dolnego prawego rogu obrazu.

W raportach błędów przedstawiane będą *błędy względne*, oznaczane jako:

$$\text{err}(\phi, \phi^*) = |\phi - \phi^*|/|\phi^*|, \quad (5.2)$$

gdzie ϕ oznacza cechę (z pominięciem indeksów dla czytelności) obliczoną z wykorzystaniem obrazów całkowych zaś ϕ^* jej dokładną wartość wyznaczaną zgodnie z definicją.

Aby dać czytelnikowi ogólny pogląd na tę sytuację, warto wspomnieć, że wyraźne błędy numeryczne pojawiają się dla implementacji w C++ już dla nastaw $\rho = \varrho = 8$, ale nie wpływają jeszcze znacząco na miary jakości detektora. Z kolei dla $\rho = \varrho = 10$ błędy stają się już bardzo widoczne i wpływają wyraźnie na pogorszenie miar jakości. Na przykład dla okna przesuwnego o rozmiarze 48×48 , około 29.7% wszystkich cech posiada błąd względny równy przynajmniej 25%. Numerycznie bezpieczne podejście, które zostanie zaprezentowane w kolejnych sekcjach, redukuje odsetek takich cech do około 0.7%.

5.2 Podział obrazów całkowych

W sekcji tej proponowana jest technika pozwalająca na redukcję błędów numerycznych. Na początek przedstawiony zostanie jej nieformalny opis, po to, aby

zaprezentować ogólny zamysł, a następnie odpowiednie matematyczne wyprowadzenie.

Obrazy całkowe zostaną zdefiniowane *kawałkami*. Aby to zrobić, każdy z obrazów całkowych zostanie podzielony na przylegające do siebie fragmenty, założymy o rozmiarze $W \times W$ (pozostałości na brzegach obrazu mogą być mniejszego rozmiaru). Każdy kawałek otrzyma swój własny „prywatny” układ współrzędnych. Mówiąc nieformalnie, indeksy (j, k) obecne we wzorze (3.15) zostaną wyzerowane w lewym górnym rogu każdego kolejnego kawałka obrazu całkowego. Podobnie też, wartości kumulowane w każdym obrazie całkowym $ii_{t,u}$ zostaną wyzerowane w tych punktach¹. Algorytm 6 demonstruje omawianą konstrukcję.

Szerokość W dla fragmentów powinna zostać tak dobrana, aby przekroczyła maksymalną możliwą szerokość okna przesuwne w procedurze detekcyjnej. Podczas tej procedury, globalne współrzędne (j, k) są wciąż używane w głównych pętlach biegnących po obrazie, ale gdy pozycja okna zostaje ustalona, np. na (j_0, k_0) , wtedy należy przeliczyć to położenie na nowe współrzędne (j'_0, k'_0) właściwe dla obecnego kawałka:

$$N = \lfloor j_0/W \rfloor, \quad M = \lfloor k_0/W \rfloor, \quad (5.3)$$

$$j'_0 = j_0 - N \cdot W, \quad k'_0 = k_0 - M \cdot W. \quad (5.4)$$

Warto zauważyć, że ze względu na wprowadzony podział na kawałki, okno przesuwne może być położone w: jednym, dwóch lub czterech kawałkach obrazu całkowego (ta ostatnia sytuacja została przedstawiona na Rys. 5.1). Dlatego, w przypadku ogólnym, wyniki operatorów przyrostu Δ dla całego obszaru okna będą musiały być obliczane poprzez odpowiednie połączenie przyrostów z czterech części widocznych na rysunku i oznaczonych poprzez P_1, P_2, P_3, P_4 . Istotną rolę w tym zadaniu będzie odgrywać indeks centralny (j_c, k_c) . W oryginalnym podejściu (bez podziału) pozycja indeksu centralnego była obliczana tylko raz, przy użyciu globalnych współrzędnych i wzoru (3.12). Nowa technika wymaga, aby indeks centralny miał współrzędne obliczone w różny sposób w zależności od części P_i , z której na niego patrzemy. Poniżej znajdują się odpowiednie wzory, które bazują na P_1 jako części referencyjnej.

$$j_{c,P_1} = (2j'_0 + w - 1)/2, \quad k_{c,P_1} = (2k'_0 + w - 1)/2, \quad (5.5)$$

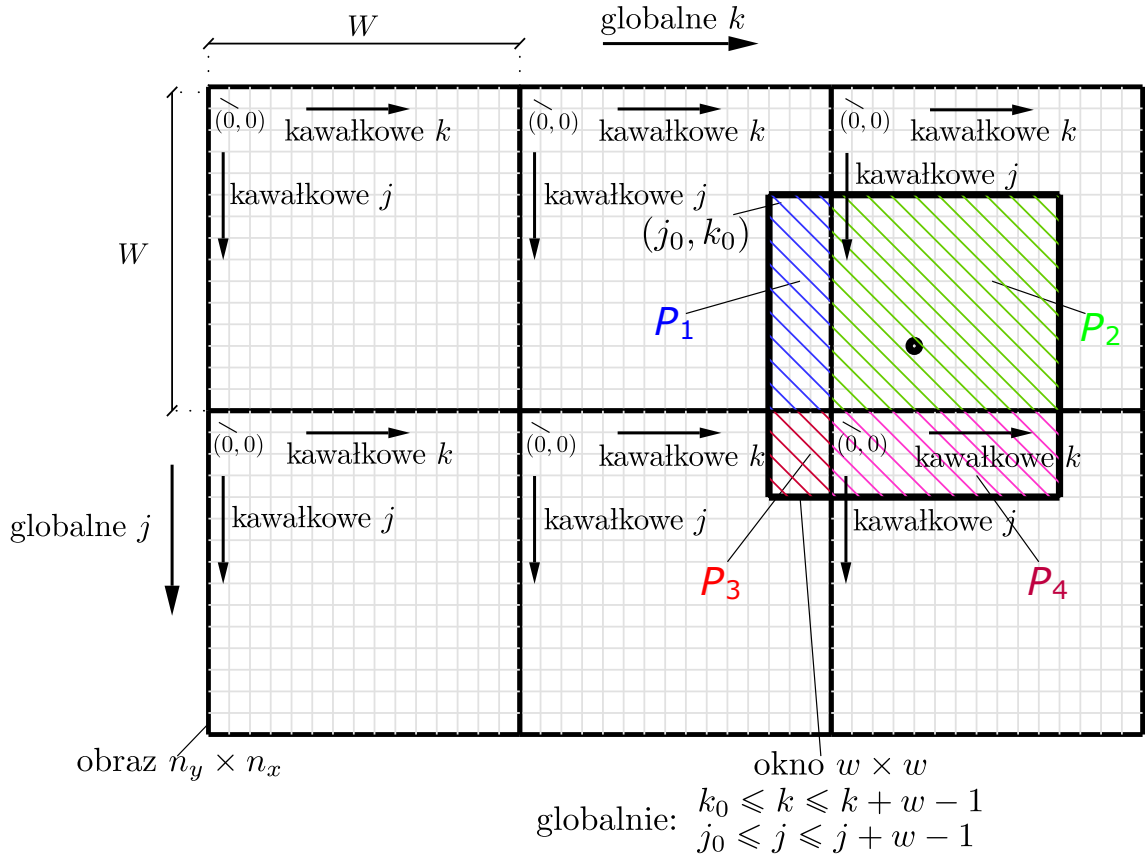
$$j_{c,P_2} = j_{c,P_1}, \quad k_{c,P_2} = k_{c,P_1} - W, \quad (5.6)$$

$$j_{c,P_3} = j_{c,P_1} - W, \quad k_{c,P_3} = k_{c,P_1}, \quad (5.7)$$

$$j_{c,P_4} = j_{c,P_1} - W, \quad k_{c,P_4} = k_{c,P_1} - W. \quad (5.8)$$

¹Przez wyzerowanie rozumie się tutaj sytuację, gdzie dla każdego z kawałków, kumulowanie wartości rozpoczyna się na nowo od wskazanych punktów, co oznacza, że występują odrębne kumulowania w ramach każdego z kawałków.

Algorytm 6 Obraz całkowy zdefiniowany kawałkami.**procedure** PIECEWISEINTEGRALIMAGE(f, t, u, W)Przygotuj tablicę $ii_{t,u}$ o wymiarze $n_x \times n_y$.Przygotuj pomocniczą tablicę ii o wymiarze n_y . $k := 0, j := 0$ **for** $x := 0, \dots, n_x - 1$ **do** **if** $x \bmod W = 0$ **then** $k := 0$ **end if** **for** $y := 0, \dots, n_y - 1$ **do** **if** $y \bmod W = 0$ **then** $j := 0$ **end if** $s := f(x, y)(k - ij)^t(k + ij)^u$ **if** $j > 0$ **then** $ii[y] := ii[y - 1] + s$ **else** $ii[y] := s$ **end if** **if** $k > 0$ **then** $ii_{t,u}[x, y] := ii_{t,u}[x - 1, y] + ii[y]$ **else** $ii_{t,u}[x, y] := ii[y]$ **end if** $j := j + 1$ **end for** $k := k + 1$ **end for****return** $ii_{t,u}$ **end procedure**



Rysunek 5.1: Podział obrazów całkowych. W ogólności, przyrost obrazu całkowego może wymagać złożenia z czterech składników związanych z częściami P_1 , P_2 , P_3 , P_4 przy użyciu redefiniowanych współrzędnych indeksu centralnego (j_c, k_c) . (Źródło: opracowanie własne)

Warto zauważyć, że powyższe wzory mogą w szczególności prowadzić do ujemnych współrzędnych. Dokładniej, w zależności od pozycji okna i części P_i , współrzędne indeksu centralnego mogą mieścić się w zakresie: $-W + w/2 < j_{c,P_i}, k_{c,P_i} < 2W - w/2$.

5.2.1 Numerycznie bezpieczny wzór dla momentów Zernike'a

Poniżej wyprowadzony zostanie wzór pozwalający na ekstrakcję momentów Zernike'a, z wysoką numeryczną stabilnością, oparty na podziale obrazów całkowych. Jako punkt startowy do wyprowadzenia został użyty wariant wzoru (3.11), biorący pod uwagę ogólną pozycję okna z globalnymi indeksami pikseli o postaci: $j_0 \leq j \leq j_0 + w - 1$, $k_0 \leq k \leq k_0 + w - 1$.

$$\begin{aligned}
 \widehat{M}_{2p+o,2q+o} &= \frac{4p+2o+2}{\pi w^2} \sum_{\substack{j_0 \leq j \leq j_0+w-1 \\ k_0 \leq k \leq k_0+w-1}} \sum f(j,k) \sum_{q \leq s \leq p} \beta_{2p+o,2q+o,p-s} \\
 &\quad \cdot \left(\underbrace{\frac{\sqrt{2}}{w}(k-k_c)}_{x_k} + i \underbrace{\frac{\sqrt{2}}{w}(j_c-j)}_{y_j} \right)^{s-q} \cdot \left(\underbrace{\frac{\sqrt{2}}{w}(k-k_c)}_{x_k} - i \underbrace{\frac{\sqrt{2}}{w}(j_c-j)}_{y_j} \right)^{s+q+o} \\
 &= \frac{4p+2o+2}{\pi w^2} \sum_{\substack{j_0 \leq j \leq j_0+w-1 \\ k_0 \leq k \leq k_0+w-1}} \sum_{q \leq s \leq p} \beta_{2p+o,2q+o,p-s} \left(\frac{\sqrt{2}}{w} \right)^{2s+o} \\
 &\quad \cdot \left(k-k_c + i(j_c-j) \right)^{s-q} \left(k-k_c - i(j_c-j) \right)^{s+q+o} \\
 &= \frac{4p+2o+2}{\pi w^2} \sum_{P \in \{P_1, \dots, P_4\}} \sum_{(j_P, k_P) \in P} f_P(j_P, k_P) \cdot \sum_{q \leq s \leq p} \beta_{2p+o,2q+o,p-s} \left(\frac{\sqrt{2}}{w} \right)^{2s+o} \\
 &\quad \cdot \left(k_P - k_{c,P} + i(j_{c,P} - j_P) \right)^{s-q} \cdot \left(k_P - k_{c,P} - i(j_{c,P} - j_P) \right)^{s+q+o} \\
 &= \frac{4p+2o+2}{\pi w^2} \sum_{q \leq s \leq p} \beta_{2p+o,2q+o,p-s} \left(\frac{\sqrt{2}}{w} \right)^{2s+o} \cdot \sum_{t=0}^{s-q} \binom{s-q}{t} \sum_{u=0}^{s+q+o} \binom{s+q+o}{u} \\
 &\quad \cdot \sum_{P \in \{P_1, \dots, P_4\}} (-k_{c,P} + i j_{c,P})^{s-q-t} (-k_{c,P} - i j_{c,P})^{s+q+o-u} \\
 &\quad \cdot \underbrace{\sum_{(j_P, k_P) \in P} f_P(j_P, k_P) (k_P - i j_P)^t (k_P + i j_P)^u}_{\Delta_P(i i_{t,u})}. \tag{5.9}
 \end{aligned}$$

W powyższym wyprowadzeniu na początku wyodrębniono czynnik $(\sqrt{2}/w)^{2s+o}$ poprzez odpowiednie przekształcenia algebraiczne. Drugie przejście ma na celu podzielenie głównego sumowania po wszystkich pikselach w oknie, na cztery mniejsze sumowania po częściach okna przesuwne należących do różnych kawałków obrazu całkowego, co zostało przedstawione na Rys. 5.1. Podany zapis przedstawia najbardziej pesymistyczny przypadek, gdy okno przecina granice podziałów wzdłuż każdej z osi. Dla przypomnienia, możliwe są jeszcze trzy prostsze przypadki: $\{P_1\}$ (brak przecięcia), $\{P_1, P_2\}$ (przecięta jedynie pionowa granica), $\{P_1, P_3\}$ (przecięta jedynie pozioma granica). Omawiane części są bezpośrednio identyfikowane poprzez: punkt zaczepienia okna detekcyjnego (j_0, k_0) , jego szerokość w oraz rozmiar kawałków W . Innymi słowy, można by zapisać funkcyjną zależność postaci $P(j_0, k_0, w, W)$ dla podkreślenia, że każda część P i piksele, które do niej należą, są jednoznacznie znane dzięki powyższemu zestawowi argumentów. Dla czytelności, zapis ten został pominięty. W tym samym przejściu, współrzędne globalne (j, k) zostały zastąpione współrzędnymi (j_P, k_P) właściwymi dla aktualnej części P . Związek pomiędzy współ-

rzędnymi globalnymi a lokalnymi można wyrazić w następujący sposób:

$$j = \begin{cases} N \cdot W + j_P, & \text{dla } P \in \{P_1, P_2\}; \\ (N + 1) \cdot W + j_P, & \text{dla } P \in \{P_3, P_4\}; \end{cases} \quad (5.10)$$

$$k = \begin{cases} M \cdot W + k_P, & \text{dla } P \in \{P_1, P_3\}; \\ (M + 1) \cdot W + k_P, & \text{dla } P \in \{P_2, P_4\}. \end{cases} \quad (5.11)$$

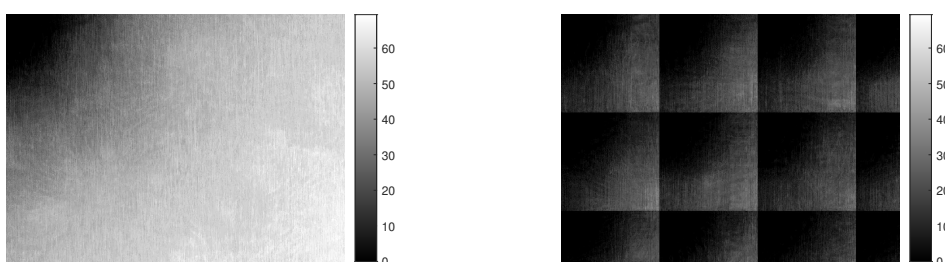
Jasność obrazu w danym pikselu zapisana przy pomocy globalnych współrzędnych jest utożsamiana ze swoim odpowiednikiem w podejściu z podziałami, tj.: $f(j, k) \equiv f_P(j_P, k_P)$. W trzecim przejściu został zastosowany dwukrotnie wzór dwumianowy w celu rozróżnienia dwóch grup czynników: $-k_{c,P} \pm i j_{c,P}$ (niezależne od aktualnego indeksu piksela) i $k_P \mp i j_P$ (zależne od niego). Na koniec zamieniono kolejność sumowania. Ostatnia linia wyprowadzenia pokazuje, w jaki sposób kosztowne obliczenia wykonywane dla wszystkich pikseli w oknie, zostały zastąpione przez tanie operatory Δ — stałoczasowe przyrosty obrazów całkowych.

W odniesieniu do stabilności numerycznej, kluczowy pomysł stojący za wzorem (5.9) polega na tym, że wzór ten używa mniejszych wartości (w porównaniu do oryginalnego podejścia) indeksów j_P , k_P , $j_{c,P}$, $k_{c,P}$ przy zachowaniu tych samych różnic $k_P - k_{c,P}$ oraz $j_{c,P} - j_P$ dla każdego rozważanego piksela. Ma to związek z oryginalnymi wyrażeniami $\frac{\sqrt{2}}{w}(k - k_c)$ i $\frac{\sqrt{2}}{w}(j_c - j)$, które generują współrzędne w kole jednostkowym. Warto zauważyć, że można wprowadzić dowolne przesunięcia np. $k := k \pm \alpha$ i $k_c := k_c \pm \alpha$ (analogicznie dla j , j_c), o ile tylko różnice pozostaną stałe. Idąc dalej, po zastosowaniu rozwinięcia dwumianowego, takie pary przesuniętych indeksów zostają rozdzielone podczas obliczeń i stają się bezpieczniejsze, gdy ich wielkości są odpowiednio zmniejszone.

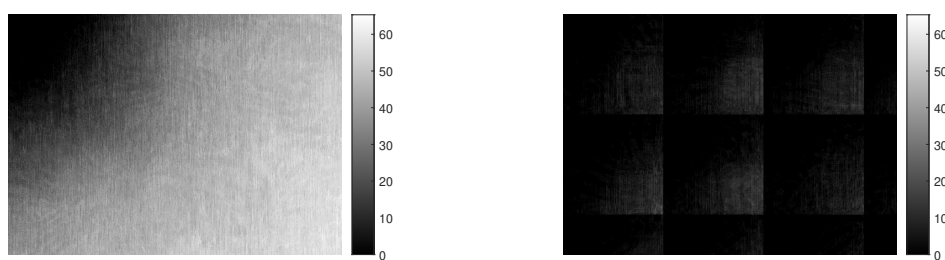
Rysunki 5.3, 5.4 oraz 5.5 ilustrują porównanie podejścia oryginalnego i podejścia bezpieczniejszego numerycznie dla różnych rozmiarów okna przesuwnego dla obrazu wejściowego z Rys. 5.2. Widać na nich, jak odsetek cech, których błąd względny jest większy od 0.25, wzrasta w momencie, gdy okno detekcyjne przesuwa się w kierunku prawego dolnego rogu obrazu o wymiarach 640×480 . Z perspektywy uczenia maszynowego, takie cechy mogą być uznawane za uszkodzone i przekazujące mało informacji. Dodatkowo widać, jaki wpływ na wielkość błędów ma sam rozmiar okna przesuwnego — im mniejsze okno, tym błędy są większe. Dzięki technice podziału, liczba takich cech zostaje znacząco zmniejszona, jak zostało to pokazane na wspomnianych rysunkach. Obserwacje te potwierdzają także wyniki zawarte w Tabelicy 5.1.



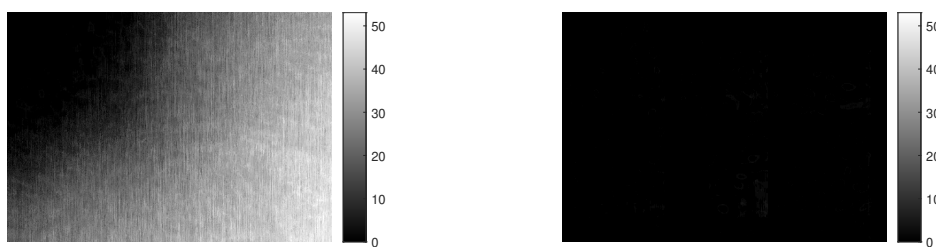
Rysunek 5.2: Przykładowy obraz wejściowy rozdzielczości 640×480 . (Źródło: opracowanie własne)



Rysunek 5.3: Odsetek cech z błędem względnym wynoszącym przynajmniej 0.25 dla każdej możliwej pozycji okna przesuwającego 48×48 dla obrazu wejściowego 640×480 — (po lewej) oryginalne podejście, (po prawej) podejście z podziałami $W = 172$. Nastawy cech $[\rho, \varrho, R] = [12, 12, 1]$. (Źródło: opracowanie własne)



Rysunek 5.4: Odsetek cech z błędem względnym wynoszącym przynajmniej 0.25 dla każdej możliwej pozycji okna przesuwającego 68×68 dla obrazu wejściowego 640×480 — (po lewej) oryginalne podejście, (po prawej) podejście z podziałami $W = 172$. Nastawy cech $[\rho, \varrho, R] = [12, 12, 1]$. (Źródło: opracowanie własne)



Rysunek 5.5: Odsetek cech z błędem względnym wynoszącym przynajmniej 0.25 dla każdej możliwej pozycji okna przesuwego 98×98 dla obrazu wejściowego 640×480 — (po lewej) oryginalne podejście, (po prawej) podejście z podziałami $W = 172$. Nastawy cech $[\rho, \varrho, R] = [12, 12, 1]$. (Źródło: opracowanie własne)

Tablica 5.1: Odsetki możliwie bezużytecznych cech (z błędami względnymi przynajmniej 0.1, 0.25, 0.5) dla obrazu 640×480 i różnych ustawień przestrzeni cech oraz różnych rozmiarów okna przesuwego. Wartości odsetków zostały uśrednione po wszystkich możliwych pozycjach okna. Mniejsze wartości zostały zaznaczone kolorem szarym.

typ obrazów całkowych i rozmiar okna	$[\rho, \varrho, R] = [6, 6, 1]$ błędy względne:			$[\rho, \varrho, R] = [8, 8, 1]$ błędy względne:			$[\rho, \varrho, R] = [10, 10, 1]$ błędy względne:			$[\rho, \varrho, R] = [12, 12, 1]$ błędy względne:		
	0.1	0.25	0.5	0.1	0.25	0.5	0.1	0.25	0.5	0.1	0.25	0.5
oryginal; 48×48	0.056593	0.015286	0.005842	11.180540	8.628347	6.876765	32.101952	29.152176	26.983073	48.513255	46.015266	44.141645
podział; 48×48	0	0	0	0.001355	0.000530	0.000265	1.130805	0.545886	0.276275	11.480231	9.088611	7.459377
oryginal; 56×56	0.011916	0.004223	0.001986	7.271465	5.183115	3.853740	27.180906	24.224691	22.040892	44.194030	41.561724	39.580353
podział; 56×56	0	0	0	0.000274	0.000080	0.000048	0.286107	0.096352	0.036970	7.319810	5.340179	4.024083
oryginal; 68×68	0.003064	0.001030	0.000555	3.424481	2.138255	1.396955	20.670146	17.720114	15.588593	38.152862	35.329513	33.212919
podział; 68×68	0	0	0	0.000051	0.000034	0.000017	0.018006	0.005435	0.002054	3.238189	1.977457	1.268873
oryginal; 82×82	0.000308	0.000140	0	1.237279	0.610829	0.311351	14.264355	11.554403	9.666185	31.649311	28.677193	26.494067
podział; 82×82	0	0	0	0	0	0	0.001893	0.000697	0.000311	0.938823	0.432811	0.202187
oryginal; 98×98	0.000090	0.000030	0.000030	0.255730	0.091225	0.039237	8.695500	6.447168	5.014129	25.019221	22.010698	19.874793
podział; 98×98	0	0	0	0	0	0	0.000240	0.000107	0.000027	0.121211	0.036564	0.014170

Wzór (5.9) odnosi się do momentów $\widehat{M}_{2p+o, 2q+o}$ ekstrahowanych z całych okien, ale może być także uogólniony do wariantu z pierścieniami tj. do $\widehat{M}_{2p+o, 2q+o, r, h}$. Bez wchodzenia w szczegóły techniczne, wystarczy nadmienić, że na poziomie implementacji należy zadbać o staranne wyznaczenie punktów krańcowych dla operacji przyrostu Δ . Należy wziąć pod uwagę rozmiary pierścieni i pozycję dziur względem części P_1, \dots, P_4 .

5.2.2 Rzędy wielkości składników biorących udział w obliczaniu momentów Zernike’a z wykorzystaniem obrazów całkowych — przykład

Tablica 5.2 przedstawia proces obliczeniowy dla konkretnego momentu $\widehat{M}_{10,2}$ i okna przesuwnego zaczepionego w punkcie $(j_0, k_0) = (256, 462)$ wewnątrz obrazu. Rozważane są trzy różne podejścia — bazujące na: definicji, podstawowym podejściu z obrazami całkowymi oraz podejściu z podziałem obrazów całkowych.

Tablica 5.2: Rzędy wielkości składników biorących udział w obliczaniu pojedynczego ZM. Przykład dla $p = 10$, $q = 2$ i okna o punkcie zaczepienia $(j_0, k_0) = (256, 462) = (1 \cdot W + 50, 2 \cdot W + 50)$ dla $W = 206$.

		definicja + rozwinięcie dwumianowe		obrazy całkowe oryginalne podejście		podziały obrazów całkowych	
s	t	składniki		składniki		składniki	
		sumy s_u	sumy s_t	sumy s_u	sumy s_t	sumy s_u	sumy s_t
1	0	10^7	10^5	10^9	10^5	$10^7, 10^8$	10^5
2	0	10^{10}	10^8	$10^{14}, 10^{15}$	10^{10}	$10^{11}, 10^{12}$	10^9
2	1	$10^{10}, 10^{11}$	10^8	$10^{14}, 10^{15}$	10^{10}	$10^{11}, 10^{12}$	10^9
3	0	$10^{13}, 10^{14}$	10^{12}	$10^{20}, 10^{21}$	10^{14}	10^{16}	10^{13}
3	1	$10^{13}, 10^{14}$	10^{12}	$10^{20}, 10^{21}$	10^{14}	10^{16}	10^{13}
3	2	10^{14}	10^{12}	$10^{20}, 10^{21}$	10^{14}	10^{16}	10^{13}
4	0	$10^{16}, 10^{17}, 10^{18}$	10^{15}	$10^{25}, 10^{26}$	10^{18}	$10^{20}, 10^{21}$	10^{16}
4	1	$10^{17}, 10^{18}$	10^{16}	$10^{25}, 10^{26}$	10^{19}	$10^{20}, 10^{21}$	10^{17}
4	2	$10^{17}, 10^{18}$	10^{16}	$10^{25}, 10^{26}$	10^{19}	$10^{20}, 10^{21}$	10^{17}
4	3	10^{18}	10^{16}	10^{26}	10^{18}	$10^{20}, 10^{21}$	10^{17}
5	0	$10^{20}, 10^{21}$	10^{18}	$10^{31}, 10^{32}$	10^{22}	$10^{24}, 10^{25}$	10^{19}
5	1	$10^{20}, 10^{21}$	10^{18}	$10^{31}, 10^{32}$	10^{23}	$10^{24}, 10^{25}$	10^{19}
5	2	$10^{20}, 10^{21}, 10^{22}$	10^{19}	$10^{31}, 10^{32}$	10^{23}	$10^{24}, 10^{25}$	10^{20}
5	3	$10^{20}, 10^{21}, 10^{22}$	10^{19}	10^{32}	10^{22}	$10^{24}, 10^{25}$	10^{20}
5	4	$10^{20}, 10^{21}, 10^{22}$	10^{19}	$10^{32}, 10^{31}$	10^{21}	$10^{24}, 10^{25}$	10^{20}
$\widehat{M}_{10,2}$		0.007158725–0.04123799 i		66.09017+97.04881 i		0.007158975–0.04123780 i	
$\phi_{10,2}$ = $ \widehat{M}_{10,2} $		0.04185474		117.4154		0.04185459	
błąd względny		0.0		2804.318		$3.560209 \cdot 10^{-6}$	

W przypadku tego trzeciego podejścia szerokość kawałków wynosi $W = 206$. Jest to konsekwencją procedury detekcyjnej z $S = 8$ skalami i wskaźnikiem przyrostu okna $\eta = 1.2$. Szerokość W musi zostać tak wybrana, aby była większa niż największe okno (z marginesem) i dlatego dla przedstawionego przykładu $W = \text{round}(48 \cdot \eta^S) = 206$.

W przykładzie, globalny punkt zaczepienia okna $(j_0, k_0) = (256, 462)$ zostaje przeliczony na punkt $(50, 50)$ po uwzględnieniu szerokości $W = 206$ kawałków w obrazie całkowym, ponieważ: $256 - W = 50$ oraz $462 - 2 \cdot W = 50$.

W Tabelicy 5.2, zamiast faktycznych wartości składników, przedstawione zostały ich rzędy wielkości. Obliczenia przebiegają z góry na dół zgodnie z Algorytmem 4. Pierwsze podejście (z definicji) nie stosuje obrazów całkowych, lecz przeprowadza obliczenia stosując rozwinięcie wielomianowe zgodne z Algorytmem 4 tak, aby można było porównać rzędy wielkości sumowanych składników.

Druga kolumna w tabelicy ukazuje jak poważne błędy mogą powstać w momencie zastosowania obrazów całkowych. Błąd względny w tym przykładzie jest na poziomie $2.8 \cdot 10^3$. Jednym z powodów powstania tak dużego błędu jest to, że składniki w ramach sumy s_t wahają się w zakresie od 10^5 do 10^{23} , co oznacza przekroczenie rozpiętości 16 dziesiętnych cyfr znaczących dla mantysy w typie `double`. Warto zauważyć, że podejście z podziałami redukuje rzędy sum składników do: 10^5 — 10^{20} i tym samym redukuje błąd względny do $3.5 \cdot 10^{-6}$.

5.3 Dodatkowe operacje występujące w przypadku numerycznie bezpiecznego podejścia

Wzór (5.9), mimo że bezpieczny numerycznie, powoduje konieczność wykonania dodatkowych obliczeń w porównaniu do wzoru (3.18). W momencie gdy okno przesuwne znajdzie się jedynie w pojedynczej części obrazu całkowego, wzory te są sobie równoważne. W innym przypadku, numerycznie bezpieczna formuła staje się bardziej kosztowna ze względu na sumowania po częściach okna $\sum_{P \in \{P_1, \dots, P_4\}} \dots$. Zamiast operacji przyrostu Δ dla każdej ustalonej trójki s, t, u we wzorze (3.18), w (5.9) mogą wystąpić: jedna, dwie lub cztery operacje przyrostu. Warto zauważyć, że im mniejsze jest okno przesuwne, tym mniej przecięć występuje i dlatego też rzadziej zachodzą pesymistyczne przypadki, gdzie dwie lub cztery operacje przyrostu są potrzebne.

W sekcji tej wyznaczona zostanie wartość oczekiwana liczby operacji przyrostu potrzebnych w numerycznie bezpiecznym podejściu. Przedstawiony zostanie odpowiedni wzór na wspomnianą wartość oczekiwaną z uwzględnieniem wszystkich istotnych stałych. Pokazuje on, ile dodatkowych operacji powstaje poprzez wprowadzenie podziału obrazów całkowych.

Potrzebne będą następujące nowe oznaczenia. Niech S oznacza liczbę skal w procedurze detekcyjnej oraz niech $w(s)$ oznacza szerokość okna detekcyjnego w danej skali:

$$w(s) = \text{round}(\eta^s w_{\min}) + \text{round}(\eta^s w_{\min}) \bmod 2, \quad (5.12)$$

gdzie $s = 0, 1, \dots, S-1$, a w_{\min} i jest najmniejszą początkową szerokością (np. $w_{\min} = 48$). Szerokość okna rośnie ze współczynnikiem η pomiędzy kolejnymi skalami (np. $\eta = 1.2$ oznacza wzrost o 20% aktualnej szerokości). Dodatkowo, niech K_x, K_y oznaczają liczbę części (obrazów całkowych) odpowiednio wzdłuż poziomej i pionowej osi. Dla uproszczenia analizy, założmy że rozmiar obrazu wejściowego jest podzielny przez W bez reszty, tj.: $n_x = K_x W$ i $n_y = K_y W$. W praktyce, okno detekcyjne przesuwane po obrazie używając skoku proporcjonalnego do ułamka jego szerokości, a mianowicie:

$$dx(s) = \text{round}(\lambda w(s)), \quad dy(s) = \text{round}(\lambda w(s)), \quad (5.13)$$

gdzie λ jest współczynnikiem skoku (np. $\lambda = 0.05$).

Poniższa formuła reprezentuje **oczekiwaną liczbę operacji przyrostu** potrzebnych we wzorze (5.9) dla każdej trójki s, t, u .

$$\begin{aligned} E_{\Delta}(S, K_x, K_y, W) = & \sum_{s=0}^{S-1} \frac{1}{dx(s)dy(s)} \left(1 \cdot (W - w(s) + 1)^2 \cdot K_x K_y + 2 \cdot (W - w(s) + 1) \right. \\ & \cdot (w(s) - 1) \cdot \left(2(K_x - 1)(K_y - 1) + (K_x - 1) + (K_y - 1) \right) \\ & \left. + 4 \cdot (w(s) - 1)^2 (K_x - 1)(K_y - 1) \right) \\ & \left/ \sum_{s=0}^{S-1} \frac{1}{dx(s)dy(s)} (K_x W - w(s) + 1)(K_y W - w(s) + 1) \right. \end{aligned} \quad (5.14)$$

Wyprowadzenie powyższego wzoru można przedstawić w sposób następujący. Należy rozważyć prawdopodobieństwa zdarzeń, w których okno detekcyjne: (a) nie przecina żadnych granic podziału obrazów całkowych, (b) przecina tylko jedną z granic — w poziomie albo pionie, (c) przekracza obie granic jednocześnie. Po pierwsze, warto zauważyć, że liczba wszystkich możliwych pozycji okna detekcyjnego w skali s będzie równa $(K_x W - w(s) + 1)(K_y W - w(s) + 1)$, gdy zastosowane zostaną skoki o 1 piksel. Ze skokami $dx(s)$ i $dy(s)$, liczba ta będzie równa

$$\frac{(K_x W - w(s) + 1)(K_y W - w(s) + 1)}{dx(s)dy(s)}, \quad (5.15)$$

gdzie zaniedbane jest zaokrąglenie do liczby całkowitej. Wartość ta posłuży jako mianownik prawdopodobieństw, które należy określić. Teraz można zaobserwować, że prawdopodobieństwo zdarzenia (a) — brak przecięcia — jest wspierane przez następującą liczbę okien:

$$\frac{(W - w(s) + 1)^2}{dx(s)dy(s)} K_x K_y, \quad (5.16)$$

biorąc pod uwagę wszystkie części. Prawdopodobieństwo zdarzenia (b) — jedno przecięcie granic — jest wspierane przez

$$\frac{(W - w(s) + 1)(w(s) - 1)}{dx(s)dy(s)} \left(2(K_x - 1)(K_y - 1) + K_x - 1 + K_y - 1 \right) \quad (5.17)$$

okien. W ten sposób, bierzemy pod uwagę $(K_x - 1)(K_y - 1)$ części, gdzie przecięcie może wystąpić w odniesieniu do osi poziomej lub pionowej (stąd mnożenie przez 2), a także $K_x + K_y - 2$ części brzegowych (z wyjątkiem części najbardziej po prawej u dołu), gdzie przecięcie może wystąpić jedynie w odniesieniu do jednej z osi. Na koniec, prawdopodobieństwo zajścia zdarzenia (c) — dwie granice przecięte — jest wspierane przez

$$\frac{(w(s) - 1)^2}{dx(s)dy(s)} (K_x - 1)(K_y - 1) \quad (5.18)$$

okien. Przez połączenie wszystkich powyższych możliwości oraz wzięciu pod uwagę zmiany skali s otrzymuje się wzór (5.14).

Dla przykładu, po ustaleniu następujących wartości stałych: $S = 8$, $K_x = 3$, $K_y = 2$, $W = 200$ oraz $w_{\min} = 48$, $\eta = 1.2$, $\lambda = 0.05$, wartość oczekiwana wynosi w przybliżeniu $E_{\Delta} \approx 1.505$. Warto nadmienić, że powyższa obliczona wartość teoretyczna nie jest daleka od tej empirycznej rejestrowanej w implementacji: 1.497.

Rozdział 6

Rozszerzona przestrzeń niezmienników Zernike’a

Rozdział ten poświęcony został *rozszerzonej przestrzeni cech*. Zawierać ona będzie już nie tylko moduły ZM jako niezmienniki, ale także rzeczywiste i urojone części odpowiednich iloczynów momentów. Pomysł ten bazuje na pracy (Wallin i Kubler, 1995), w której zaproponowano kompletną listę niezmienników Zernike’a, a także na wcześniejszej pracy (Abu-Mostafa i Psaltis, 1984), w której autorzy wskazali możliwość wykorzystania informacji fazowych z zespolonych momentów (CM) do obrotowej niezmienniczości.

Z perspektywy niniejszej pracy (skoncentrowanej na zadaniu detekcji) potrzebne szczegóły, które należy rozważyć i uzupełnić w związku z rozszerzoną przestrzenią cech, to: odpowiedni schemat indeksowania cech, algorytmiczna procedura ekstrakcji oraz łączna liczba cech.

Dla przypomnienia — momenty Zernike’a są niezmiennicze ze względu na skalowanie, ale nie są niezmiennicze ze względu na obrót (jedynie ich moduły). Pozwalają one jednak tworzyć odpowiednie wyrażenia, które będą posiadały taką własność.

Założmy, że f' oznacza pewną wersję funkcji f , która została obrócona o kąt α , tj. $f'(r, \theta) = f(r, \theta + \alpha)$. Łatwo jest sprawdzić, że spełniona jest następująca tożsamość

$$M'_{p,q} = e^{iq\alpha} M_{p,q}, \quad (6.1)$$

gdzie $M'_{p,q}$ reprezentuje moment dla obróconej funkcji f' (Bera, Kłesk i Sychel, 2019; Sheng i Shen, 1994) — dowód własności został przedstawiony przez (2.10) w Rozdziale 2. Można zaobserwować, że w szczególności *moduły* ZM są niezmiennicze ze względu na obrót, ponieważ:

$$|M'_{p,q}| = |e^{iq\alpha} M_{p,q}| = |e^{iq\alpha}| |M_{p,q}| = |M_{p,q}|. \quad (6.2)$$

Z drugiej strony, można spojrzeć bardziej ogólnie na następujące *iloczynny* momentów

$$M_{p,q}{}^n M_{v,s} \quad (6.3)$$

z pierwszym czynnikiem podniesionym do pewnej naturalnej potęgi n . Po obrocie, iloczyn ten staje się równy:

$$\begin{aligned} M_{p,q}{}^n M_{v,s}' &= e^{inq\alpha} M_{p,q}{}^n e^{is\alpha} M_{v,s} \\ &= e^{i(nq+s)\alpha} M_{p,q}{}^n M_{v,s}. \end{aligned} \quad (6.4)$$

Stąd, narzucając

$$nq + s = 0 \quad (6.5)$$

można otrzymać poszukiwane niezmienniki, ponieważ czynnik $e^{i(nq+s)\alpha}$ zależny od kąta obrotu znika. Warto zauważyć także, że warunek (6.5) nie zawiera indeksów wielomianowych p i v . Stąd, dla każdej odpowiednio przygotowanej trójki n, q, s spełniającej (6.5), można wygenerować sporo niezmienników pozwalając, aby p oraz v zmieniały się zgodnie z ich ograniczeniami tj. $|q| \leq p \leq \rho$, $|s| \leq v \leq \rho$, gdzie oba $p - |q|$ i $v - |s|$ są parzyste.

Jak wspomniano wcześniej, iloczyny (6.3) mogą być traktowane jako bardziej ogólne niż moduły ZM. Jest tak ponieważ, dla specjalnych przypadków, gdzie: $n = 1$, $v = p$ i $s = -q$, iloczyny te upraszczają się do kwadratów modułów ze względu na sprzężenie zespolone:

$$M_{p,q}{}^1 M_{v,s} = M_{p,q} M_{p,-q} = |M_{p,q}|^2. \quad (6.6)$$

Bazując na powyższym poglądowym wprowadzeniu, poniżej przedstawiony zostanie dokładny schemat indeksowania cech dla rozszerzonej przestrzeni. Tak jak już powiedziano, główną rolę będzie odgrywało wyrażenie

$$M_{p,q}{}^n M_{v,s}, \quad nq + s = 0,$$

służące do generowania niezmienników. Należy rozważyć, które piątki (n, p, q, v, s) powinny być uwzględnione w ostatecznej kolekcji indeksów cech oraz jaką informację niosą cechy z nimi związane. Po pierwsze będzie można wyróżnić wśród takich piątek (krotek) kilka grup. Po drugie, pamiętając że $M_{p,q}{}^n M_{v,s}$ jest liczbą zespoloną, można będzie użyć zarówno jej części *rzeczywistej* jak i *urojonej* jako oddzielnych cech. W tym celu krotka musi zostać rozszerzona w taki sposób, aby zawierała sześć indeksów — (n, p, q, v, s, i) — gdzie ostatni indeks $i \in \{0, 1\}$ określa, odpowiednio, która z części $M_{p,q}{}^n M_{v,s}$, rzeczywista czy urojona zostaje użyta.

Dla uproszczenia, zapisy podane poniżej będą wykonane dla niezmienników rozumianych zgodnie z definicją momentów. Niemniej jednak można równoważnie myśleć

także o $\widehat{M}_{p,q}^n \widehat{M}_{v,s}$ lub też $\widehat{M}_{p,q,r,h}^n \widehat{M}_{v,s,r,h}$ jako wyrażeniach generujących niezmienniki.

6.1 Grupy iloczynów i schemat indeksowania

Na początek należy rozważyć krotkę *zer*: $(0, 0, 0, 0, 0, 0)$. Warto zauważyć, że upraszcza to iloczyn $M_{p,q}^n M_{v,s}$ do:

$$1 \cdot M_{0,0} = 1/\pi \int_0^{2\pi} \int_0^1 f(r, \theta) r dr d\theta.$$

Iloczyn ten niesie informację o średniej jasności pikseli w analizowanym oknie obrazu. Warto także zaobserwować, że ta informacja jest niezmiennicza ze względu na obrót. Oczywiście można uwzględnić krotkę *zer* w ostatecznym zbiorze indeksów. Z drugiej strony, nie ma sensu uwzględniać krotek $(n, 0, 0, 0, 0, 0)$ dla $n \geq 1$. Takie krotki reprezentowałyby kolejne potęgi średniej jasności, co stanowiłoby niepotrzebne powielenia. Dodatkowo nie ma potrzeby uwzględniać krotek $(n, 0, 0, 0, 0, 1)$ dla $n \geq 1$. Dla przypomnienia, 1 na ostatnim miejscu wskazuje, że dana krotka reprezentuje część urojoną, która w tym przypadku wynosiłaby zero.

Druga grupa iloczynów, jaką należy uwzględnić przyjmuje postać: $M_{p,0}^n M_{v,0}$. Oba czynniki nie zawierają części harmonicznych (kątowych) i dlatego reprezentują iloczyny skalarne funkcji obrazu z wielomianami radialnymi. W niniejszej pracy niezmienniki będą uwzględniane tylko dla $n = 1$ i $p < v$. Jednym z powodów takiej decyzji jest fakt, że iloczyny są przemienne, np. $M_{2,0}^1 M_{6,0} = M_{6,0}^1 M_{2,0}$, stąd dla $p > v$ nie otrzymuje się żadnej nowej informacji. Decyzja o nieuwzględnianiu przypadków, gdzie $p = v$ lub $n > 1$ jest kwestią osądu, ale można ją uzasadnić tym, że np. $M_{2,0}^2 M_{6,0}$ niesie informację silnie związaną (skorelowaną) z $M_{2,0}^1 M_{6,0}$. Dla przykładu, gdy $\rho = \varrho = 6$ w tej grupie można uwzględnić następujące krotki indeksów:

$$\{(1, 0, 0, 2, 0, 0), (1, 0, 0, 4, 0, 0), (1, 0, 0, 6, 0, 0), (1, 2, 0, 4, 0, 0), \\ (1, 2, 0, 6, 0, 0), (1, 4, 0, 6, 0, 0)\}.$$

Warto zaznaczyć, że części zespolone nie zostały uwzględnione.

Trzecia wyróżniona grupa uwzględnia niezerowe indeksy harmoniczne q i s oraz $n = 1$. Oznacza to, że musi zachodzić $q = -s$, aby spełnić warunek (6.5). Pomimo to, dla uproszczenia, na poziomie implementacji przyjęto, że wszystkie indeksy są nieujemne (w szczególności: $q, s \geq 0$) i poprzez to pracuje się tak naprawdę (i równoważnie) z wyrażeniami postaci:

$$M_{p,q}^n \overline{M}_{v,s} = M_{p,q}^n M_{v,-s}, \quad (6.7)$$

wykorzystując sprzężenie zespolone. W grupie tej możliwe jest, aby $p \leq v$, pamiętając że gdy $p = v$, wtedy przypadek reprezentuje kwadraty modułów i tylko część rzeczywista jest użyteczna. Natomiast, gdy $p < v$ wtedy obie części rzeczywista i urojona są użyteczne. Dla przykładu wypisano poniżej krotki dla $\rho = \varrho = 6$, $n = 1$ i $q = 1$ (ale warto zauważyć, że q może zwiększać się aż do ϱ):

$$\{(1, 1, 1, 1, 1, 0), (1, 1, 1, 3, 1, 0), (1, 1, 1, 3, 1, 1), (1, 1, 1, 5, 1, 0), \\ (1, 1, 1, 5, 1, 1), (1, 3, 1, 3, 1, 0), (1, 3, 1, 5, 1, 0), (1, 3, 1, 5, 1, 1), \\ (1, 5, 1, 5, 1, 0)\}.$$

Czwarta wyróżniona grupa, odnosi się do przypadków zachodzących dla $n > 1$. Oznacza to, że $q \leq \lfloor \varrho/n \rfloor$ i $s = nq$ (ze względu na narzuconą nieujemność). Warto także przypomnieć, że z definicji $v \geq s$. Ponownie, na potrzeby przykładu, poniżej wypisane zostały krotki dla $\rho = \varrho = 6$, $n = 3$, co oznacza, że możliwe wartości q to 1 lub 2:

$$\{(3, 1, 1, 3, 3, 0), (3, 1, 1, 3, 3, 1), (3, 1, 1, 5, 3, 0), (3, 1, 1, 5, 3, 1), \\ (3, 3, 1, 3, 3, 0), (3, 3, 1, 3, 3, 1), (3, 3, 1, 5, 3, 0), (3, 3, 1, 5, 3, 1), \\ (3, 5, 1, 3, 3, 0), (3, 5, 1, 3, 3, 1), (3, 5, 1, 5, 3, 0), (3, 5, 1, 5, 3, 1), \\ (3, 2, 2, 6, 6, 0), (3, 2, 2, 6, 6, 1), (3, 4, 2, 6, 6, 0), (3, 4, 2, 6, 6, 1), \\ (3, 6, 2, 6, 6, 0), (3, 6, 2, 6, 6, 1)\}.$$

Warto zauważyć, że powyższe krotki uwzględniają części rzeczywiste i urojone. Jest tak ponieważ $n > 1$ oznacza, że $q < s$. Innymi słowy przypadki kwadratów modułów nigdy nie mogą zaistnieć.

Podany poniżej Algorytm 7 uwzględnia wszystkie omówione powyżej grupy i może być traktowany jako rutynowa procedura do generowania wszystkich możliwych indeksów (krotek) dla rozszerzonej przestrzeni niezmienników Zernike’a. W algorytmie tym, użyte zostały oznaczenia p_{expr} , q_{expr} , v_{expr} , s_{expr} przez analogię do Algorytmu 4, aby wskazać, że indeksy te są wyrażeniami, które zostaną później rozłożone na ich całkowite połówki i flagi parzystości np. $p_{\text{expr}} = 2p + o$. Rozłożone wartości to te, które są używane bezpośrednio, gdy obliczenia są przeprowadzane zgodnie z Propozycją 1.

Dla czytelności, Algorytm 7 nie uwzględnia *pierścieni*. Można jednak w łatwy sposób wprowadzić tę modyfikację poprzez dodatkowe indeksy, generując (przez dwie dodatkowe pętle) krotki tak, aby miały formę:

$$(n, p_{\text{expr}}, q_{\text{expr}}, v_{\text{expr}}, s_{\text{expr}}, i, r, h). \quad (6.8)$$

Algorytm 7 Indeksy dla rozszerzonych niezmienników Zernike'a.

```
procedure INDEXES( $\rho, \varrho$ )  
   $\mathcal{I} := \{(0, 0, 0, 0, 0, 0)\}$   
  for  $n := 1, \dots, \max\{1, \varrho\}$  do  
    if  $n = 1$  then  
       $q_{\min} := 0$   
    else  
       $q_{\min} := 1$   
    end if  
     $q_{\max} := \lfloor \varrho/n \rfloor$   
    for  $q_{\text{expr}} := q_{\min}, \dots, q_{\max}$  do  
       $s_{\text{expr}} := n \cdot q_{\text{expr}}$   
      for  $q_{\text{expr}} \leq p_{\text{expr}}, p_{\text{expr}} + 2, \dots \leq \rho$  do  
        if  $n = 1$  then  
           $v_{\min} := p_{\text{expr}}$   
        else  
           $v_{\min} := s_{\text{expr}}$   
        end if  
        if  $n = 1$  and  $q_{\text{expr}} = 0$  then  
           $v_{\min} := p_{\text{expr}} + 2$   
        end if  
        for  $v_{\min} \leq v_{\text{expr}}, v_{\text{expr}} + 2, \dots \leq \rho$  do  
           $\mathcal{I} := \mathcal{I} \cup (n, p_{\text{expr}}, q_{\text{expr}}, v_{\text{expr}}, s_{\text{expr}}, 0)$   
           $c := q = 0$   
            or ( $q_{\text{expr}} = s_{\text{expr}}$  and  $p_{\text{expr}} = v_{\text{expr}}$ )  
          if not  $c$  then  
             $\mathcal{I} := \mathcal{I} \cup (n, p_{\text{expr}}, q_{\text{expr}}, v_{\text{expr}}, s_{\text{expr}}, 1)$   
          end if  
        end for  
      end for  
    end for  
  end for  
  return  $\mathcal{I}$   
end procedure
```

Tablica 6.1 przedstawia dokładną liczbę cech w obu przestrzeniach, rozszerzonej i podstawowej. Tablica także nie uwzględnia wariantu z *pierścieniami* (w każdym przypadku wartość zostałaby zwiększona o współczynnik $2R - 1$).

Tablica 6.1: Liczba cech w rozszerzonej (czarne) i podstawowej (szare) przestrzeni cech. Wersja z pierścieniami nie została uwzględniona w tablicy.

$\rho \backslash \varrho$	0	1	2	3	4	5	6	7	8	9	10
0	1										
1	1	2									
2	2	3	6								
3	2	6	11	16							
4	4	8	20	25	34						
5	4	13	29	45	56	63					
6	7	16	43	59	87	94	111				
7	7	23	56	89	121	141	162	171			
8	11	27	75	108	165	185	233	242	265		
9	11	36	92	148	211	250	306	330	355	374	
10	16	41	116	172	268	307	400	424	484	503	530

6.2 Procedura ekstrakcji cech

Mając przygotowany cały schemat indeksowania, można przystąpić do przedstawienia procedury ekstrakcji cech wywoływanej dla każdego analizowanego okna — Algorytm 8. Lista jego argumentów wejściowych jest analogiczna do tej w Algorytmie 4 z jedną różnicą. Oprócz zbioru \mathcal{I} z indeksami cech, potrzebny jest jeszcze zbiór pomocniczy, oznaczony jako \mathcal{I}' . Różnica między tymi dwoma zbiorami jest taka, że \mathcal{I} przechowuje pełną 8-elementową krotkę w formie (6.8), podczas gdy \mathcal{I}' można postrzegać jako słownik krótszych 4-elementowych krotek, które reprezentują unikalne kombinacje momentów i par pierścien-dziura występujących w \mathcal{I} . Powodem takiego zapisu jest fakt, że w ogólności obliczanie rozszerzonej przestrzeni niezmienników jest procesem dwuetapowym. Na początku potrzebne unikalne momenty są wyznaczone i zapamiętywane (na tym etapie przydatny jest słownik \mathcal{I}'). Następnie, momenty te, są używane do generowania potrzebnych iloczynów $\widehat{M}_{p,q,r,h}^n \widehat{M}_{p,q,v,s}$, opisywanych przez indeksy ze zbioru \mathcal{I} . Oczywiście, drugi etap jest lżejszy obliczeniowo, gdyż obejmuje on tylko trzy zespolone operacje: potęgowanie, sprzężenie zespolone i mnożenie. Słownik \mathcal{I}' może zostać przygotowany jednokrotnie, przed procedurą detekcyjną np.

jak zostało to pokazane w Algorytmie 9. Warto także pamiętać, że procedura SINGLEMOMENT, dotycząca pierwszego etapu, działa zgodnie z Propozycją 1.

Algorytm 8 Ekstrakcja cech bazująca na rozszerzonej przestrzeni niezmienników Zernike'a.

procedure EXTRACTEXTENDEDFEATURES($\{ii_{t,u}\}, \mathcal{I}, \mathcal{I}', j_0, k_0, w$)

Przygotuj tablicę L^Δ wypełnioną zerami zespolonymi.

Przygotuj trójki opisujące pierścienie $(j_r), (k_r), (v_r)$ dla danego okna zgodnie z (3.38), (3.37).

for $(p_{\text{expr}}, q_{\text{expr}}, r, h) \in \mathcal{I}'$ **do**

Oblicz $\widehat{M}_{p_{\text{expr}}, q_{\text{expr}}, r, h}$ poprzez wywołanie

SINGLEMOMENT($\{ii_{t,u}\}, p_{\text{expr}}, q_{\text{expr}}, r, h, L^\Delta, (j_r), (k_r), (v_r)$)

i zapamiętaj wyznaczoną wartość.

end for

for $(n, p_{\text{expr}}, q_{\text{expr}}, v_{\text{expr}}, s_{\text{expr}}, i, r, h) \in \mathcal{I}$ **do**

$\varphi := \widehat{M}_{p_{\text{expr}}, q_{\text{expr}}, r, h}^n \cdot \widehat{M}_{v_{\text{expr}}, s_{\text{expr}}, r, h}$

if $i = 0$ **then**

$\varphi := \text{Re } \varphi$

else

$\varphi := \text{Im } \varphi$

end if

$\varphi_{n, p_{\text{expr}}, q_{\text{expr}}, v_{\text{expr}}, s_{\text{expr}}, i, r, h} := \varphi$

Zapamiętaj $\varphi_{n, p_{\text{expr}}, q_{\text{expr}}, v_{\text{expr}}, s_{\text{expr}}, i, r, h}$.

end for

return $\{\varphi_{n, p_{\text{expr}}, q_{\text{expr}}, v_{\text{expr}}, s_{\text{expr}}, i, r, h}\}$

end procedure

Algorytm 9 Unikalne podindeksy występujące w rozszerzonych niezmiennikach Zernike'a.

procedure DISTINCTSUBINDEXES(\mathcal{I})

$\mathcal{I}' := \{\}$ \triangleright pusty słownik na indeksy unikalnych momentów

for $(n, p_{\text{expr}}, q_{\text{expr}}, v_{\text{expr}}, s_{\text{expr}}, i, r, h) \in \mathcal{I}$ **do**

if $(p_{\text{expr}}, q_{\text{expr}}, r, h) \notin \mathcal{I}'$ **then**

$\mathcal{I}' := \mathcal{I}' \cup (p_{\text{expr}}, q_{\text{expr}}, r, h)$

end if

if $(v_{\text{expr}}, s_{\text{expr}}, r, h) \notin \mathcal{I}'$ **then**

$\mathcal{I}' := \mathcal{I}' \cup (v_{\text{expr}}, s_{\text{expr}}, r, h)$

end if

end for

return \mathcal{I}'

end procedure

Rozdział 7

Eksperymenty

Rozdział przedstawia eksperymenty przeprowadzone na dwóch zbiorach danych, które zostały wcześniej opisane w Rozdziale 4. Niezmienne, względem uprzednich opisów, pozostają także sposoby nauki i rodzaje klasyfikatorów, a także nastawy eksperymentów. Nowe wyniki zostały zestawione z tymi z pierwszych eksperymentów z Rozdziału 4.

Wykorzystane zostaną dwie przestrzenie cech: bazujące na modułach momentów Zernike’a (ZM-M) (3.18) oraz bazujące na *rozszerzonych* iloczynach niezmienników (ZM-E) (6.3). Każde z powyższych mogą występować dodatkowo w dwóch odmianach: wersja bez oraz z redukcją błędów numerycznych (te drugie oznaczone przez dopisek „-NER”¹). Jak wcześniej, liczba cech będzie zapisywana w nawiasach. Biorąc pod uwagę przedstawione oznaczenia, przykładowo ZM-E-NER (14250) reprezentować będzie rozszerzoną wersję niezmienników zawierającą ponad 14 tysięcy cech, obliczanych zgodnie ze wzorem (5.9) bazując na podziałach obrazów całkowych.

7.1 Zbiór 1 — litera A

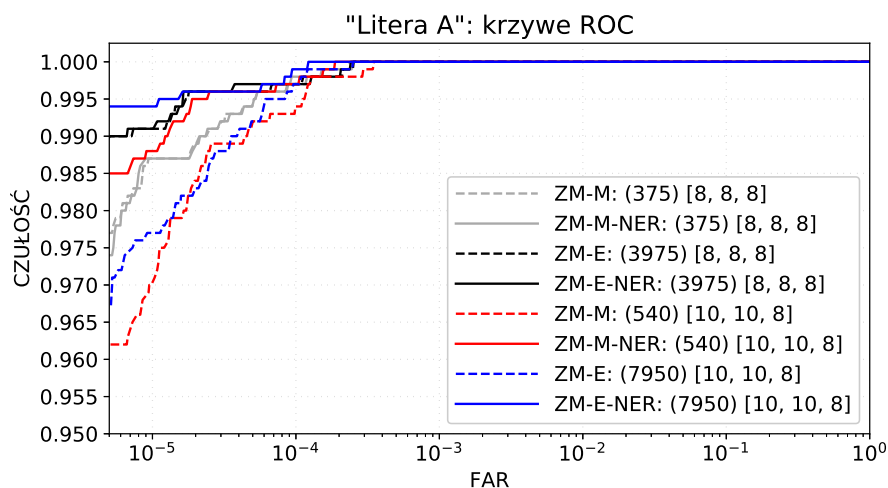
W omawianym eksperymencie nauczone zostały cztery klasyfikatory:

- dla 375 cech (moduły ZM),
- dla 540 cech (moduły ZM),
- dla 3975 cech (wersja rozszerzona ZM-E),
- dla 7950 cech (wersja rozszerzona ZM-E).

Każdy z nauczonych klasyfikatorów został następnie przetestowany na zbiorze walidacyjnym w dwóch wariantach: bez redukcji błędów numerycznych i z nią.

¹ang. *numerical error reduction*

Rys. 7.1 przedstawia krzywe ROC ze skalą logarytmiczną dla osi FAR. Już na tej podstawie można zauważyć, że krzywe ciągłe, które odpowiadają wersji z redukcją błędów numerycznych, reprezentują lepsze wyniki, tj. leżą generalnie powyżej ich odpowiedników bez redukcji błędów numerycznych (oznaczonych krzywymi przerywanymi). Jest to dobrze widoczne szczególnie w przypadku większych stałych ρ i ϱ (niebieskie i czerwone linie na Rys. 7.1).



Rysunek 7.1: „Litera A”: krzywe ROC dla danych walidacyjnych. (Źródło: opracowanie własne)

Dla każdego z nauczonych klasyfikatorów, wybrany został próg klasyfikacji, dla którego osiągnięta została maksymalna dokładność. Tablica 7.1 przedstawia wartości trzech wskaźników: czułości, FAR oraz dokładności, dla wybranych progów. Najlepsze wartości wskaźników prezentuje ostatni z klasyfikatorów, tj. ten dla 7950 cech w wersji z redukcją błędów numerycznych.

Tablica 7.1: Wyniki dla zbioru walidacyjnego: „Litera A”.

przestrzeń cech (liczba cech)	czułość	FAR na obraz	FAR na okno [$\cdot 10^{-6}$]	dokładność na okno
ZM-M (375) [8,8,8]	.964	20/500	2.13	.999994027
ZM-M-NER (375) [8,8,8]	.965	21/500	2.24	.999994027
ZM-E (3975) [8,8,8]	.978	3/500	0.32	.999997334
ZM-E-NER (3975) [8,8,8]	.979	3/500	0.32	.999997440
ZM-M (540) [10,10,8]	.951	21/500	2.24	.999992534
ZM-M-NER (540) [10,10,8]	.977	18/500	1.92	.999995627
ZM-E (7950) [10,10,8]	.952	9/500	0.96	.999993921
ZM-E-NER (7950) [10,10,8]	.991	1/500	0.11	.999998933

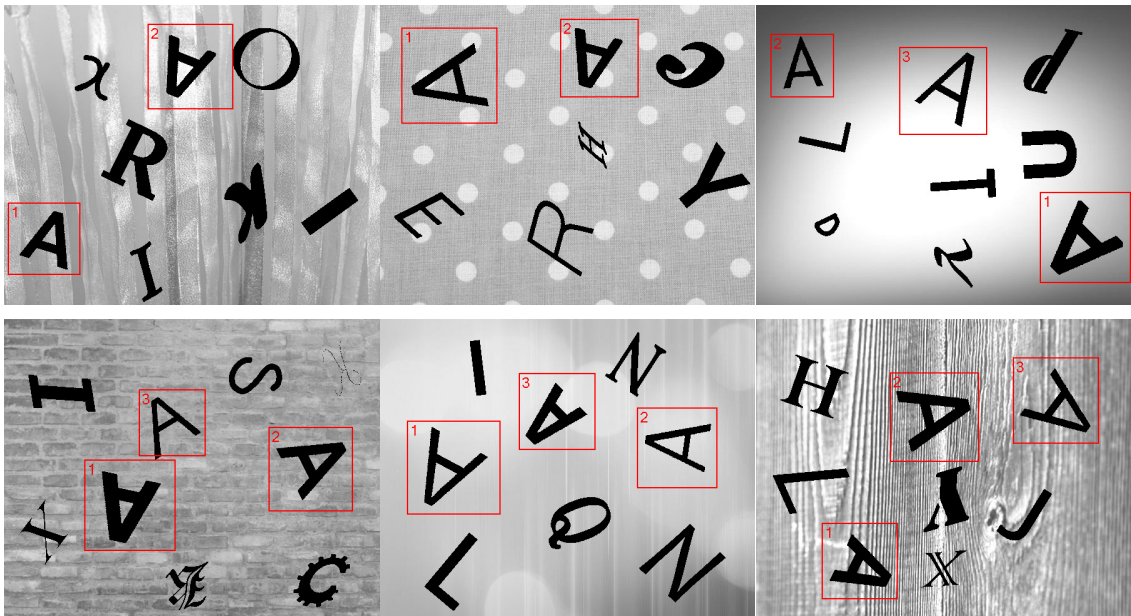
Następnie, wszystkie klasyfikatory zostały ostatecznie przetestowane na trzecim z przygotowanych zbiorów, tj. zbiorze testowym. Wyniki eksperymentu zostały

zaprezentowane w Tablicy 7.2. Na jej podstawie można potwierdzić poprzednie spostrzeżenia — wersje z redukcją błędów numerycznych, dają takie same, lub lepsze, rezultaty detekcji. Na podstawie tablicy można także dokonać wyboru najlepszego z klasyfikatorów. Jest nim klasyfikator bazujący na 7950 cechach w wersji z redukcją błędów numerycznych — co potwierdzają poprzednie obserwacje.

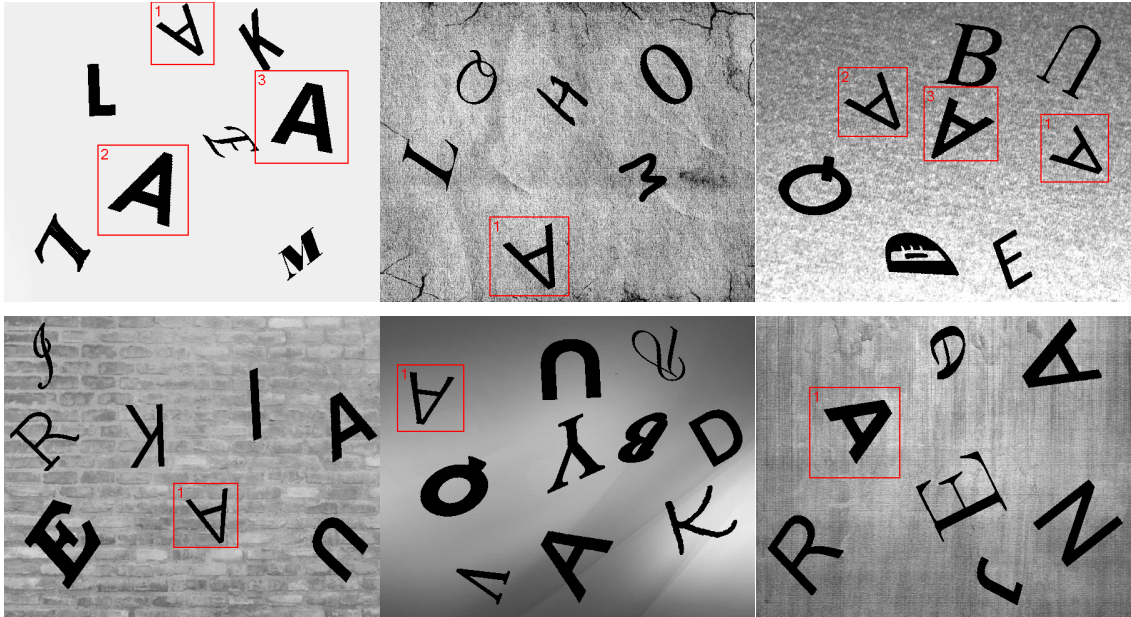
Tablica 7.2: Wyniki dla zbioru testowego: „Litera A”.

przebieg cech (liczba cech)	czułość	FAR na obraz	FAR na okno [$\cdot 10^{-6}$]	dokładność na okno
ZM-M (375) [8,8,8]	.979	8/200	2.13	.999995734
ZM-M-NER (375) [8,8,8]	.979	8/200	2.13	.999995734
ZM-E (3975) [8,8,8]	.979	3/200	0.80	.999997067
ZM-E-NER (3975) [8,8,8]	.979	2/200	0.53	.999997334
ZM-M (540) [10,10,8]	.961	12/200	3.20	.999992801
ZM-M-NER (540) [10,10,8]	.987	8/200	2.13	.999996534
ZM-E (7950) [10,10,8]	.953	4/200	1.07	.999994134
ZM-E-NER (7950) [10,10,8]	.992	0/200	0	.999999200

Rysunki 7.2, 7.3 przedstawiają przykładowe wyniki detekcji. Można zaobserwować, że wybrany klasyfikator bardzo dobrze poradzi sobie z zadaniem wskazywania wybranej litery.



Rysunek 7.2: ‘Litera A’: przykłady detekcji. (Źródło: opracowanie własne)



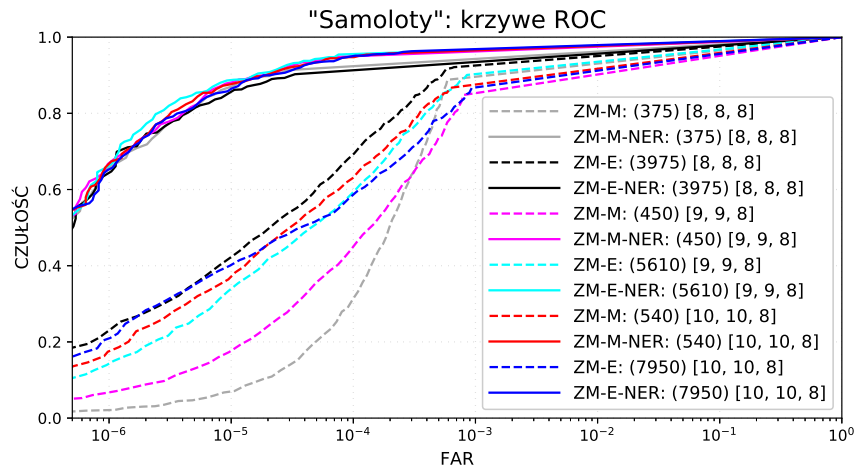
Rysunek 7.3: ‘Litera A’: przykłady detekcji. W ostatnim wierszu znajdują się przykłady z brakiem wskazań — fałszywe negatywy (niewykryte litery A). (Źródło: opracowanie własne)

7.2 Zbiór 2 — samoloty

W omawianym eksperymencie nauczonych zostało sześć klasyfikatorów:

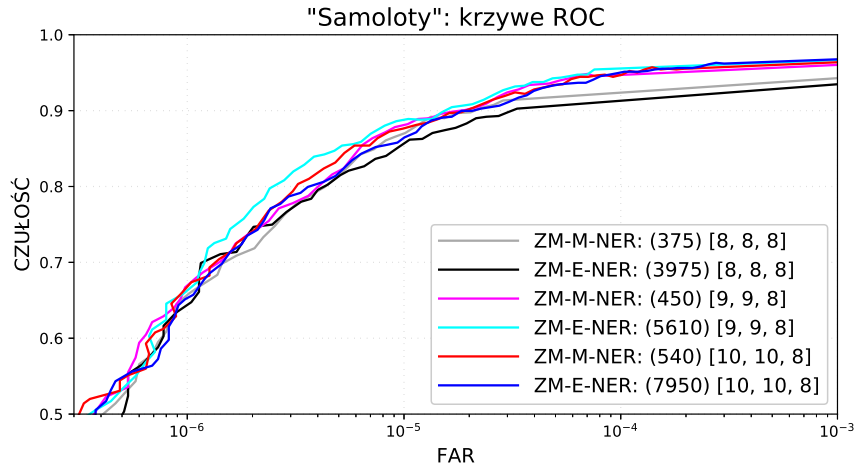
- dla 375 cech (moduły ZM),
- dla 450 cech (moduły ZM),
- dla 540 cech (moduły ZM),
- dla 3975 cech (wersja rozszerzona ZM-E),
- dla 5610 cech (wersja rozszerzona ZM-E),
- dla 7950 cech (wersja rozszerzona ZM-E).

Każdy z nauczonych klasyfikatorów został następnie przetestowany na zbiorze walidacyjnym w dwóch wariantach: bez redukcji błędów numerycznych i z nią. Rys. 7.4 przedstawia krzywe ROC ze skalą logarytmiczną dla osi FAR. W przeciwieństwie do pierwszego opisanego eksperymentu, widać wyraźne różnice pomiędzy wersjami z i bez redukcji błędów numerycznych. W związku z tym, wersje bez redukcji błędów numerycznych nie będą brane pod uwagę w dalszej analizie.



Rysunek 7.4: „Samoloty”: krzywe ROC dla danych walidacyjnych. (Źródło: opracowanie własne)

W celu lepszej analizy, fragmenty pozostałych krzywych ROC zostały przedstawione na osobnym wykresie (Rys. 7.5). Widać na nim, że wszystkie warianty są bardzo podobne. Można jednak przypuszczać, że najlepszym klasyfikatorem okaże się ten oznaczony jasnym niebieskim kolorem, tj. bazujący na 5610 cechach.



Rysunek 7.5: „Samoloty”: wycinki krzywych ROC dla danych walidacyjnych. (Źródło: opracowanie własne)

Dla każdego z sześciu przypadków (wersje z redukcją błędów numerycznych), wybrany został próg klasyfikacji, dla którego osiągnięta została maksymalna dokładność na zbiorze walidacyjnym. Tablica 7.3 przedstawia wartości trzech wskaźników: czułości, FAR oraz dokładności, dla wybranych progów. Najlepsze wartości wskaźników prezentuje czwarty z klasyfikatorów, tj. ten dla 5610 cech.

Tablica 7.3: Wyniki dla zbioru walidacyjnego: „Samoloty”.

przeźrenie cech (liczba cech)	czułość	FAR na obraz	FAR na okno [·10 ⁻⁶]	dokładność na okno
ZM-M-NER (375) [8,8,8]	.762	126/327	2.80	.999991781
ZM-E-NER (3975) [8,8,8]	.747	91/327	2.02	.999992203
ZM-M-NER (450) [9,9,8]	.771	119/327	2.64	.999992137
ZM-E-NER (5610) [9,9,8]	.797	108/327	2.40	.999992981
ZM-M-NER (540) [10,10,8]	.768	107/327	2.38	.999992337
ZM-E-NER (7950) [10,10,8]	.771	109/327	2.42	.999992359

Następnie, wszystkie 6 podejść zostało ostatecznie przetestowanych na trzecim z przygotowanych zbiorów, tj. zbiorze testowym zawierającym 513 obrazów (łącznie 71 053 965 okien). Wyniki eksperymentu zostały zaprezentowane w Tablicy 7.4. Widać, że najwyższe wyniki uzyskano dla czwartego klasyfikatora — tak samo jak w przypadku zbioru walidacyjnego.

Tablica 7.4: Wyniki dla zbioru testowego: „Samoloty”.

przeźrenie cech (liczba cech)	czułość	FAR na obraz	FAR na okno [·10 ⁻⁶]	dokładność na okno
ZM-M-NER (375) [8,8,8]	.669	212/513	2.98	.999991119
ZMs-E-NER (3975) [8,8,8]	.654	137/513	1.93	.999991893
ZM-M-NER (450) [9,9,8]	.687	185/513	2.60	.999991809
ZM-E-NER (5610) [9,9,8]	.706	197/513	2.77	.999991978
ZM-M-NER (540) [10,10,8]	.698	188/513	2.65	.999991964
ZM-E-NER (7950) [10,10,8]	.694	195/513	2.74	.999991795

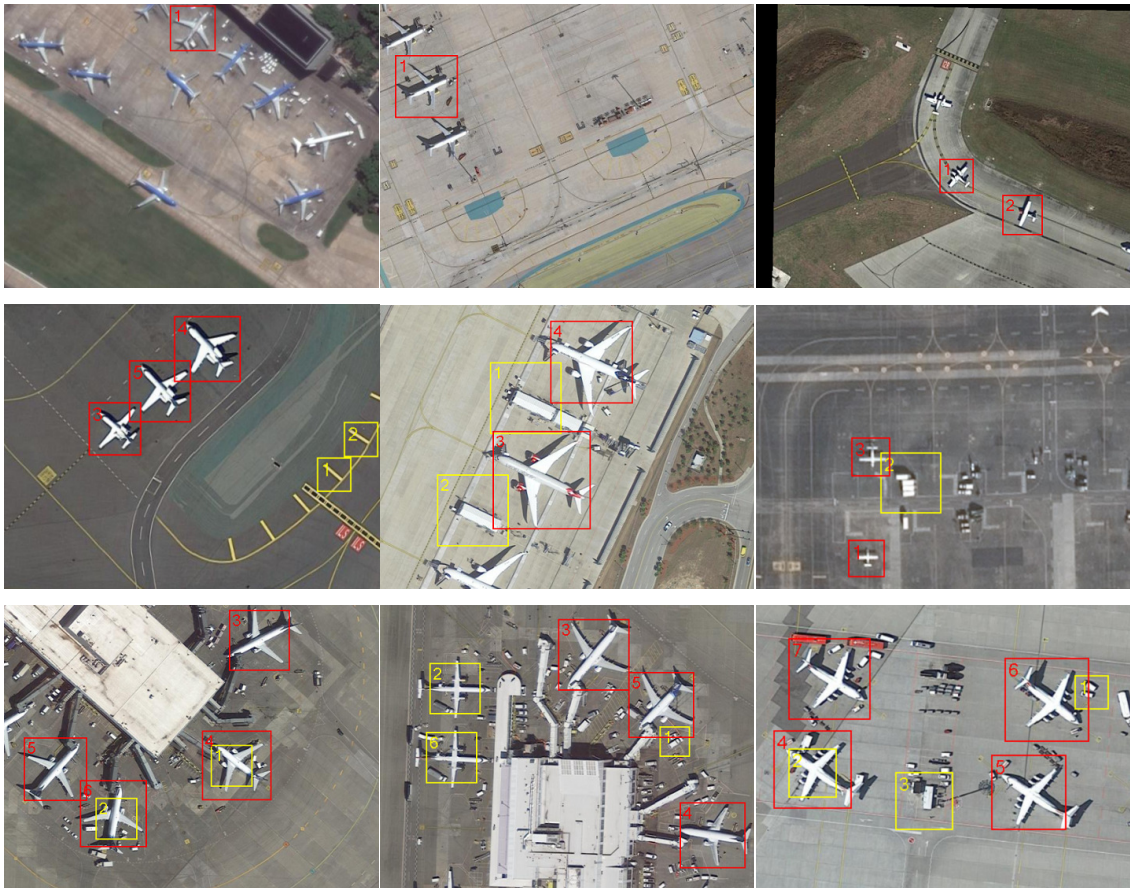
Rysunki 7.6, 7.7, 7.8 oraz 7.9 przedstawiają przykładowe wyniki detekcji dla wybranego przypadku. Na Rysunkach 7.6 oraz 7.7 widać, że klasyfikator poprawnie wskazał samoloty bez względu na ich wielkość, ostrość zdjęcia, czy kolor tła.

**Rysunek 7.6:** „Samoloty”: przykłady detekcji. Poprawne wskazania. (Źródło: opracowanie własne)

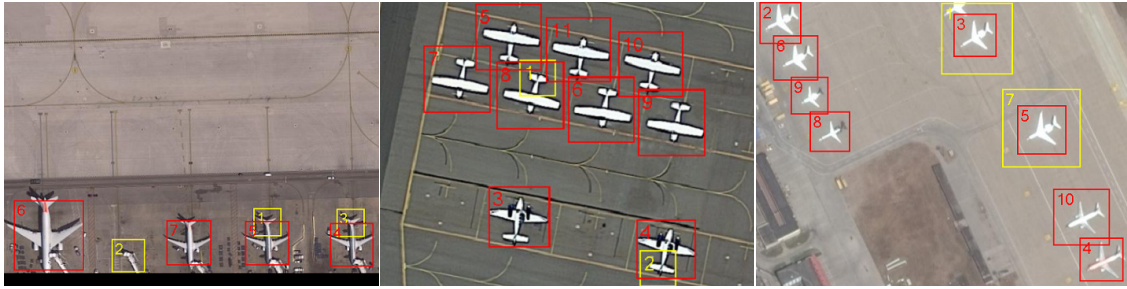


Rysunek 7.7: „Samoloty”: przykłady detekcji. Poprawne wskazania. (Źródło: opracowanie własne)

Na Rysunkach 7.8, 7.9 można zaobserwować różne przykłady nieprawidłowych wskazań lub też ich brak. W pierwszym wierszu Rys. 7.8 znajdują się trzy obrazy na których widać, że część z maszyn nie została rozpoznana, przy dobranym progu (są to fałszywe negatywy). Klasyfikator jako pozytywy wskazywał także inne obiekty/elementy znajdujące się na płytach lotnisk np. oznakowanie, czy inne pojazdy (drugi wiersz Rys. 7.8). W ostatnim wierszu Rys. 7.8 oraz na Rys. 7.9 można zaobserwować przykłady, gdzie umiejscowienie samolotów zostało poprawnie wskazane, jednak wskazania nie obejmują całych maszyn bądź też zawierają zbyt duży obszar wokół nich. Zdarzało się także, że wskazywane, jako samoloty, były ogony maszyn, które mają zachowany kształt całej maszyny (pierwszy oraz drugi obrazek na Rys. 7.9). Wszystkie te sytuacje przyczyniały się do ilościowego obniżenia miar dokładności otrzymanych detektorów.



Rysunek 7.8: „Samoloty”: przykłady detekcji. Błędne wskazania lub ich brak. (Źródło: opracowanie własne)



Rysunek 7.9: „Samoloty”: przykłady detekcji. Błędne wskazania lub ich brak. (Źródło: opracowanie własne)

7.3 Porównanie wydajności czasowych

Tablica 7.5 prezentuje średnie czasy analizy 200 okien o różnych wymiarach dla trzech wersji wyznaczania momentów Zernike’a: z definicji, przy pomocy obrazów całkowych oraz przy pomocy obrazów całkowych z redukcją błędów numerycznych. Widać w niej, jak wprowadzenie techniki opartej na zbiorze zespolonych obrazów całkowych zaproponowanych w pracy wpłynęło na istotne zmniejszenie czasu ekstrakcji cech. W momencie gdy, dla zaproponowanej podstawowej metody, czas analizy okien różnej wielkości pozostaje bez większych zmian, obliczaniu momentów z definicji towarzyszy znaczny wzrost czasu obliczeń wraz ze wzrostem rozmiaru analizowanych okien. Pokazuje to, że przedstawiona w pracy metoda, została uniezależniona od liczby pikseli w oknie.

Tablica 7.5: „Litera A”: średnie czasy analizy 200 okien obrazu przy pomocy różnych sposobów obliczania momentów Zernike’a (obliczenia równoległe na: Intel Core i7-4790K CPU 4.00GHz, 4/8 c/t).

rozmiar okna	100	120	144	172	208	średnia
ZM z definicji	2 774 283 μ s	3 935 776 μ s	5 653 212 μ s	8 037 760 μ s	11 569 000 μ s	
ZM-M	288 μ s	268 μ s	273 μ s	278 μ s	302 μ s	282 μ s
ZM-M-NER	543 μ s	543 μ s	545 μ s	672 μ s	953 μ s	651 μ s

Poniższe tablice przedstawiają porównanie czasów wykonania podejść podstawowych oraz tych numerycznie bezpiecznych, na przykładzie eksperymentu dotyczącego wykrywania litery A.

Na podstawie Tablicy 7.6 oraz 7.7 można określić przyrost czasu dla podstawowej wersji cech (modułów). Jak widać, czas przygotowania jednego obrazu całkowego zwiększył się o 1.2ms, natomiast czas całej procedury detekcyjnej wzrósł o 363ms. Jest to zrozumiałe, ze względu na fakt dodatkowych obliczeń towarzyszących metodzie bezpiecznej numerycznie. Zaobserwowany wzrost nie jest jednak mocno znaczący.

Tablica 7.6: „Litera A” (ZM-M, podejście podstawowe): wydajność czasowa dla obrazu o wymiarach 640×480 (obliczenia równoległe na: Intel Core i7-4790K CPU 4.00GHz, 4/8 c/t).

ZM-M (540) [10, 10, 8], $B = 64$, $T = 512$	
ilość (lub operacje)	czas lub ilość
liczba analizowanych okien	18 752
całkowity czas procedury detekcyjnej	1066 ms
liczba przygotowanych obrazów całkowych	36
czas przygotowania wszystkich obrazów całkowych	95 ms
czas przygotowania 1 obrazu całkowego	2.6 ms
czas na 1 okno	56.85 μ s (zamortyzowany: 51.78 μ s)
liczba unikalnych cech użytych przez klasyfikator	538
czas na 1 okno i 1 cechę	105.67 ns (zamortyzowany: 96.25 ns)

Tablica 7.7: „Litera A” (ZMs-M-NER, podejście z podziałami): wydajność czasowa dla obrazu o wymiarach 640×480 (obliczenia równoległe na: Intel Core i7-4790K CPU 4.00GHz, 4/8 c/t).

ZM-M-NER (540) [10, 10, 8], $B = 64$, $T = 512$	
ilość (lub operacje)	czas lub ilość
liczba analizowanych okien	18 752
całkowity czas procedury detekcyjnej	1429 ms
liczba przygotowanych obrazów całkowych	36
czas przygotowania wszystkich obrazów całkowych	138 ms
czas przygotowania 1 obrazu całkowego	3.8 ms
czas na 1 okno	76.21 μ s (zamortyzowany: 68.90 μ s)
liczba unikalnych cech użytych przez klasyfikator	538
czas na 1 okno i 1 cechę	141.65 ns (zamortyzowany: 128.07 ns)

Na podstawie Tablicy 7.8 oraz 7.9 można porównać czasy wykonania dla wersji rozszerzonej. W tym przypadku czas potrzebny do przygotowania jednego obrazu całkowego zwiększył się o 1.2 ms, czas detekcji wzrósł trzykrotnie.

Tablica 7.8: „Litera A” (ZM-E, podejście podstawowe): wydajność czasowa dla obrazu o wymiarach 640×480 (obliczenia równoległe na: Intel Core i7-4790K CPU 4.00GHz, 4/8 c/t).

ZM-E (7950) [10, 10, 8], $B = 64$, $T = 512$	
ilość (lub operacje)	czas lub ilość
liczba analizowanych okien	18 752
całkowity czas procedury detekcyjnej	1559 ms
liczba przygotowanych obrazów całkowych	36
czas przygotowania wszystkich obrazów całkowych	95 ms
czas przygotowania 1 obrazu całkowego	2.6 ms
czas na 1 okno	83.14 μ s (zamortyzowany: 78.10 μ s)
liczba unikalnych cech użytych przez klasyfikator	3 505
czas na 1 okno i 1 cechę	23.72 ns (zamortyzowany: 22.28 ns)

Tablica 7.9: „Litera A” (ZM-E-NER, podejście z podziałami): wydajność czasowa dla obrazu o wymiarach 640×480 (obliczenia równoległe na: Intel Core i7-4790K CPU 4.00GHz, 4/8 c/t).

ZM-E-NER (7950) [10, 10, 8], $B = 64$, $T = 512$	
ilość (lub operacje)	czas lub ilość
liczba analizowanych okien	18 752
całkowity czas procedury detekcyjnej	4704 ms
liczba przygotowanych obrazów całkowych	36
czas przygotowania wszystkich obrazów całkowych	138 ms
czas przygotowania 1 obrazu całkowego	3.8 ms
czas na 1 okno	250.85 μ s (zamortyzowany: 243.50 μ s)
liczba unikalnych cech użytych przez klasyfikator	3 505
czas na 1 okno i 1 cechę	71.57 ns (zamortyzowany: 69.47 ns)

Rozdział 8

Podsumowanie pracy

Celem niniejszej rozprawy doktorskiej było opracowanie algorytmu, który umożliwiłby szybkie obliczanie momentów Zernike'a w ramach procedury detekcyjnej tak, aby uniezależnić się od pozycji okna i liczby pikseli w oknie. Postawiony cel został zrealizowany.

W pracy zaproponowano zestaw zespolonych obrazów całkowych (skonstruowanych w jednym globalnym układzie współrzędnych kartezjańskich) wyznaczanych przed procedurą detekcyjną, który pozwala na obliczenie momentów Zernike'a dla dowolnych kwadratowych okien obrazu (zdefiniowanych w wielu lokalnych układach współrzędnych biegunowych) w czasie stałym. Ze względu na stosunkowo niewielką liczbę cech, przedstawiono modyfikację podstawowego pomysłu, w której wprowadza się tzw. pierścienie. Pomysł bazuje na wyznaczeniu momentów z pewnych fragmentów okna (tj. właśnie z pierścieni) zamiast z całego okna. Pierwsze eksperymenty pokazały, że dla prostych zbiorów danych, zaproponowana metoda daje zadowalające rezultaty, jednak przy bardziej skomplikowanym zbiorze danych, jakość detekcji nie jest wystarczająca. Następnie, przedstawiono rozwiązanie, pozwalające na zmniejszenie błędów numerycznych powstałych przy wprowadzeniu obrazów całkowych do wyznaczenia momentów Zernike'a. Zaproponowano technikę opartą na podziale obrazów całkowych, która pozwala na znaczną redukcję błędów numerycznych, co zostało poparte przez zaprezentowane tabele i ilustracje graficzne. Dodatkowo, zaproponowano rozszerzoną przestrzeń niezmienników Zernike'a, która pozwala na znaczne zwiększenie liczby ekstrahowanych cech.

Praca zawiera dwa eksperymenty, na zbiorach danych o różnym stopniu skomplikowania, które potwierdzają, że udało się zredukować błędy numeryczne i uzyskać lepsze wyniki w zadaniach detekcji — szczególnie dla drugiego, bardziej złożonego zbioru, widać znaczącą poprawę w stosunku do podstawowej wersji algorytmu.

Należy zauważyć, że informacja zawarta w momentach Zernike'a nie jest wy-

starczająca do opisywania bardziej złożonych obiektów i wykrywania ich z zadowalającym efektem (np. obiektów bardziej realistycznych i trudnych do wykrycia, jak w przypadku eksperymentu „Samoloty”). W związku z tym, ich stosowanie powinno być ograniczone do pewnych stosunkowo prostych kształtów geometrycznych, np. dla obrazów o niewielkich wariacjach, co potwierdził prosty eksperyment „Litera A”. Słaba wartość informacji niesiona przez momenty Zernike’a jest łatwo zauważalna w zadaniu rekonstrukcji obrazów — aby zrekonstruować istotne szczegóły, konieczne jest stosowanie momentów istotnie wysokich rzędów. W kontekście procedury detekcyjnej, ten fakt niesie niestety ze sobą wzrost złożoności obliczeniowej i trudności numeryczne (w zadaniu detekcji powinniśmy ograniczać się do momentów stosunkowo niskich rzędów). Tym samym, należy uczciwie zaznaczyć, że aktualne wyniki literaturowe dotyczące detekcji rzeczywistych obiektów z wykorzystaniem wspomnianych cech Haara czy też deskryptora HOG są lepsze. Z drugiej strony należy jednocześnie przypomnieć, że cechy te nie posiadają ważnej własności, jaką jest obrotowa niezmienniczość. Stąd też, zdaniem autorki, podjęte w niniejszej pracy badania nad szybką (stałoczasową) ekstrakcją momentów Zernike’a są zasadne, pomimo tylko częściowego powodzenia wypracowanych algorytmów.

Warto także wspomnieć o istnieniu typu zmiennoprzecinkowego z poczwórną precyzją (ang. *quadruple-precision floating point*), w którym 113 bitowa mantysa przekłada się na około 34 dziesiętne cyfry znaczące. W kontekście momentów Zernike’a, zastosowanie wspomnianego typu pozwoliłoby na wyznaczanie momentów wyższego rzędu, umożliwiając tym samym rozpoznawanie bardziej skomplikowanych obiektów oraz rezygnację z zaproponowanej (dość skomplikowanej) metody redukcji błędów. Niestety, wspomniany typ nie jest powszechnie dostępny dla kompilatorów.

8.1 Dalsze kierunki badań

Kolejnym kierunkiem badań, będącym naturalną kontynuacją niniejszej rozprawy doktorskiej, może być zmodyfikowanie zaproponowanej metody tak, aby dodatkowo ją przyspieszyć i zastosować w zadaniach detekcji obiektów z obrazu kamery w czasie rzeczywistym. Opisany poniżej pomysł stanowi modyfikację metody ang. *frame skipping* zastosowanej np. w pracy (Ming, Ma i Ling, 2012).

W celu przyspieszenia zaproponowanej metody, można by wykonywać pełen skan obrazu tylko co którąś pozyskiwaną klatkę, założmy co 5 klatek. Na początku, wykonywany byłby pełen skan obrazu w celu odnalezienia (wykrycia) poszukiwanych obiektów, określając ich bieżącą pozycję na obrazie. Dla kolejnej klatki, okno przesuwne umiejscawiane byłoby jedynie w bliskich otoczeniach znalezionych w pełnym

przeszukiwaniu obiektów, znajdując tym samym ich zaktualizowaną pozycję. Dla następnej klatki, okno przesuwne byłoby umieszczane w otoczeniach znalezionych dla poprzedniej klatki położeń itd. jeszcze dla dwóch klatek. Po 4 klatkach, następowaloby powtórne pełne przeszukanie obrazu, co umożliwiłoby m.in. odszukanie obiektów, które dopiero się na nim pojawiły.

Takie wstępne określenie położenia obiektów, można by wykonywać także przy pomocy kamer termowizyjnych, gdy wykrywane obiekty wytwarzają ciepło, lub kamery głębi w momencie, gdy zależałoby nam na detekcji obiektów znajdujących się blisko kamery.

Spis rysunków

1	Główne osiągnięcia w dziedzinie widzenia komputerowego na przestrzeni lat. (Źródło: Szeliski (2011))	8
1.1	Ilustracja zadania detekcji twarzy na obrazach. (Źródło: opracowanie własne)	14
1.2	Wyznaczanie obrazu całkowego ii dla fragmentu obrazu f . (Źródło: opracowanie własne)	16
1.3	Wyznaczanie przyrostu obrazu całkowego. (Źródło: opracowanie własne)	17
2.1	Przykłady rekonstrukcji obrazów wykorzystującej wielomiany Zernike'a. (Źródło: opracowanie własne)	27
2.2	Części rzeczywiste ortogonalnych baz Zernike'a i Fouriera–Mellina zdefiniowane na kole jednostkowym. (Źródło: opracowanie własne)	30
2.3	Części urojone ortogonalnych baz Zernike'a i Fouriera–Mellina zdefiniowane na kole jednostkowym. (Źródło: opracowanie własne)	30
2.4	Wielomiany radialne baz Zernike'a oraz Fouriera–Mellina. (Źródło: opracowanie własne)	31
2.5	Ilustracja tłumienia informacji (z zastosowaniem proporcji 10%:90%) dla: momentów zespolonych (CM), momentów Zernike'a (ZM) i ortogonalnych momentów Fouriera–Mellina (OFMM). (Źródło: opracowanie własne)	34
3.1	Sztuczka z pracy (Singh i Upneja, 2012) pozwalająca na obliczenie OFMM: kwadratowe okno obrazu wpisane w koło jednostkowe, wartości 0 leżą wewnątrz koła poza kwadratem. (Źródło: opracowanie własne)	39
3.2	Ilustracja procedury detekcyjnej z wykorzystaniem okna przesuwanego. (Źródło: opracowanie własne)	41

3.3	Koncept pierścieni: (a) podział okna dla $R = 5$, (b) przykład dziedziny całkowania dla $r = 1$ i $h = 1$ (wariant z ‘dziurą’), (c) przykład dziedziny całkowania dla $r = 1$ i $h = 0$ (wariant bez ‘dziury’). (Źródło: opracowanie własne)	49
4.1	Litery i tła użyte do tworzenia zbioru „Litera A”. Pozytywy: litery A (a), negatywy: pozostałe litery (b), tła (c). (Źródło: opracowanie własne)	57
4.2	„Litera A”: krzywe ROC dla danych walidacyjnych. (Źródło: opracowanie własne)	58
4.3	„Litera A”: przykłady detekcji. Ostatni wiersz pokazuje brak wskazania, czyli przykład fałszywego negatywu (niewykryta litera A na pierwszym obrazie), oraz dwa obrazy z nieprawidłowymi wskazaniem, czyli przykładami fałszywych pozytywów (litera K oraz P wskazane jako szukana litera A). (Źródło: opracowanie własne)	59
4.4	Przykłady próbek ze zbioru uczącego. Pozytywy: samoloty (a), negatywy: (b), „trudne” negatywy: (c). (Źródło: opracowanie własne)	61
4.5	„Samoloty”: krzywe ROC dla danych walidacyjnych. (Źródło: opracowanie własne)	62
5.1	Podział obrazów całkowych. W ogólności, przyrost obrazu całkowego może wymagać złożenia z czterech składników związanych z częściami P_1, P_2, P_3, P_4 przy użyciu redefiniowanych współrzędnych indeksu centralnego (j_c, k_c) . (Źródło: opracowanie własne)	67
5.2	Przykładowy obraz wejściowy rozdzielczości 640×480 . (Źródło: opracowanie własne)	70
5.3	Odsetek cech z błędem względnym wynoszącym przynajmniej 0.25 dla każdej możliwej pozycji okna przesuwnego 48×48 dla obrazu wejściowego 640×480 — (po lewej) oryginalne podejście, (po prawej) podejście z podziałami $W = 172$. Nastawy cech $[\rho, \varrho, R] = [12, 12, 1]$. (Źródło: opracowanie własne)	70
5.4	Odsetek cech z błędem względnym wynoszącym przynajmniej 0.25 dla każdej możliwej pozycji okna przesuwnego 68×68 dla obrazu wejściowego 640×480 — (po lewej) oryginalne podejście, (po prawej) podejście z podziałami $W = 172$. Nastawy cech $[\rho, \varrho, R] = [12, 12, 1]$. (Źródło: opracowanie własne)	70

5.5	Odsetek cech z błędem względnym wynoszącym przynajmniej 0.25 dla każdej możliwej pozycji okna przesuwonego 98×98 dla obrazu wejściowego 640×480 — (po lewej) oryginalne podejście, (po prawej) podejście z podziałami $W = 172$. Nastawy cech $[\rho, \varrho, R] = [12, 12, 1]$. (Źródło: opracowanie własne)	71
7.1	„Litera A”: krzywe ROC dla danych walidacyjnych. (Źródło: opracowanie własne)	85
7.2	‘Litera A’: przykłady detekcji. (Źródło: opracowanie własne)	86
7.3	‘Litera A’: przykłady detekcji. W ostatnim wierszu znajdują się przykłady z brakiem wskazań — fałszywe negatywy (niewykryte litery A). (Źródło: opracowanie własne)	87
7.4	„Samoloty”: krzywe ROC dla danych walidacyjnych. (Źródło: opracowanie własne)	88
7.5	„Samoloty”: wycinki krzywych ROC dla danych walidacyjnych. (Źródło: opracowanie własne)	88
7.6	„Samoloty”: przykłady detekcji. Poprawne wskazania. (Źródło: opracowanie własne)	89
7.7	„Samoloty”: przykłady detekcji. Poprawne wskazania. (Źródło: opracowanie własne)	90
7.8	„Samoloty”: przykłady detekcji. Błędne wskazania lub ich brak. (Źródło: opracowanie własne)	91
7.9	„Samoloty”: przykłady detekcji. Błędne wskazania lub ich brak. (Źródło: opracowanie własne)	92

Spis tablic

3.1	Liczba obrazów całkowych zgodnie ze wzorem (3.24).	47
3.2	Liczba operacji przyrostu Δ biorących udział w obliczeniach wartości wzoru (3.18).	48
4.1	„Litera A”: nastawy eksperymentu.	57
4.2	Wyniki dla zbioru walidacyjnego: „Litera A”.	58
4.3	Wyniki dla zbioru testowego: „Litera A”.	59
4.4	„Samoloty”: nastawy eksperymentu.	61
5.1	Odsetki możliwie bezużytecznych cech (z błędami względnymi przynajmniej 0.1, 0.25, 0.5) dla obrazu 640×480 i różnych ustawień przestrzeni cech oraz różnych rozmiarów okna przesuwne. Wartości odsetków zostały uśrednione po wszystkich możliwych pozycjach okna. Mniejsze wartości zostały zaznaczone kolorem szarym.	71
5.2	Rzędy wielkości składników biorących udział w obliczaniu pojedynczego ZM. Przykład dla $p = 10$, $q = 2$ i okna o punkcie zaczepienia $(j_0, k_0) = (256, 462) = (1 \cdot W + 50, 2 \cdot W + 50)$ dla $W = 206$	72
6.1	Liczba cech w rozszerzonej (czarne) i podstawowej (szare) przestrzeni cech. Wersja z pierścieniami nie została uwzględniona w tablicy. . . .	81
7.1	Wyniki dla zbioru walidacyjnego: „Litera A”.	85
7.2	Wyniki dla zbioru testowego: „Litera A”.	86
7.3	Wyniki dla zbioru walidacyjnego: „Samoloty”.	89
7.4	Wyniki dla zbioru testowego: „Samoloty”.	89
7.5	„Litera A”: średnie czasy analizy 200 okien obrazu przy pomocy różnych sposobów obliczania momentów Zernike’a (obliczenia równoległe na: Intel Core i7-4790K CPU 4.00GHz, 4/8 c/t).	92
7.6	„Litera A” (ZM-M, podejście podstawowe): wydajność czasowa dla obrazu o wymiarach 640×480 (obliczenia równoległe na: Intel Core i7-4790K CPU 4.00GHz, 4/8 c/t).	93

7.7	„Litera A” (ZMs-M-NER, podejście z podziałami): wydajność czasowa dla obrazu o wymiarach 640×480 (obliczenia równoległe na: Intel Core i7-4790K CPU 4.00GHz, 4/8 c/t).	93
7.8	„Litera A” (ZM-E, podejście podstawowe): wydajność czasowa dla obrazu o wymiarach 640×480 (obliczenia równoległe na: Intel Core i7-4790K CPU 4.00GHz, 4/8 c/t).	94
7.9	„Litera A” (ZM-E-NER, podejście z podziałami): wydajność czasowa dla obrazu o wymiarach 640×480 (obliczenia równoległe na: Intel Core i7-4790K CPU 4.00GHz, 4/8 c/t).	94

Spis algorytmów

1	Schemat procedury detekcyjnej.	15
2	Algorytm uczący klasyfikatora Discrete AdaBoost.	18
3	Algorytm uczący klasyfikatora Real AdaBoost.	19
4	Ekstrakcja cech opartych na ZM z użyciem tablic.	53
5	Ekstrakcja pojedynczej cechy opartej na ZM z użyciem tablic.	54
6	Obraz całkowity zdefiniowany kawałkami.	66
7	Indeksy dla rozszerzonych niezmienników Zernike'a.	80
8	Ekstrakcja cech bazująca na rozszerzonej przestrzeni niezmienników Zernike'a.	82
9	Unikalne podindeksy występujące w rozszerzonych niezmiennikach Zernike'a.	83

Bibliografia

- Abu-Mostafa, Y. i Psaltis, D. (1984). “Recognitive Aspects of Moment Invariants”. W: *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-6.6, s. 698–706.
- Acasandrei, L. i Barriga, A. (2014). “Embedded Face Detection Application Based on Local Binary Patterns”. W: *2014 IEEE Intl Conf on High Performance Computing and Communications (HPCC,CSS,ICSS)*, s. 641–644.
- Ballard, D. H. (1981). “Generalizing the Hough transform to detect arbitrary shapes”. W: *Pattern Recognition* 13.2, s. 111–122. ISSN: 0031-3203.
- Bera, A., Kłesk, P. i Sychel, D. (2019). “Constant-Time Calculation of Zernike Moments for Detection with Rotational Invariance”. W: *IEEE Transactions on Pattern Analysis and Machine Intelligence*.
- Costea, A. i Nedeveschi, S. (2014). “Word Channel Based Multiscale Pedestrian Detection without Image Resizing and Using Only One Classifier”. W: *2014 IEEE Conference on Computer Vision and Pattern Recognition*, s. 2393–2400.
- Crow, F. (1984). “Summed-area Tables for Texture Mapping”. W: *SIGGRAPH Comput. Graph.* 18.3, s. 207–212. ISSN: 0097-8930.
- Dalal, N. i Triggs, B. (2005). “Histograms of Oriented Gradients for Human Detection”. W: *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05) - Volume 1*. IEEE Computer Society, s. 886–893.
- de Campos, T., Babu, B. i Varma, M. (2009). “Character Recognition in Natural Images”. W: *VISAPP 2009 - Proceedings of the 4th International Conference on Computer Vision Theory and Applications*. T. 2, s. 273–280.
- Ding, J. i in. (czer. 2019). “Learning RoI Transformer for Detecting Oriented Objects in Aerial Images”. W: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Duda, R. O. i Hart, P. E. (sty. 1972). “Use of the Hough Transformation to Detect Lines and Curves in Pictures”. W: *Commun. ACM* 15.1, s. 11–15. ISSN: 0001-0782.

- Freund, Y. i Schapire, R. E. (1999). “A Short Introduction to Boosting”. W: *In Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*. Morgan Kaufmann, s. 1401–1406.
- Friedman, J., Hastie, T. i Tibshirani, R. (2000). “Additive logistic regression: a statistical view of boosting”. W: *The Annals of Statistics* 28.2, s. 337–407.
- Gaddigoudar, P. K. i in. (2017). “Pedestrian detection and tracking using particle filtering”. W: *2017 International Conference on Computing, Communication and Automation (ICCCA)*, s. 110–115.
- Goldberg, D. (mar. 1991). “What Every Computer Scientist Should Know About Floating-point Arithmetic”. W: *ACM Comput. Surv.* 23.1, s. 5–48.
- Gu, J. i in. (2002). “A novel algorithm for fast computation of Zernike moments”. W: *Pattern Recognition* 35.12, s. 2905–2911.
- Gueham, M. i in. (lip. 2008). “Automatic Recognition of Shoeprints using Fourier-Mellin Transform”. W: *2008 NASA/ESA Conference on Adaptive Hardware and Systems*, s. 487–491. ISBN: 978-0-7695-3166-3.
- Gurbină, M., Lascu, M. i Lascu, D. (2019). “Tumor Detection and Classification of MRI Brain Image using Different Wavelet Transforms and Support Vector Machines”. W: *2019 42nd International Conference on Telecommunications and Signal Processing (TSP)*, s. 505–508.
- Hechri, A. i Abdellatif, M. (mar. 2012). “Automatic Detection and Recognition of Road Sign for Driver Assistance System”. W: *2012 16th IEEE Mediterranean Electrotechnical Conference*, s. 888–891.
- Hosny, K. M., Shouman, M. A. i Salam, H. M. A. (2011). “Fast computation of orthogonal Fourier–Mellin moments in polar coordinates.” W: *J. Real-Time Image Processing* 6.2, s. 73–80.
- Hough, P. (grud. 1962). *Method and means for recognizing complex patterns*. U.S. Patent 3069654.
- Hu, M.-K. (1962). “Visual pattern recognition by moment invariants”. W: *IRE Transactions on Information Theory* 8.2, s. 179–187.
- Huang, C.-J. H. i in. (2005). “Application of machine learning techniques to Web-based intelligent learning diagnosis system”. W: *Fourth International Conference on Hybrid Intelligent Systems (HIS'04)*, s. 242–247.
- Huang, C. i Huang, J. (mar. 2017). “A Fast HOG Descriptor Using Lookup Table and Integral Image”. W: *arXiv: Computer Vision and Pattern Recognition*.
- Huffman, A. D. (1971). “Impossible objects as nonsense sentences”. W: *Machine Intelligence*.

- Iscan, Z., Dokur, Z. i Ölmez, T. (mar. 2010). “Tumor Detection by Using Zernike Moments on Segmented Magnetic Resonance Brain Images”. W: *Expert Syst. Appl.* 37.3, s. 2540–2549. ISSN: 0957-4174.
- Juan, L. i Gwun, O. (2009). “A Comparison of SIFT, PCA-SIFT and SURF”. W: *International Journal of Image Processing* 3.4, s. 143–152.
- Kan, C. i Srinath, M. (sty. 2002). “Invariant character recognition with Zernike and orthogonal Fourier-Mellin moments”. W: *Pattern Recognition* 35, s. 143–154.
- Karimunnisa, S. i in. (2020). “Detection of Bone Fractures Automatically with Enhanced Performance with Better Combination of Filtering and Neural Networks”. W: *2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA)*, s. 189–193.
- Kleşk, P. (2017). “Constant-Time Fourier Moments for Face Detection — Can Accuracy of Haar-Like Features Be Beaten?” W: *Artificial Intelligence and Soft Computing: 16th International Conference, ICAISC 2017*. T. 10245.I. Springer Int. Publishing, s. 530–543.
- Kleşk, P., Bera, A. i Sychel, D. (2020). “Reduction of Numerical Errors in Zernike Invariants Computed via Complex-Valued Integral Images”. W: *Computational Science – ICCS 2020*. Springer International Publishing, s. 327–341. ISBN: 978-3-030-50420-5.
- Kleşk, P., Godziuk, A. i in. (2015). “Fast Analysis of C-Scans From Ground Penetrating Radar via 3-D Haar-Like Features With Application to Landmine Detection”. W: *IEEE Transactions on Geoscience and Remote Sensing* 53.7, s. 3996–4009.
- Lewis, J. (list. 1995). “Fast Template Matching”. W: *Vision Interface* 95, s. 120–123.
- Liang, X. i in. (2020). “Star Identification Algorithm Based on Image Normalization and Zernike Moments”. W: *IEEE Access* 8, s. 29228–29237.
- Liao, S., Jain, A. K. i Li, S. Z. (2016). “A Fast and Accurate Unconstrained Face Detector”. W: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38.2, s. 211–223.
- Liu, Q., Zhu, H. i Li, Q. (2011). “Object recognition by combined invariants of orthogonal Fourier-Mellin moments”. W: *2011 8th International Conference on Information, Communications Signal Processing*, s. 1–5.
- Mai, J. i Ning, M. (lip. 2015). “Sub-pixel edge detection method based on Zernike moment”. W: *The 27th Chinese Control and Decision Conference (2015 CCDC)*, s. 3673–3676.

- Malek, M. E., Azimifar, Z. i Boostani, R. (2017). “Facial age estimation using Zernike moments and multi-layer perceptron”. W: *22nd Int. Conference on Digital Signal Processing (DSP)*, s. 1–5.
- Meena, G., Sharma, D. i Mahrishi, M. (2020). “Traffic Prediction for Intelligent Transportation System using Machine Learning”. W: *2020 3rd International Conference on Emerging Technologies in Computer Engineering: Machine Learning and Internet of Things (ICETCE)*, s. 145–148.
- Ming, A., Ma, H. i Ling, C. X. (2012). “Object Tracking in Frame-Skipping Video Acquired Using Wireless Consumer Cameras”. W: *International Journal of Advanced Robotic Systems* 9.4, s. 131.
- Mohri, M., Rostamizadeh, A. i Talwalkar, A. (2012). *Foundations of Machine Learning*. Adaptive Computation and Machine Learning series. MIT Press. ISBN: 9780262018258.
- Mukundan, R. i Ramakrishnan, K. (1998). *Moment Functions in Image Analysis — Theory and Applications*. World Scientific. ISBN: 9789810235246.
- Pahwa, K. i Agarwal, N. (2019). “Stock Market Analysis using Supervised Machine Learning”. W: *2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon)*, s. 197–200.
- Papageorgiou, C., Oren, M. i Poggio, T. (1998). “A general framework for object detection”. W: *Computer Vision, 1998. Sixth International Conference on*, s. 555–562.
- Papakostas, G. i in. (2007a). “Fast Computation of Orthogonal Fourier-Mellin Moments Using Modified Direct Method”. W: *Systems, Signals and Image Processing, 2007 and 6th EURASIP Conference*, s. 153–156.
- Papakostas, G. i in. (2007b). “Fast numerically stable computation of orthogonal Fourier–Mellin moments”. W: *Computer Vision, IET* 1.1, s. 11–16. ISSN: 1751-9632.
- Pu, G. i in. (2021). “A hybrid unsupervised clustering-based anomaly detection method”. W: *Tsinghua Science and Technology* 26.2, s. 146–153.
- Qadri, M. T. i Asif, M. (2009). “Automatic Number Plate Recognition System for Vehicle Identification Using Optical Character Recognition”. W: *2009 International Conference on Education Technology and Computer*, s. 335–338.
- Quinlan, J. R. (mar. 1986). “Induction of Decision Trees”. W: *Machine Learning* 1.1, s. 81–106. ISSN: 0885-6125.
- Rajaraman, V. (sty. 2016). “IEEE standard for floating point numbers”. W: *Resonance* 21.1, s. 11–30.

- Rasolzadeh, B., Petersson, L. i Pettersson, N. (2006). “Response Binning: Improved Weak Classifiers for Boosting”. W: *IEEE Intelligent Vehicles Symposium*, s. 344–349.
- Redmon, J. i Farhadi, A. (2017). “YOLO9000: Better, Faster, Stronger”. W: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, s. 6517–6525.
- Redmon, J. i Farhadi, A. (2018). “YOLOv3: An Incremental Improvement”. W: *arXiv: Computer Vision and Pattern Recognition*.
- Ren, H. i Li, Z. (2015). “Object Detection Using Generalization and Efficiency Balanced Co-Occurrence Features”. W: *2015 IEEE International Conference on Computer Vision (ICCV)*, s. 46–54.
- Ren, S. i in. (2017). “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”. W: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39.6, s. 1137–1149.
- Rokach, L. i Maimon, O. (2005). “Top-down induction of decision trees classifiers - a survey”. W: *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 35.4, s. 476–487.
- Ross, M. i in. (2013). “Using Support Vector Machines to Classify Student Attentiveness for the Development of Personalized Learning Systems”. W: *2013 12th International Conference on Machine Learning and Applications*. T. 1, s. 325–328.
- Rowley, H. A., Baluja, S. i Kanade, T. (1998). “Neural network-based face detection”. W: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20.1, s. 23–38.
- Said, Y., Atri, M. i Tourki, R. (2011). “Human detection based on integral Histograms of Oriented Gradients and SVM”. W: *Communications, Computing and Control Applications (CCCA 2011)*. IEEE, s. 1–5.
- Schapire, R. E. (lip. 1990). “The Strength of Weak Learnability”. W: *Machine Learning* 5.2, s. 197–227. ISSN: 0885-6125.
- Schneiderman, H. i Kanade, T. (2000). “A statistical method for 3D object detection applied to faces and cars”. W: *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No.PR00662)*. T. 1, 746–751 vol.1.
- Sheng, Y. i Shen, L. (1994). “Orthogonal Fourier-Mellin moments for invariant pattern recognition”. W: *Journal of the Optical Society of America A* 11.6, s. 1748–1757.
- Singh, C. i Upneja, R. (2012). “Accurate Computation of Orthogonal Fourier-Mellin Moments”. W: *Journal of Mathematical Imaging and Vision* 44.3, s. 411–431.

- Sornam, M., Kavitha, M. S. i Shalini, R. (2016). “Segmentation and classification of brain tumor using wavelet and Zernike based features on MRI”. W: *2016 IEEE Intl Conf on Advances in Computer Applications (ICACA)*, s. 166–169.
- Szeliski, R. (2011). *Computer Vision: Algorithms and Applications*. 1st. Springer-Verlag London. ISBN: 978-1-84882-934-3.
- Teh, C.-H. i Chin, R. T. (1988). “On image analysis by the methods of moments”. W: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 10.4, s. 496–513.
- Terrillon, J.-C. i in. (lut. 2000). “Invariant neural-network based face detection with orthogonal Fourier-Mellin moments”. W: *Proceedings 15th International Conference on Pattern Recognition. ICPR-2000*. T. 2, 993–1000 vol.2.
- Viola, P. i Jones, M. (2001). “Rapid Object Detection using a Boosted Cascade of Simple Features”. W: *Conference on Computer Vision and Pattern Recognition (CVPR’2001)*. IEEE, s. 511–518.
- Wallin, A. i Kubler, O. (1995). “Complete sets of complex Zernike moment invariants and the role of the pseudoinvariants”. W: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 17.11, s. 1106–1110.
- Wang, K., Wang, H. i Wang, J. (2020). “Terrain Matching by Fusing HOG With Zernike Moments”. W: *IEEE Transactions on Aerospace and Electronic Systems* 56.2, s. 1290–1300.
- Wu, J., Xin, J. i Zheng, N. (2015). “SVM learning from imbalanced microanueyrysm candidate datasets used feature selection by gini index”. W: *2015 IEEE International Conference on Information and Automation*, s. 1637–1641.
- Wu, J., Yang, S. i Zhang, L. (2011). “Pedestrian detection based on improved HOG feature and robust adaptive boosting algorithm”. W: *2011 4th International Congress on Image and Signal Processing*. T. 3, s. 1535–1539.
- Xia, G.-S. i in. (sty. 2018a). *DOTA: A Large-scale Dataset for Object DeTectiion in Aerial Images*. <https://captain-whu.github.io/DOTA/index.html>. [Dostęp 19-Maj-2020].
- Xia, G.-S. i in. (czer. 2018b). “DOTA: A Large-Scale Dataset for Object Detection in Aerial Images”. W: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Xing, M. i in. (2016). “Traffic sign detection and recognition using color standardization and Zernike moments”. W: *2016 Chinese Control and Decision Conference (CCDC)*, s. 5195–5198.
- Yin, Y., Meng, Z. i Li, S. (2017). “Feature extraction and image recognition for the electrical symbols based on Zernike moment”. W: *2017 IEEE 2nd Advanced In-*

- formation Technology, Electronic and Automation Control Conference (IAEAC)*, s. 1031–1035.
- Zernike, F. (1934). “Beugungstheorie des Schneidenverfahrens und seiner verbesserten Form, der Phasenkontrastmethode”. W: *Physica* 1.8, s. 668–704.
- Zhou, Z. i in. (2016). “Moving object detection based on Zernike moments”. W: *2016 5th International Conference on Computer Science and Network Technology (ICCSNT)*, s. 696–699.