

# **Społeczeństwo Globalnej Informacji**

**Społeczeństwo  
Globalnej Informacji**

**Komitet Redakcyjny**

przewodniczący

prof. dr hab. inż. Janusz FILIPIAK

członkowie

prof. dr hab. Tomasz GOBAN-KLAS

dr hab. inż. Andrzej R. PACH

dr hab. inż. Zdzisław PAPIR

prof. dr hab. inż. Ryszard TADEUSIEWICZ

sekretarz

dr inż. Maria WRONIEWICZ-RAMS



# **Komputerowa analiza i przetwarzanie obrazów**

Ryszard TADEUSIEWICZ  
Przemysław KOROHODA



Wydawnictwo  
Fundacji Postępu Telekomunikacji  
Kraków 1997

Recenzent:  
prof. dr hab. inż. Leszek RUTKOWSKI  
Skład komputerowy i przygotowanie wersji elektronicznej  
inż. Irena WOJDYŁO  
Korekta:  
mgr Anna KOŁEK

**ISBN 83-86476-15-X**

Druk: Drukarnia Narodowa  
Kraków, ul. Marszałka J. Piłsudskiego 19

# 1 Spis treści

<b>1 Wstęp .....</b>	<b>1</b>
<b>2 Metody pozyskiwania obrazów cyfrowych.....</b>	<b>11</b>
2.1 Struktura obrazów cyfrowych .....	11
2.1.1 2.1.1. Dyskretyzacja obrazu .....	11
2.1.2 Źródła sygnałów zamienianych na obrazy .....	14
2.1.3 Przechowywanie obrazów .....	15
2.1.4 Drukowanie obrazów .....	16
2.2 Zasady tworzenia obrazu cyfrowego.....	17
2.2.1 Obraz jako funkcja .....	17
2.2.2 Skutki dyskretyzacji obrazu .....	20
2.3 Urządzenie do wprowadzania obrazu .....	28
2.3.1 Ogólny schemat systemu wprowadzania obrazu .....	28
2.3.2 Przetwornik obrazu - <i>Frame Grabber</i> .....	33
2.3.3 Kamera .....	34
2.3.4 Przetwornik CCD .....	40
2.3.5 Automatyka kamery .....	43
2.4 Działanie Frame Grabbera .....	54
2.5 Poziomy dalszej obróbki obrazów .....	56
<b>3 Klasyczne metody komputerowego przetwarzania obrazu.....</b>	<b>58</b>
3.1 Podział i ogólna charakterystyka algorytmów przetwarzania obrazu ..	58
3.2 Przekształcenia geometryczne.....	60
3.3 Przekształcenia punktowe .....	62
3.3.1 Przekształcenia oparte na przetwarzaniu pojedynczych punktów .....	62
3.3.2 Przekształcenia oparte na arytmetycznym przeliczaniu pojedynczych punktów .....	66
3.3.3 Realizacja przekształceń punktowych z użyciem LUT .....	70
3.3.4 Wyrównywanie histogramu.....	71
3.3.5 Punktowe operacje wykonywane na dwu obrazach.....	76
3.3.6 Binaryzacja.....	80
3.4 Kontekstowa filtracja obrazu .....	83
3.4.1 Ogólne problemy filtracji cyfrowej .....	83
3.4.2 Filtry liniowe .....	85
3.4.2.1 Podstawowe własności filtra liniowego .....	85
3.4.2.2 Konwolucja .....	85
3.4.2.3 Konwolucje dyskretne .....	87

3.4.2.4 Filtry dolnoprzepustowe.....	89
3.4.2.5 Filtry górnoprzepustowe - gradienty .....	93
3.4.2.6 Filtry górnoprzepustowe wykrywające narożniki.....	102
3.4.2.7 Filtry górnoprzepustowe wykrywające krawędzie - laplasjany .	105
3.4.3 Filtry nieliniowe .....	109
3.4.3.1 Filtry kombinowane wykrywające krawędzie .....	109
3.4.3.2 Operacje logiczne.....	116
3.4.3.3 Filtry medianowe.....	119
3.4.3.4 Filtry adaptacyjne.....	124
3.5 Realizacja funkcji przetwarzania obrazów przez specjalizowane procesory sprzętowe .....	125
3.5.1 Struktura specjalizowanego systemu do przetwarzania obrazów .....	125
3.5.2 Procesory operacji punktowych .....	125
3.5.3 Procesor obliczający histogram.....	127
3.5.4 Procesory konwolucji.....	127
3.5.4.1 Ogólna struktura.....	127
3.5.4.2 Procesor HSP-48908.....	129
3.5.4.3 Procesor PDSP-16488.....	130
3.5.4.4 Procesor IMS-A110 .....	131
3.5.5 Moduł procesora logicznego .....	132
3.5.6 Procesor filtracji medianowej.....	133
<b>4 Transformacja Fouriera.....</b>	<b>135</b>
4.1 Wprowadzenie.....	135
4.2 Transformacja Fouriera dla cyfrowych sygnałów jednowymiarowych .....	136
4.3 Transformacja Fouriera dla obrazów cyfrowych.....	146
4.3.1 Symetrie w dwuwymiarowej transformacji Fouriera obrazu o rzeczywistych wartościach pikseli .....	153
4.3.2 Przykłady obrazów cyfrowych oraz ich <i>F</i> -obrazów .....	157
4.3.3 Przykłady charakterystycznych związków pomiędzy treścią obrazu i <i>F</i> -obrazem przedstawionym w postaci poziomów szarości .....	164
4.3.4 Zawartość informacji wizualnej w poszczególnych składowych <i>F</i> -obrazu .....	168
4.3.5 Filtracja obrazów w dziedzinie Fouriera .....	170
4.3.6 Niezerowa faza filtra .....	180
4.3.7 Odpowiedź impulsowa filtra .....	185
4.3.8 Filtracja splotowa.....	189
4.3.9 <i>F</i> -obrazy wybranych filtrów splotowych.....	193
4.3.10 Rozplatanie, czyli filtracja odwrotna.....	199
4.3.11 Podsumowanie .....	204
<b>5 Przekształcenia morfologiczne.....</b>	<b>205</b>
5.1 Uwagi wstępne .....	205

5.2 Ogólny algorytm przekształcenia morfologicznego.....	206
5.3 Typowe przekształcenia morfologiczne.....	207
5.3.1 Erozja .....	207
5.3.2 Dylatacja .....	212
5.3.3 Otwarcie i zamknięcie .....	216
5.3.4 Otwarcie i zamknięcie właściwe, automediana .....	220
5.3.5 Detekcja ekstremów .....	220
5.3.6 Ścienianie .....	221
5.3.6.1 Szkieletyzacja.....	223
5.3.6.2 Obcinanie gałęzi.....	227
5.4 Inne przekształcenia morfologiczne.....	231
5.4.1 Wyznaczanie centroidów .....	231
5.4.2 Pogrubianie .....	231
5.4.3 Dylatacja bez stykania obszarów (SKIZ) .....	232
5.4.4 Wypukłe otoczenie .....	235
5.4.5 Przekształcenie trafi - nie trafi.....	235
5.5 Alfabet Golay'a.....	236
5.6 Złożone przekształcenia morfologiczne.....	238
5.6.1 Rekonstrukcja.....	238
5.6.2 Czyszczenie brzegu.....	238
5.6.3 Zalewanie otworów .....	239
5.6.4 Funkcja odległości .....	241
5.6.5 Erozja warunkowa.....	242
<b>6 Analiza obrazu.....</b>	<b>243</b>
6.1 Uwagi wstępne .....	243
6.2 Techniki segmentacji .....	243
6.2.1 Podział technik segmentacji .....	243
6.2.2 Segmentacja przez podział obszaru.....	244
6.2.3 Segmentacja metodą wykrywania krawędzi .....	245
6.2.4 Segmentacja przez rozrost obszaru .....	246
6.2.5 Segmentacja oparta na statystyce .....	248
6.2.6 Segmentacja obiektów stykających się.....	248
6.3 Technika indeksacji obrazu.....	252
6.4 Pomiary .....	256
6.4.1 Rola i znaczenie pomiarów parametrów obiektów na obrazach.....	256
6.4.2 Liczebność elementów .....	257
6.4.3 Pole powierzchni.....	259
6.4.4 Długości krawędzi.....	259
6.4.5 Długości rzutów .....	260
6.4.6 Średnice Fereta.....	262
6.4.7 Momenty bezwładności.....	263

6.5 Współczynniki kształtu .....	263
6.5.1 Definicje współczynników kształtu .....	263
6.5.2 Własności współczynników kształtu .....	265
6.6 Wykrywanie obiektów ruchomych .....	269
<b>7 Literatura.....</b>	<b>271</b>

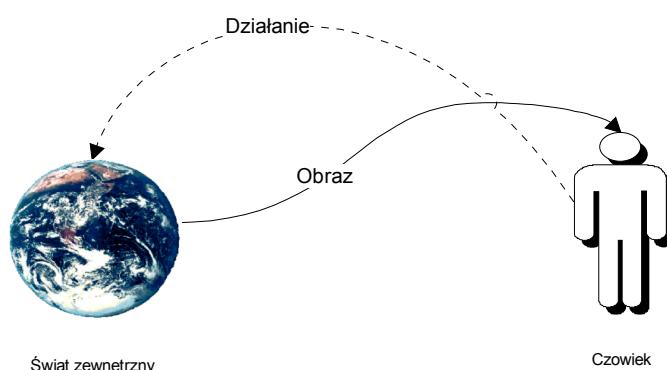
# 1 Wstęp

Jedną z najważniejszych umiejętności człowieka jest podejmowanie działań w zależności od wniosków wyciąganych z analizy bodźców odbieranych przez narządy zmysłów. W związku z tym automatyzacja wielu stanowisk pracy, czyli zastępowanie pracy człowieka działaniem odpowiednich systemów technicznych, musi odwoływać się także do automatyzacji procesu percepcji poprzez stosowanie odpowiednich systemów sensorycznych. Uwzględniając fakt, że najistotniejszym narządem zmysłu człowieka jest wzrok, ponieważ informacje przez niego uzyskiwane mają największy wpływ na nasze działanie w różnych okolicznościach (w tym także na różnych stanowiskach pracy), jako jedno z najpilniejszych zadań automatyki można wskazać zadanie stworzenia automatycznych systemów wizyjnych.

Systemy wizyjne zyskują w ostatnich latach coraz większą popularność. Znajdują zastosowanie w coraz większej ilości dziedzin. Stosowane są w wyposażeniu robotów, przy nadzorowaniu procesów przemysłowych, sterowaniu ruchem drogowym, itp. Operacje przetwarzania obrazu pozwalają na łatwą analizę zdjęć satelitarnych powierzchni Ziemi. Przez zastosowanie odpowiedniej operacji przetwarzania obrazów można uzyskać informacje, które normalnie nie są rozróżniane przez system wzrokowy człowieka. W medycynie analiza obrazu jest często stosowana do przetwarzania obrazów organów wewnętrznych człowieka, ujawniając miejsca gdzie nastąpiły zmiany chorobowe u pacjenta. Kolejną dziedziną, w której często wykorzystuje się przetwarzanie obrazu, jest mineralogia. Inną, godną wymienienia (szczególnie na AGH), jest metalurgia - analizy obrazów znajdują zastosowanie w przeróbce surowców mineralnych i w procesie oceny próbek materiałów kompozytowych. Technikę komputerową w odniesieniu do obrazów wykorzystuje się także w celach artystycznych, czego najlepszym przykładem może być tworzenie teledysków.

W celu efektywnego wykorzystania obrazu jako źródła informacji należy przetworzyć go na postać cyfrową, a następnie przeprowadzić jego szczegółowy proces analizy, w skład którego wchodzą: segmentacja, lokalizacja obiektów oraz wyznaczanie ich cech. Jednak przed rozpoczęciem tego procesu konieczne jest polepszenie jakości obrazu, a w szczególności jego filtracja (eliminacja zakłóceń) oraz wyostrzanie. Te ostatnie operacje odwołują się do procesu przetwarzania obrazu. Dodatkowo przetwarzanie obrazu daje możliwość automatycznej regulacji położenia i wielkości badanego obiektu w kadrze, co pozwala na maksymalne wykorzystanie informacji, jaką możemy uzyskać z systemu wizyjnego.

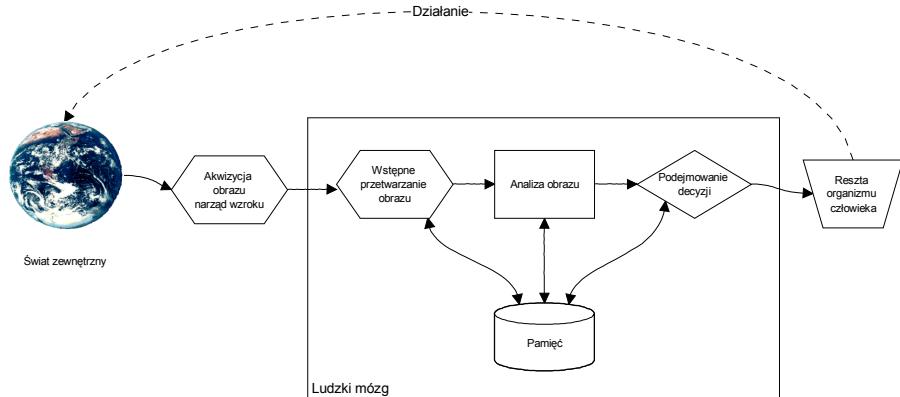
Książka ta zawiera podstawowe wiadomości na temat budowy i metod działania sztucznych systemów wizyjnych, wystarczające do tego, by rozpocząć działania w zakresie budowy takich systemów, ich programowania i eksploatacji. Pogłębienie i poszerzenie podanych w tym podręczniku informacji jest możliwe po wykorzystaniu wiadomości zawartych w cytowanych na końcu książki monografiach.



Rys. 1.1. Ogólny schemat naturalnego procesu przetwarzania obrazu<sup>1</sup>.

Zadania realizowane przez komputerowe systemy przetwarzania obrazów są zaskakująco złożone i trudne. Zaskoczenie wynika z tego, że każdy człowiek dokonuje czynności przetwarzania i rozpoznawania obrazów przez praktycznie cały czas swojej pracy i wypoczynku, przy czym czynności te nasz mózg wykonuje w sposób tak szybki naturalny, że aż nie zdajemy sobie sprawy z ogromnej złożoności tego procesu (patrz rys. 1.1). Uczymy się tej umiejętności pozyskiwania informacji za pomocą wzroku we wczesnym okresie niemowlęcym i doskonaliśmy ją przez całe życie, wydaje się więc ona czymś naturalnym i oczywistym. Tymczasem próba stworzenia sztucznego odpowiednika tego procesu ujawnia całą złożoność tego, co po prostu nazywamy patrzeniem i postrzeganiem (patrz rys. 1.2.). Zautomatyzowanie tego procesu jest więc niesłychanie trudne, przynieść jednak może wielkie korzyści. Z tego powodu, a także dlatego, że budowa zautomatyzowanych systemów wizyjnych stanowi jedno z najbardziej ambitnych zagadnień cybernetyki - warto się tym zajmować!

<sup>1</sup> Wiele rysunków w tej książce (także i przedstawiony wyżej rysunek nr 1.1) pochodzi z prac dyplomowych studentów wydziału Elektrotechniki, Automatyki i Elektroniki AGH, którzy pod opieką naukową autorów tej książki przygotowywali liczne opracowania na temat różnych szczegółowych zagadnień, programów i algorytmów związanych z analizą i przetwarzaniem obrazów. Wykaz tych prac dyplomowych podano na końcu książki w wykazie literatury. Wiele rysunków zaczerpnięto także z rozprawy habilitacyjnej dr inż. Kazimierza Wiatra z Katedry Elektroniki AGH. Autorom udostępnionych do wykorzystania ilustracji składamy w tym miejscu serdeczne podziękowania.



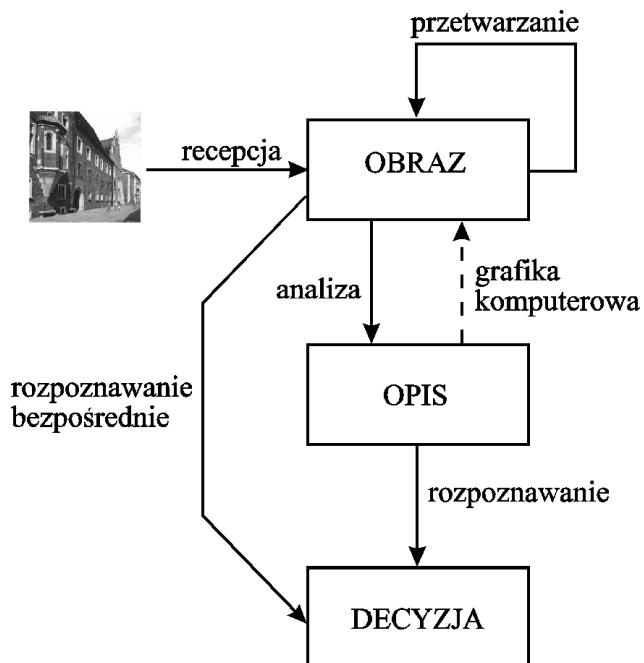
Rys. 1.2. Schemat naturalnego przetwarzania obrazu.

Celem sztucznego przetwarzania lub analizy obrazu jest takie automatyczne przetworzenie i przeanalizowanie obrazu wybranych obiektów lub całego otoczenia systemu zautomatyzowanego, aby uzyskać użyteczną informację na temat interesujących obiektów (na przykład będących przedmiotem manipulacji ze strony robota przemysłowego) lub na temat otoczenia, które może wpływać (i zwykle znacząco wpływa) na sterowany automatycznie proces.

Odnotejmy w tym miejscu pewne ważne stwierdzenie o charakterze strukturalnym. Otóż łatwo stwierdzić, że na proces widzenia (zarówno naturalnego, jak i sztucznego) składa się szereg operacji (patrz rys. 1.3), takich jak:

- recepcja (akwizycja) obrazu;
- przetwarzanie obrazu (filtracja wstępna, eliminacja zakłóceń, kompresja obrazu, eksponowanie ważnych cech, itp.);
- analiza obrazu (wydobycie cech opisujących obraz);
- rozpoznanie obrazu i jego semantyczna interpretacja.

Wymienione operacje uzupełnia jeszcze pokazana na rysunku 1.3 operacja bezpośredniego rozpoznawania obrazu (bez jego wstępnej analizy), co bywa praktykowane w prostych systemach rozpoznających z góry ustalone i raczej dość proste zestawy wzorów graficznych (jako przykład można tu przytoczyć tzw. zadania OCR - *Optical Character Recognition* - czyli automatycznego rozpoznawania znaków alfanumerycznych) oraz - nie omawiana w tej książce - problematyka grafiki komputerowej, będącej techniką w pewnym stopniu odwrotną w stosunku do techniki analizy obrazu. O ile bowiem w analizie obrazu chodzi o to, by na podstawie obrazu znaleźć taki jego opis, który w skróconej formie informuje o wszystkich istotnych cechach obiektów stanowiących treść obrazu – o tyle grafika komputerowa wychodzi od skróconego opisu jakichś obiektów i ich cech, a zadaniem komputera jest wizualizacja tych obiektów w postaci odpowiedniego (dwu- lub pseudotrójwymiarowego) obrazu.

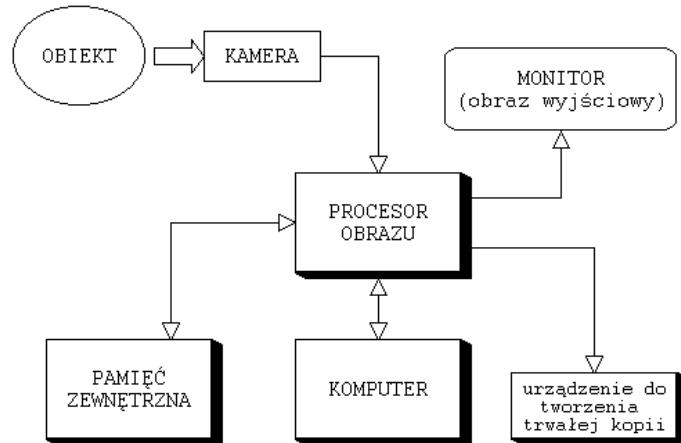


Rys. 1.3. Operacje składające się na proces automatycznego widzenia.

Najbardziej czasochłonny i angażujący największe moce obliczeniowe jest etap filtracji obrazu, zaś najbardziej skomplikowaną i trudną w realizacji operacją jest - wbrew pozorom - analiza obrazu. Natomiast rozpoznawanie obrazu, przez długi czas traktowane jako jeden z kluczowych problemów sztucznej inteligencji, w istocie nie zawiera aż tak wielu kłopotliwych szczegółów. Dlatego w tej książce najwięcej miejsca poświęcono zagadnieniom filtracji obrazów, dość obszernie potraktowano także zadania związane z ich analizą, natomiast bardzo skrótnie potraktowano zagadnienia związane z rozpoznawaniem (także i z tego powodu, że na temat rozpoznawania obrazów wydana została inna książka<sup>2</sup>). Omówimy teraz wymienione procesy nieco dokładniej, sygnalizując równocześnie, co będzie zawarte w poszczególnych, kolejnych rozdziałach książki. Posłużymy się przy tym ogólnym schematem systemu wizyjnego, pokazanym na rysunku 1.4. Wyróżnić na nim można następujące elementy:

- moduł wprowadzania obrazu,
- urządzenie do wyświetlania obrazów,
- urządzenie do tworzenia trwałej kopii obrazu,
- pamięć zewnętrzna,
- procesor obrazu.

<sup>2</sup> R. Tadeusiewicz, M. Flasiński: *Rozpoznawanie obrazów*, PWN, Warszawa 1991.



Rys. 1.4. System cyfrowego przetwarzania obrazu.

Procesor obrazu jest podstawowym elementem każdego systemu cyfrowego przetwarzania obrazu. W szczególności składa się on z zestawu urządzeń realizujących cztery podstawowe funkcje:

- przyjęcie obrazu
- zapamiętanie obrazu
- właściwą obróbkę (przetwarzanie i analizę) obrazu
- wyświetlenie obrazu.

Zadaniem modułu wejściowego w omawianym systemie jest zamiana obrazu na jego cyfrową reprezentację, akceptowaną przez procesor obrazu. Stosuje się dwa rozwiązania: kamery telewizyjne, które na wyjściu mają sygnał analogowy (najpowszechniej stosowane) oraz kamery pracujące w innych standardach mające na wyjściu sygnał cyfrowy, gotowe do podłączenia bezpośrednio do magistrali systemów cyfrowych. Dodatkowo obraz wprowadzany bywa do procesora obrazu bezpośrednio z pamięci zewnętrznej (na przykład z pamięci cyfrowego aparatu fotograficznego).

Do **urządzeń wyjściowych** należą wszelkiego rodzaju monitory oraz drukarki i inny sprzęt do tworzenia trwałych kopii (wydruków, fotografii, slajdów). Urządzeniami tymi nie będziemy się w tej książce zajmowali.

Obraz, po wprowadzeniu go w formie cyfrowej do komputera, wymaga z reguły różnych zabiegów doskonalących jego jakość, co powoduje, że konieczny jest proces **przetwarzania obrazu**. Cechą charakterystyczną procesu przetwarzania obrazu jest fakt, że zarówno na wejściu algorytmu przetwarzającego informacje, jak i na jego wyjściu występują **obrazy** (por. rys. 1.3). Jest to spostrzeżenie bogate w konsekwencje. Główną z nich jest konstatacja faktu, że algorytm przetwarzający, który musi wytworzyć dziesiątki tysięcy lub nawet miliony punktów

obrazu wynikowego, po prostu **musi** działać stosunkowo wolno na szeregowym (sekwencyjnym) komputerze. Jego pracę można wprawdzie znakomicie przyspieszyć przechodząc do komputerów o architekturze równoległej - jednak to zagadnienie nie jest szczegółowo podejmowane i rozwijane w tej książce.

Przy interaktywnym przetwarzaniu obrazu dobrze jest rozdzielić operacje graficzne między głównym (głównymi) procesorem systemu a procesorem graficznym znajdującym się w podsystemie adaptera graficznego. Taki specjalizowany procesor graficzny może posiadać moc przetwarzania obrazu znacznie przewyższającą główny procesor i wspomagać go w operacjach prezentacji obrazu użytkownikowi. Na etapie zastosowań przemysłowych z punktu widzenia ekonomicznego i praktycznego najlepiej jest zastosować specjalizowany system mikroprocesorowy, wykonujący część operacji sprzętowo. Generalnie zatem będziemy rozważać tzw. **procesor obrazu**, który na ogół jest po prostu odpowiednio oprogramowanym uniwersalnym komputerem, co umożliwia wykonywanie dowolnej liczby operacji na obrazie, uzależniając je jedynie od posiadanego oprogramowania i mocy obliczeniowej komputera. Czasami dodatkowo układ cyfrowego procesora obrazu jest realizowany (zwłaszcza w zakresie operacji wstępnego przetwarzania obrazu) w postaci odrębnego procesora na karcie wprowadzającej obraz do komputera (na przykład *Frame Grabber DT 2862*<sup>3</sup>).

Po procesie przetwarzania obrazu następuje jego **analiza**. Wymagania stawiane głównemu komputerowi przeznaczonemu do analizy obrazów to szybki procesor, duża pojemność pamięci RAM oraz znaczna przestrzeń adresowa procesora. Narzucającym się rozwiązaniem jest wykorzystanie przy analizie obrazu popularnych komputerów klasy PC. Takie rozwiązanie niesie wiele korzyści w stosunku do rozwiązań wymagających tworzenia specjalistycznych systemów do przetwarzania obrazów:

- jest znacznie tańsze od rozwiązań specjalizowanych dzięki masowości produkcji komputerów klasy PC;
- komputery PC są powszechnie dostępne;
- komputery PC są już obecnie bardzo rozwinięte technologicznie;
- komputery PC stwarzają większe możliwości konfiguracyjne przez różnorodność oferty urządzeń peryferyjnych i wyposażenia;
- używany sprzęt PC może być łatwo wykorzystywany także i w innej działalności.

Rozwiązanie oparte na komputerach PC jest szczególnie atrakcyjne na etapie badawczym, kiedy dopiero opracowywana jest teoria działania systemu i szaco-

---

<sup>3</sup> Karta ta zawiera moduły umożliwiające przyjęcie i dyskretyzację sygnału telewizyjnego, własną pamięć dla 4 kadrów, możliwość prostego przetwarzania obrazów 512×512 z 8-bitową dokładnością w czasie rzeczywistym, moduły wyjściowe i tzw. LUT (ang. *Look-Up Table*). Są to tablice konwersji, które umożliwiają przedefiniowanie macierzy jasności według ustalonego algorytmu (np. umożliwiają uzyskanie obrazu pozytywowego na podstawie negatywu).

wane są wymagania dla systemu i urządzeń współpracujących. System docelowy, mający (ewentualnie) duże znaczenie praktyczne i powielany w wielu egzemplarzach może być natomiast często korzystniej wykonany jako specjalizowany. Zapewni to większą jego efektywność i praktyczność.

Decyzja użycia w zadaniu przetwarzania i analizy obrazów komputerów klasy PC ma jednak także pewne mankamenty. W komputerach tych, najczęściej opartych na procesorach firmy *Intel*, występuje systemem operacyjny MS-DOS, który znacznie ogranicza możliwości wykorzystania pamięci. Aby ominąć tę niedogodność, stosuje się nowocześniejsze systemy operacyjne (np. Windows 97 lub Windows NT) albo wykorzystuje się odpowiednie rozwiązania programowo - sprzętowe (np. standard *Lotus Intel Microsoft Expanded Memory System* lub EMS), co jednak odbywa się zwykle kosztem spowolnienia obliczeń. Innym utrudnieniem wystającym w procesorach Intela jest sposób adresowania pamięci w blokach po 64kB, co utrudnia operowanie na tablicach o wymiarach  $512 \times 512$  lub większych. Mimo, że wad tych nie posiadają procesory z rodziny 68000 firmy Motorola (ciągłe adresowanie przestrzeni pamięciowej, możliwość tworzenia w pamięci dużych bloków danych) to powszechność stosowania standardu oraz znaczna liczba pakietów programowych narzuca stosowanie w systemach cyfrowego przetwarzania obrazów komputera IBM lub kompatybilnego.

Poszczególnym elementom opisanego schematu odpowiadać będą kolejne rozdziały książki (czasem jednemu elementowi poświęconych będzie kilka rozdziałów, co nie zmienia ogólnej zasady). I tak problematyce metod pozyskiwania i akwizycji obrazów cyfrowych poświęcony będzie **rozdział 2**. Jak już powiedziano, proces **recepkcji obrazu** wymaga zaangażowania dwojakiego rodzaju aparatury: systemu dokonującego zamiany dwu- lub trójwymiarowego obiektu na sygnały elektryczne (na przykład odpowiedniej kamery video) oraz systemu zamieniającego sygnał wizyjny na odpowiednią strukturę danych cyfrowych. Obydwie te grupy urządzeń i związane z nimi problemy omówione są we wskazanym rozdziale.

Jak już powiedziano, ważnym etapem jest **przetwarzanie** obrazu, którego wynikiem są inne obrazy, wolne od zakłóceń oraz o wyraźniej zaznaczonych interesujących nas cechach. Ponieważ zarówno cele, jak i metody przetwarzania obrazów mogą być bardzo zróżnicowane, zagadnieniom związanym z przetwarzaniem obrazu poświęcono aż trzy rozdziały książki. Najpierw omówiono „klasyczne” liniowe oraz wybrane nieliniowe, a także kontekstowe i bezkontekstowe metody przetwarzania obrazów. Metody te opisano w **rozdziale 3**. Filtrację obrazów często interpretuje się w kategoriach częstotliwościowych, więc mimo licznych kontrowersji związanych z oceną rzeczywistej użyteczności metod przetwarzania obrazów z wykorzystaniem przekształcenia Fouriera, w **rozdziale 4** przedstawiono zasadnicze informacje związane z wykorzystaniem transformacji widmowej do przetwarzania obrazów. Wreszcie w **rozdziale 5**

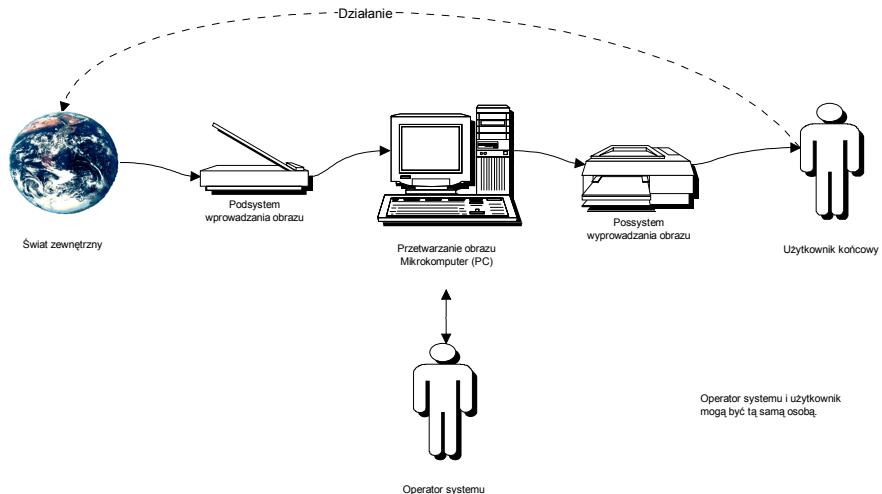
opisano metody przetwarzania obrazów wywodzące się z morfologii matematycznej.

Po etapie doskonalenia obrazu i jego przetwarzania w większości systemów zautomatyzowanych trzeba przejść do etapu jego **analizy**. Wynikiem analizy obrazu mogą być dane jakościowe i ilościowe, opisujące określone cechy obrazu lub całej grupy obrazów (na przykład sekwencji kolejnych kadrów na filmie video). Wybrane wiadomości na temat metod analizy obrazów zebrane w **rozdziale 6**. Podczas analizy obrazu gubiona jest bezpowrotnie pewna część informacji (gdyż pełny obraz, z setkami jego subtelnego szczegółów, zastąpiony zostaje w tym procesie dość ograniczonym zbiorem cech), jednak rezultat analizy może zawierać wszystkie informacje użyteczne, a trudne do wyodrębnienia “na pierwszy rzut oka” w obrazie wyjściowym. Dlatego obraz po procesie analizy jest nieporównanie bardziej podatny na zastosowanie metod i algorytmów rozpoznawania czy też na inne metody oceny jego merytorycznej zawartości, niż obraz nie poddany analizie.

Na zakończenie tego wstępu jeszcze kilka uwag na temat zastosowań komputerowych systemów wizyjnych (patrz rys. 1.5). Praktycznymi dziedzinami, w których istnieje obecnie pilna potrzeba „sztucznego widzenia”, w tym głównie przetwarzania, analizy i rozpoznawania obrazu, są między innymi:

- automatyka (polepszenie autonomiczności robotów wyręczających człowieka w wielu mechanicznych pracach, polepszenie jakości działania systemów regulacji automatycznej);
- medycyna (automatyczna analiza i rozpoznawanie obrazów preparatów tkankowych, rentgenogramów, ultrasonogramów, obrazów z tomografii rentgenowskiej i NMR, itp.);
- kryminalistyka (wyszukiwanie w obrazie cech i szczegółów nie rozróżnialnych ludzkim wzrokiem);
- geodezja i kartografia (automatyczne przetwarzanie wielkich ilości danych obrazowych - zwłaszcza zdjęcia lotnicze i satelitarne);
- komunikacja (wykrywanie obecności, kierunku i natężenia ruchu pojazdów, określanie wielkości kolejki pojazdów czekających na skrzyżowaniach, automatyczne wykrywanie kolizji i wypadków);
- laboratoria badawcze (kontrola materiałów, kontrola jakości wyrobów);
- zabezpieczanie obiektów pod szczególnym nadzorem (kontrola bagażu, nadzór nad ruchem w pewnym obszarze, wykrywanie intruzów);
- wojskowość (kierowanie naprowadzaniem inteligentnej broni na współczesnym polu walki);
- astronomia i astrofizyka (analiza obrazów o zakresie widmowym przekraczającym możliwości ludzkiego wzroku);

- metrologia (pomiary dokonywane na podstawie obrazu).



Rys. 1.5. Ogólny schemat zastosowania techniki przetwarzania obrazu przy pomocy systemu komputerowego.

Analizując zastosowania technik komputerowego przetwarzania obrazów w wymienionych wyżej dziedzinach można stwierdzić, że mimo wielu jeszcze niedoskonałości sztuczne przetwarzanie obrazu może mieć następujące cechy prze- wyższające ludzki zmysł wzroku:

- niższa cena analizy,
- szybszy czas reakcji,
- powtarzalność wyników,
- możliwość automatycznego rejestrowania wyników,
- możliwość dowolnego programowania procesu przetwarzania,
- brak czynników ludzkich takich jak zmęczenie, znudzenie, stres, podatność na przekupstwo, itp.,
- praca w dowolnym zakresie zarówno światła widzialnego, jak i podczerwieni czy nadfioletołu,
- praca w miejscach niedostępnych dla człowieka,
- możliwość łatwego i taniego powielania sprawdzonych rozwiązań,
- brak instynktu samozachowawczego (broń),
- brak oporów moralnych przed zniszczeniem.

Na koniec kilka uwag porządkowych.

Książka ta powstała w oparciu o doświadczenia ponad dziesięciu lat wykładów, ćwiczeń i laboratoriów studenckich, prowadzonych przez obydwu autorów w ra-

mach szeregu przedmiotów na kilku wydziałach AGH. Wykorzystano w niej także doświadczenia z blisko stu prac magisterskich i kilku doktorskich, przygotowanych pod kierunkiem pierwszego z autorów w latach 1980 - 1997. Na wyborze materiału do prezentacji w książce i na wyborze samego sposobu prezentacji zaważyło więc dość bogate doświadczenie, pokazujące, które zagadnienia z obszernej problematyki *computer vision* są najbardziej potrzebne, które sprawiają kłopoty, jakie są najczęstsze źródła nieporozumień, niepowodzeń i rozczarowań przy próbach praktycznych zastosowaniach - słowem, co **koniecznie** trzeba wiedzieć, by skutecznie działać w dziedzinie komputerowego przetwarzania i analizy obrazów. Równocześnie trzeba jednak podkreślić, że celem książki jest wprowadzenie Czytelnika w **podstawowe** problemy związane z przetwarzaniem i analizą obrazów, dlatego dla bardzo wielu zagadnień szczegółowych zabrakło w książce miejsca. W związku z tym specyfika różnych zastosowań przetwarzania obrazów (na przykład zdjęć lotniczych), analizy obrazów (na przykład medycznych), ich wykorzystania (na przykład w robotyce) oraz rozpoznawania w specyficznych zadaniach (na przykład automatycznego rozpoznawania pisma) w minimalnym tylko stopniu została uwzględniona przy pisaniu tej książki, która może być traktowana jako wprowadzenie do tematu, w żadnym jednak przypadku nie powinna pozostać jedynym źródłem potrzebnych informacji!

## 2 Metody pozyskiwania obrazów cyfrowych

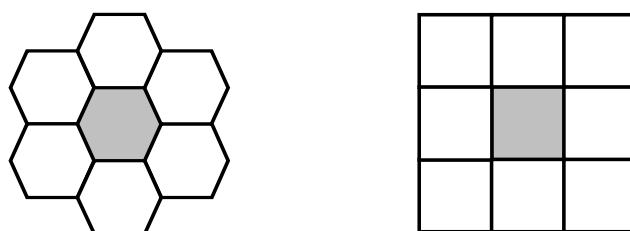
### 2.1 Struktura obrazów cyfrowych

#### 2.1.1 Dyskretyzacja obrazu

Wyjściowym zagadnieniem przy przetwarzaniu i analizie obrazu jest odpowiednie zdefiniowanie sztucznej reprezentacji obrazu. Obraz jest notacją "wyglądu" otaczającego świata. Złożone komputerowe przetwarzanie obrazu zakłada dyskretną reprezentację obrazu<sup>4</sup>. Jednak rozpatrywana, sztuczna reprezentacja obrazu musi być znacznie uproszczona w stosunku do ludzkiego zmysłu wzroku, który z kolei ustępuje wielu innym, występującym w przyrodzie (na przykład znana jest niewiarygodna ostrość wzroku niektórych ptaków drapieżnych). Ograniczenie rozmiaru rozważanego tu obrazu cyfrowego wynika ze znikomej wydajności współczesnych systemów komputerowych w stosunku do sprawności ludzkiego mózgu. Mózg ten przetwarza w czasie rzeczywistym strumień danych o wielkości co najmniej 100 Mb/s związany ze zmysłem wzroku. Jest to zdecydowanie za dużo, jak na możliwości przetwarzania dostępnych komputerów. Ograniczenia reprezentacji obrazu mogą być realizowane jednocześnie na wielu płaszczyznach:

- ograniczenie zdolności rozpoznawania szczegółów;
- ograniczenie ilości możliwych do rozróżniania stanów elementu obrazu (kolorów);
- analizowanie obrazu płaskiego zamiast przestrzennego;
- analizowanie obrazu statycznego zamiast dynamicznego.

W obecnych systemach przetwarzania obrazu wykorzystuje się generalnie dwa sposoby rozmieszczenia cyfrowych elementów obrazu: według siatki heksagonalnej lub kwadratowej (rys. 2.1).

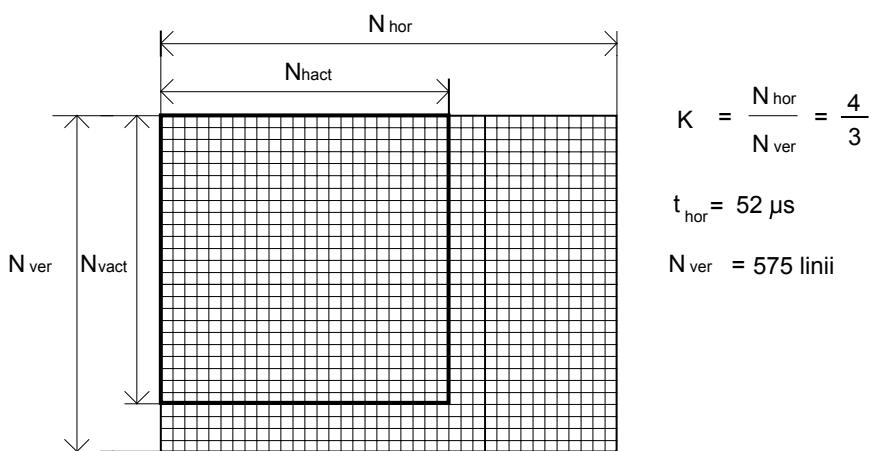


Rys. 2.1. Struktura siatki heksagonalnej i siatki kwadratowej.

---

<sup>4</sup> Reprezentacja obrazu wykorzystywana przez ludzki wzrok również jest skwantowana, przy najmniej, jeśli chodzi o relacje przestrzenne.

Sposób pierwszy jest bardziej zbliżony do naturalnego układu receptorów w nacządrze wzroku człowieka, ale drugi jest znacznie prostszy i wygodniejszy w obsłudze, dlatego jest on (jako tzw. *raster kwadratowy*) bardziej rozpowszechniony w komputerowej analizie obrazu. Wynika stąd, że przeważająca ilość urządzeń wprowadzających lub wyprowadzających obraz z komputera posługuje się siatką kwadratową. Wszystkie powszechnie wykorzystywane formaty graficzne również opierają się na niej. W niniejszej pracy ograniczymy się zatem tylko do modeli opartych o siatkę kwadratową (rys. 2.2).



Rys. 2.2. Struktura siatki (rastra) typowego obrazu cyfrowego.

Wybrane rozmiary siatki (rastra) obrazu cyfrowego dyktują częstotliwość próbkowania sygnału wizyjnego. Częstotliwość ta, będącą głównym parametrem używanego przetwornika A/C, może być wyznaczona z następującej formuły:

$$f = \frac{K * N_{ver}}{t_{hor}} = 14,75 \text{ MHz}$$

gdzie:  $K = N_{hor} / N_{ver} = 4/3$  - proporcje obrazu,

$N_{hor} = 575$  - liczba widocznych na ekranie linii poziomych,

$t_{hor} = 52 \mu s$  - aktywny czas wybierania linii (PAL).

Wybrany określony typ rastra musimy zdecydować, jak wiele elementów ma ten raster zawierać, czyli ustalić, jak duża ma być **rozdzielcość obrazu**<sup>5</sup>. Wybór właściwej rozdzielcości obrazu jest sprawą bardzo ważną, gdyż rozdzielcość jest miarą zdolności rozpoznawania szczegółów obrazu. Dobór właściwej rozdzielcości jest zawsze kompromisem. Z jednej strony - im większej rozdzielcości jest obraz, tym więcej szczegółów zawiera i więcej informacji można

<sup>5</sup> Wyraża się ona ilością elementów podstawowych składających się na obraz. Najczęściej przy płaskich obrazach o kwadratowej siatce zapisywana jest ona jako iloczyn ilości elementów w poziomie i pionie obrazu.

z niego odczytać, a z drugiej strony - liniowy wzrost rozdzielczości obrazu pociąga za sobą kwadratowy wzrost czasu przetwarzania i objętości samej reprezentacji. Sposób postępowania jest zazwyczaj następujący: pozyskuje się obraz o stosunkowo dużej rozdzielczości, by w trakcie kolejnych, wstępnych etapów obróbki obrazu dojść do rozdzielczości mniejszej, praktycznej (patrz podrozdział 2.2). Najczęściej spotykane rozdzielczości w analizie obrazu to całkowite potęgi dwójki<sup>6</sup> od 256x256 do 4096x4096.

Specjalizowane systemy przetwarzania obrazu wykorzystują zazwyczaj jedną, ustaloną rozdzielczość obrazu. Większość systemów uniwersalnych umożliwia jednak analizę obrazów o dowolnej rozdzielczości.

Każdy z elementów dyskretnej reprezentacji obrazu może przyjmować tylko jeden spośród ograniczonej ilości stanów. Ilość ta, popularnie nazywana ilością kolorów, może być także w komputerowej reprezentacji obrazu interpretowana jako ilość bitów przeznaczonych na zapamiętanie stanu jednego elementu obrazu (**bpp**<sup>7</sup>).

Dobór właściwej, z punktu widzenia przetwarzania obrazu, ilości kolorów jest także kompromisem pomiędzy jakością obrazu, a co za tym idzie ilością ukrytej w nim informacji, a zajętością pamięci. Sposób postępowania jest zazwyczaj następujący: pozyskuje się obraz o stosunkowo dużej ilości kolorów, by w trakcie kolejnych, wstępnych etapów obróbki obrazu dojść do mniejszej, praktycznej ilości kolorów. Oto lista najczęściej wykorzystywanych formatów:

- **binarny** (1 bpp). Ten najprostszy format, zajmujący najmniej pamięci, ma podstawowe znaczenie w analizie obrazu. Dużo algorytmów przetwarzania obrazu bazuje na tym formacie.
- **monochromatyczny** (8 bpp). Nazwa tego formatu wynika stąd, że najczęściej wartość danego elementu obrazu wyraża jego względową jasność. Za pomocą tego formatu można zakodować 256 stopni szarości.
- **kolorowy** (24 lub 32<sup>8</sup> bpp). Najczęściej po 8 kolejnych bitów w tym formacie opisuje nasycenie jednej z trzech barw podstawowych RGB<sup>9</sup>. Za pomocą tego formatu można zapisać około 17 milionów różnych odcienей kolorów. Jest to wartość w większości przypadków zupełnie wystarczająca. Wadą tego formatu jest stosunkowo duża zajętość pamięci.

<sup>6</sup> W praktycznych rozwiązaniach najczęściej stosuje się wymiary obrazu będące potegami liczby 2, aby w maksymalnym stopniu wykorzystać architekturę sprzętu komputerowego i ułatwić przeprowadzenie szybkiej transformacji Fouriera (FFT).

<sup>7</sup> ang. *bits per pixel*

<sup>8</sup> Format 32 bpp zawiera zazwyczaj tyle samo informacji, co tryb 24 bpp. Dodatkowe 8 bitów dodawane jest celem zwiększenia wydajności operowania na obrazie przez najbardziej rozpowszechnione systemy komputerowe o szynie danych szerokości 32, 64 lub 128 bitów.

<sup>9</sup> ang. RGB = red - green - blue. W systemach poligraficznych stosuje się najczęściej inny, 4-kolorowy model CMYK.

Oprócz wyżej wymienionych stosuje się jeszcze wiele innych, pośrednich формatów. Specjalizowane systemy przetwarzania obrazu wykorzystują zazwyczaj jedną, ustaloną ilość kolorów. Większość systemów uniwersalnych umożliwia jednak analizę obrazów o dowolnej ilości kolorów.

Najczęściej przedmiotem analizy obrazu jest obraz płaski, dwuwymiarowy. Ociągający nas świat jest jednak przestrzenny, trójwymiarowy. Praktyka<sup>10</sup> wykazuje, że aby poprawnie zinterpretować trójwymiarowy obraz i określić przestrzenne relacje widzianych przedmiotów wystarczą 2 obrazy dwuwymiarowe powstałe przez rzutowanie trójwymiarowej przestrzeni na dwa, przesunięte względem siebie punkty. W praktycznych rozwiązańach stosuje się 2 lub więcej takich obrazów.

Z przyczyn bardzo ograniczonych możliwości współczesnych systemów komputerowych głównie analizuje się obrazy statyczne. Ideałem byłoby jednak analizowanie obrazu dynamicznego, takiego jakim jest w rzeczywistości obraz video. Takie analizowanie sekwencji obrazów pozwala między innymi na korelowanie danych uzyskanych z wcześniejszych i późniejszych obrazów, co w wielu zastosowaniach ma spore znaczenie (na przykład w robotyce). Jednak aby uzyskać w miarę płynny obraz ruchu należałoby analizować około 25 obrazów (klatek) w ciągu sekundy. Wymaga to jednak 25-krotnego zwiększenia prędkości przetwarzania informacji, wielokrotnego zwiększenia zapotrzebowania na pamięć oraz znacznej komplikacji algorytmu. Z tego względu komputerowa analiza obrazów ruchomych jest jeszcze rzadko spotykana w praktyce i dlatego w tej książce zagadnienie to będzie pominięte.

### 2.1.2 Źródła sygnałów zamienianych na obrazy

Akwizycja obrazu jest procesem przetwarzania informacji o otaczającym świecie lub nie-elektronicznej reprezentacji obrazu (fotografie, plany, mapy, dokumenty papierowe) na postać cyfrową, dogodną do obróbki. Do akwizycji obrazu mogą zostać wykorzystane różne urządzenia techniczne. Oto opis kilku z najczęściej wykorzystywanych w tym celu, z uwzględnieniem ich zalet i wad:

- kamera video CCD
  - możliwość uzyskiwania dynamicznego obrazu;
  - duża prędkość przetwarzania;
  - tania;
  - niewielkie wymiary i masa;
  - bardzo mała rozdzielcość;
- cyfrowy aparat fotograficzny
  - wysoka rozdzielcość;
  - bardzo wysoka cena;

---

<sup>10</sup> Ludzki wzrok

- skaner
  - bardzo wysoka rozdzielcość;
  - możliwość wprowadzania jedynie wcześniej przygotowanych obrazów;
  - bardzo długi czas przetwarzania;
  - wysoka cena;
  - znaczna masa i wymiary;
- manualne tworzenie obrazu przy pomocy digitizera i programu graficznego
  - całkowita kontrola nad zawartością i parametrami obrazu;
  - niska cena;
  - odmiennosć w stosunku do obrazów rzeczywistego świata;
  - wysoka pracochłonność.

W konkretnych przypadkach wyboru odpowiedniej metody akwizycji obrazu dokonuje się w zależności od założonych parametrów reprezentacji obrazu i przeznaczenia systemu. W dalszym ciągu w tej książce głównie rozważane będą obrazy wprowadzane za pomocą kamery video (najczęściej obecnie z przetwornikiem CCD).

### 2.1.3 Przechowywanie obrazów

Z wprowadzaniem obrazu do systemu cyfrowego wiąże się kwestia jego przechowywania. Metoda przechowywania cyfrowej reprezentacji obrazu zależy od przyjętych parametrów tej reprezentacji (objętości) i sposobu działania systemu przetwarzania obrazu. W trakcie samego procesu analizy reprezentacja obrazu przechowywana jest zazwyczaj w pamięci operacyjnej komputera dokonującego analizy. Stosowny rozmiar tejże pamięci zależy głównie od objętości samego obrazu. Biorąc pod uwagę, że często trzeba rozważyć osobno obraz źródłowy i osobno obraz wynikowy, a ponadto biorąc pod uwagę, że ta sama pamięć powinna mieścić także oprogramowanie dokonujące analizy (z własnymi obszarami roboczymi) oraz system operacyjny komputera, rozmiar jej powinien być co najmniej trzykrotnie większy od rozmiaru pojedynczego przetwarzanego obrazu. Jeżeli chodzi o przechowywanie reprezentacji obrazu pomiędzy etapami przetwarzania lub jego archiwację, to możliwości jest wiele. Niestety mając na uwadze ogromne zazwyczaj rozmiary reprezentacji obrazu, popularne dyskietki 1.4 MB nie nadają się w większości przypadków. Na szczęście w ostatnich czasach powstało wiele alternatywnych rozwiązań. Oto lista kilku najbardziej popularnych nośników z określeniem ich podstawowych zalet i wad:

- dyski twarde (0.5 ÷ 8 GB)
- bardzo krótki czas dostępu;
- stosunkowo duża pojemność;

- są standardowym wyposażeniem komputerów osobistych;
- niemożliwość przeniesienia;
- stosunkowo wysoka cena;
- dyski wymienne SyQuest (270 MB)
  - bardzo krótki czas dostępu;
  - stosunkowo niewielka pojemność;
- streamery DAT (1 ÷ 16 GB)
  - ogromna pojemność;
  - niewielkie wymiary i masa;
  - niska cena nośnika;
  - bardzo długi czas dostępu;
  - wysoka cena napędów;
- dyski optyczne CD-ROM (650 MB)
  - bardzo niska cena nośnika;
  - duża odporność na czynniki zewnętrzne (trwałość);
  - możliwość tylko jednokrotnego zapisu;
- dyski magneto-optyczne MO (256 ÷ 650 MB)
  - niska cena nośnika;
  - duża odporność na czynniki zewnętrzne (trwałość);
  - wysoka cena napędów.

Wybór właściwej metody przechowywania obrazu zależy od metody przetwarzania obrazu i rozmiaru jego reprezentacji. Przy przechowywaniu reprezentacji obrazu duże znaczenie ma zastosowanie algorytmów kompresji obrazu. Dzięki zastosowaniu tych algorytmów można redukować ilość potrzebnego miejsca od kilku do kilkudziesięciu razy przy znikomym wzroście czasu przetwarzania.

#### 2.1.4 Drukowanie obrazów

Przy przetwarzaniu obrazów przy pomocy systemów komputerowych, najczęściej pośrednie wyniki analizy i obrazy wyjściowe obserwowane są na ekranach monitorów stanowiących standardowe wyposażenie komputerów. Monitory takie posiadają zazwyczaj rozdzielcość od 640x480 do 1600x1200 i ilość kolorów większą niż zdolność rozróżniania ludzkiego oka. Przy wykorzystywaniu monitorów do śledzenia pośrednich wyników analizy obrazu bardzo istotnym jest wybór odpowiednich kart graficznych współpracujących z takimi monitorami. Karty takie powinny charakteryzować się następującymi cechami:

- odpowiednią ilością wbudowanej pamięci do wyświetlania obrazu o założonej rozdzielcości i ilości kolorów;
- samodzielnym wykonywaniem podstawowych operacji graficznych celem uzyskania właściwej prędkości i odciążenia głównego procesora systemu zajmującego się analizą obrazu;

- zapewnieniem odpowiednio wysokiej częstotliwości odświeżania ekranu, nie powodującej nadmiernego zmęczenia wzroku.

Do uzyskania trwałego zapisu obrazu można skorzystać z jednego z następujących urządzeń:

- popularne drukarki laserowe lub atramentowe
  - tanie;
  - szybkie przygotowanie wydruku;
  - są standardowym wyposażeniem komputerów osobistych;
  - w większości przypadków czarno-białe;
  - niska rozdzielcość;
  - ograniczona wielkość wydruku;
  - wysokie koszty eksploatacji;
- naświetlarki poligraficzne
  - bardzo wysoka rozdzielcość;
  - pełna paleta kolorów;
  - bardzo wysoka cena;
  - długie przygotowanie wydruku;
- printer CRT
  - bardzo wysoka rozdzielcość;
  - pełna paleta kolorów;
  - wysokie koszty eksploatacji;
  - bardzo wysoka cena;
  - ograniczona wielkość wydruku.

Wybór konkretnego rozwiązania zależy zawsze od celu tworzenia systemu... i od zasobności kieszony osoby tworzącej system.

## 2.2 Zasady tworzenia obrazu cyfrowego

### 2.2.1 Obraz jako funkcja

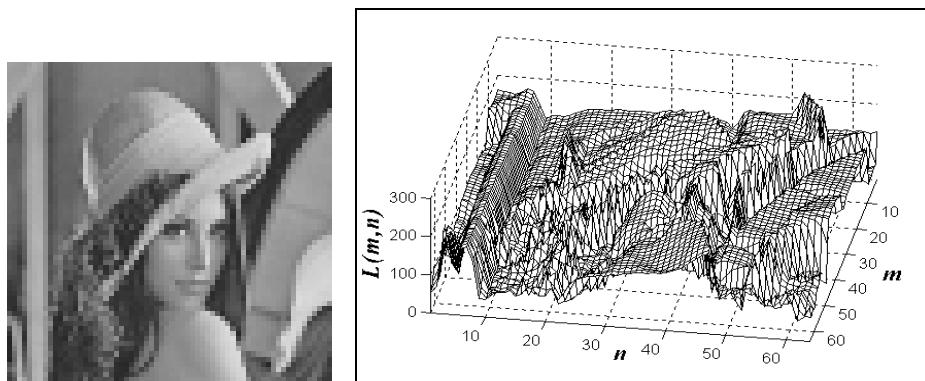
Stanowisko do automatycznego przetwarzania i analizy obrazu musi posiadać przede wszystkim system służący do wprowadzania obrazu i jego przetwarzania na postać cyfrową. Obraz cyfrowy powstaje w wyniku dyskretyzacji obrazu analogowego, dostarczonego przez odpowiedni przetwornik optoelektroniczny (kamerę TV). System ten zaczynać się musi kamerą TV<sup>11</sup> wraz z modułem zewnętrznego i automatycznego sterowania ręcznego systemami *iris* (jasność obrazu), *focus* (ostrość obrazu) oraz *zoom* (rozmiar obiektów na obrazie). Niekiedy w systemach wizyjnych specjalnego przeznaczenia (na przykład śledzących ruch

---

<sup>11</sup> Na przykład w rozwiązaniach stosowanych w Katedrze Automatyki AGH stosuje się kamery firmy Bischke typu CCD-SFI-5612P.

pojazdów na skrzyżowaniu) kamera wyposażona jest dodatkowo w system pozwalający na swobodne jej obracanie w dwóch płaszczyznach, co pozwala na automatyczne lub ręczne wyszukiwanie analizowanych obiektów i swobodne wybieranie pola widzenia.

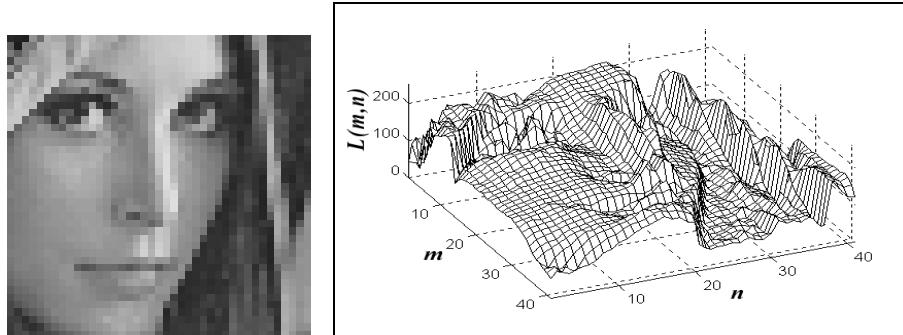
Obraz pozyskany przez kamerę jest analogowy, trzeba go więc przekształcić do postaci cyfrowej. Jeśli centralną jednostką przetwarzającą obraz jest mikrokomputer typu PC (najczęściej kompatybilny z IBM, na przykład z procesorem Pentium 133MHz), wówczas podczas wprowadzania i digitalizacji obrazu chętnie posłużymy się gotową kartą cyfrowej obróbki obrazu *frame-grabber* (na przykład w rozwiązaniach stosowanych w Katedrze Automatyki AGH stosuje się karty *frame-grabber* firmy Data Translation typu DT 2862). Oczywiście podsystem akwizycji obrazu można także zbudować we własnym zakresie<sup>12</sup> i wówczas trzeba nieco dokładniej poznać jego elementy strukturalne, jednak z możliwości tej korzysta się obecnie raczej wyjątkowo. Obrazy przetwarzane przez kamerę stanowią rzuty przestrzeni trójwymiarowej na płaszczyznę światłoczułą przetwornika optoelektronicznego. W tym rozdziale ograniczać się będziemy do przypadku, w którym mamy do czynienia jedynie z obrazem monochromatycznym. Obrazy kolorowe mogą być bowiem uważane za złożenie trzech obrazów w kolorach podstawowych: czerwonym, niebieskim i zielonym. W technice wizyjnej analogowy obraz rzutowany na płaszczyznę światłoczułą przetwornika optoelektronicznego jest reprezentowany przez dwuwymiarową funkcję  $L(x,y)$ , której argumenty  $x$  i  $y$  opisują powierzchniowe współrzędne punktu obrazu, zaś wartość funkcji określona jest przez poziom jasności obrazu (**luminancję** - w odróżnieniu od **chrominancji**, oddającej walor barwny). Dla podkreślenia monochromatyczności obrazu używać także będziemy niekiedy określenia „stopień szarości” (ang. *gray level*) dla określenia wartości funkcji  $L(x,y)$  w danym punkcie.



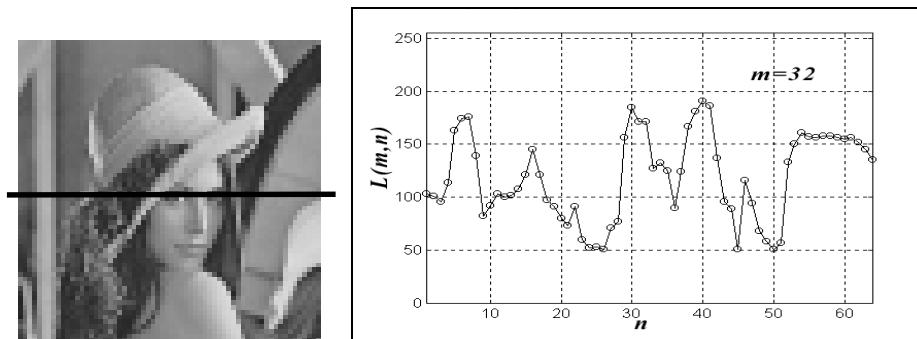
Rys. 2.3. Obraz „Lena” (64 x 64) i jego wykres, gdy obraz jest traktowany jako funkcja.

<sup>12</sup> O takich własnoręcznie budowanych systemach przetwarzania obrazów była mowa w książce R. Tadeusiewicza: *Systemy wizyjne robotów przemysłowych*, WNT, Warszawa 1993.

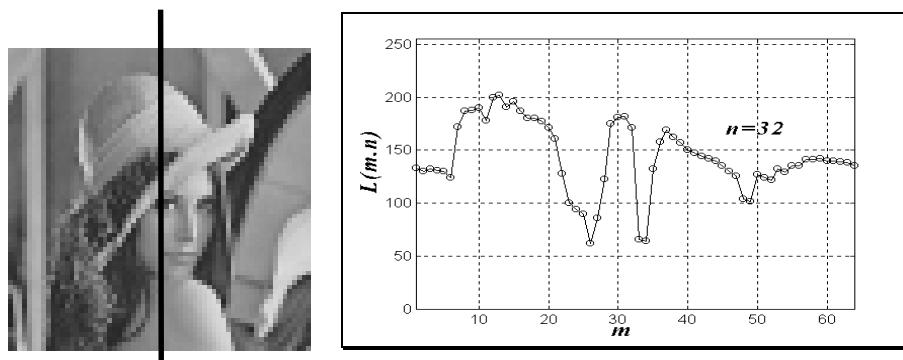
Aby przedstawić obraz rzeczywisty w postaci skończonej liczby wartości funkcji jasności, należy poddać go procesowi dyskretyzacji, składającemu się z procesów próbkowania i kwantowania. W ten sposób funkcja  $L(x,y)$  o argumentach zmieniających się w sposób ciągły zostaje zamieniona na macierz  $L(m,n)$  o  $M$  wierszach i  $N$  kolumnach, której elementy zawierają **skwantowane** poziomy jasności.



Rys. 2.4. Fragment obrazu „Lena” (rozmiar 41x41) i jego wykres jako funkcji.



Rys. 2.5. Poziomy przekrój obrazu „Lena” jako funkcja jednej zmiennej  $L(m^*,n)$  dla ustalonego  $m^* = 32$ .



Rys. 2.6. Pionowy przekrój obrazu „Lena” jako funkcja jednej zmiennej  $L(m,n^*)$  dla ustalonego  $n^* = 32$ .

Z punktu widzenia dalszych rozważań będzie bardzo ważne, by pamiętać i uwzględniać, że obraz  $L(m,n)$  może być rozważany i traktowany jako funkcja dwóch zmiennych - tak, jak to pokazano na rysunkach 2.3 i 2.4.

Na wykresie pokazanym na rysunku 2.3. można stosunkowo łatwo odnaleźć niektóre lepiej zauważalne struktury rozważanego obrazu - na przykład jasny pionowy słupek w lewej stronie zdjęcia poza głową dziewczyny, ciemny łukowy akcent w prawej części zdjęcia oraz przebiegające diagonalnie przez obraz rondo kapelusza. Trudniej natomiast doszukać się w wykresie funkcji innych elementów obrazu, takich jak na przykład twarz dziewczyny. Nawet powiększenie fragmentu obrazu (jak na rysunku 2.4) nie pozwala na swobodną interpretację w tym obszarze - jedyną rzucającą się w oczy strukturą są na tym obrazie „przepaściste otchłanie” oczu pięknej Leny i głęboka rozpadlina cienia oddzielającego kosmyk jej włosów.

Skoro obraz  $L(m,n)$  traktować można jako określona funkcję dwóch zmiennych, to również przekroje obrazu (to znaczy zbiory wartości pikseli obrazu  $L(m^*,n)$  dla ustalonego  $m = m^*$  i zmiennego  $n$ , względnie  $L(m,n^*)$  dla ustalonego  $n = n^*$  i zmiennego  $m$ ) można traktować odpowiednio jako funkcje jednej zmiennej (rys. 2.5 i 2.6).

## 2.2.2 Skutki dyskretyzacji obrazu

W związku z przetwarzaniem obrazu do postaci cyfrowej, a dokładniej w związku z jego próbkowaniem przestrzennym i dyskretyzacją przetwarzanych wartości - pojawiają się dwa istotne pojęcia: **rozdzielcość przestrzenna** (gęstość punktów próbkowania) oraz **rozdzielcość poziomów jasności** (gęstość reprezentacji stopni szarości). Im większa jest rozdzielcość przestrzenna obrazu, tym więcej szczegółów funkcji  $L(x,y)$  może być odnalezione i wykorzystane na obrazie cyfrowym  $L(m,n)$ . Ilustruje to rysunek 2.7, na którym pokazano ten sam obraz cyfrowy przedstawiony przy użyciu coraz mniejszych rozdzielcości (zakładano przy tym, co się zwykle praktykuje, że  $M = N$ ).

Na ilość informacji zawartych w obrazie cyfrowym wpływa także **rozdzielcość poziomów jasności**. Przyjęcie określonej rozdzielcości poziomów jasności oznacza tylko tyle, że o ile wartości funkcji luminancji obrazu analogowego spełniają warunek

$$L(x,y) \in R$$

gdzie  $R$  oznacza zbiór liczb rzeczywistych, o tyle wartości funkcji luminancji obrazu cyfrowego spełniają warunek

$$L(m,n) \in N$$

gdzie  $N$  oznacza zbiór liczb całkowitych z przedziału  $[0, 2^B - 1]$ , a  $B$  jest przyjęta liczbą bitów dla reprezentacji jednego punktu obrazu.



Rys. 2.7. Obraz cyfrowy „Lena” przedstawiany z coraz mniejszą rozdzielczością przestrzenną (kolejno pokazano obrazy dla  $M = N = 256 \times 256$ ,  $128 \times 128$ ,  $64 \times 64$ ,  $32 \times 32$ ,  $16 \times 16$  i  $8 \times 8$  pikseli).

Dla typowych systemów wizyjnych stosowanych w automatyce górną granicą dokładności reprezentacji punktów obrazowych jest 256 poziomów szarości na jeden punkt (8 bitów/piksel). Rozdzielczości znane z aplikacji pracujących w trybie “off-line” (np. 12 czy nawet 14 bitów/piksel w fotometrii) nie stosuje się ze względu na wysokie wymogi dotyczące mocy obliczeniowych systemu, a także dlatego, że w większości aplikacji pracujących w czasie rzeczywistym nie ma takich potrzeb i zadowolić się można nawet mniejszymi rozdzielczościami.

Na rysunku 2.8. pokazano, jak wygląda ten sam obraz przy coraz mniejszej liczbie ustalonych wartości funkcji luminancji (malejące  $B$ ). Widać, że w tym przypadku także dochodzi do pewnej utraty elementów merytorycznej zawartości obrazu, jednak utrata ta nie jest tak radykalna, jak przy zmniejszaniu rozdzielczości przestrzennej. Dlatego możliwe jest nawet przejście do granicy ( $B = 1$ ), a obraz pozostaje zrozumiały i czytelny.

Rozdzielczości poziomów jasności na wprowadzonym obrazie można swobodnie zmniejszyć, wykorzystując formułę:

$$L'(m, n) = \left\lceil \text{round} \left( \frac{L(m, n) - \left( \frac{\delta}{2} - 1 \right)}{\delta} \right) \right\rceil \cdot \delta + \left( \frac{\delta}{2} - 1 \right), \quad \text{gdzie } \delta = \frac{2^B}{2^{B'}}$$



Rys. 2.8. Obraz cyfrowy „Lena” przedstawiany z coraz mniejszą rozdzielczością poziomów jasności (kolejno dla  $B = 8, 5, 4, 3, 2, 1$ ).

W przytoczonym wzorze wartości opatrzone symbolem „prim” ( $L'$ ,  $B'$ ) oznaczają odpowiednio nową (przeskalowaną) wartość jasności piksela i liczby bitów, zaś analogiczne wartości bez „prim” oznaczają wartości po przeskalowaniu. Sensusowe skalowanie otrzymuje się wyłącznie przy założeniu, że  $B' < B$ , natomiast nie ma sposobu na to, by po zmniejszeniu liczby poziomów jasności z powrotem zwiększyć ją i uzyskać obraz o większej rozdzielczości skali szarości (można tylko rozszerzyć skalę). Jest to po prostu niewykonalne. Odnośnie do rozdzielczości przestrzennej warto wspomnieć, że po wprowadzeniu obrazu cyfrowego do komputera możliwe jest jej obliczeniowe modyfikowanie, zwane często skalowaniem obrazu. Modyfikacja taka wprowadza jednak zawsze pewne zniekształcenia do obrazu, co warto dokładniej obejrzeć, by mieć świadomość, jak ważną i odpowiedzialną sprawą jest dokonanie odpowiedniego wyboru rozdzielczości od razu na etapie konfigurowania sprzętu służącego do wprowadzania obrazów cyfrowych do systemu, w którym będą one potem przetwarzane, analizowane i rozpoznawane. Sprawdziwszy efekty skalowania na prostym przykładowym obrazie będziemy świadomi i uwrażliwieni na to, co nam grozi gdy dokonujemy wspomnianych skalowań zbyt bezkrytyczne.

Na rysunku 2.9 pokazano fragment obrazu „Lena” o rozmiarach 94x94 piksele.



Rys. 2.9. Fragment obrazu „Lena” o rozmiarach 94x94 piksele.

Po przeskalowaniu tego obrazka do (pozornie) większej rozdzielczości 128x128 otrzymujemy obraz o większej liczbie pikseli, ale o tej samej ilości informacji, co powoduje, że obraz ten jest zniekształcony. Zniekształcenia te są bardziej lub mniej widoczne, w zależności od tego, jaką technikę zastosuje się do określania wartości pikseli uzupełniających obraz do wymaganego rozmiaru. Na rysunku 2.10 pokazano obraz powiększony metodą uzupełniania na zasadzie „najbliższego sąsiada”. Wadą tej metody jest łatwo zauważalna „ziarnistość” wynikowego obrazu.



Rys. 2.10. Obraz z rysunku 2.9. powiększony do dwa razy większej rozdzielczości (188 x 188 pikseli) metodą „najbliższego sąsiada”.

Alternatywna możliwość polega na zastosowaniu przy dodawaniu pikseli dla reprodukcji obrazu w powiększonych rozmiarach określonego typu interpolacji.

W najprostszym przypadku może to być interpolacja dwuliniowa, tak, jak to pokazano na rysunku 2.11. Obraz w takim przypadku optycznie wydaje się wierniejszy i dokładniejszy, jednak w rzeczywistości każda interpolacja szkodzi obrazowi znacznie dotkliwiej, niż „gruboskórną” technika powielania wartości pewnych pikseli na zasadzie najbliższego sąsiada.

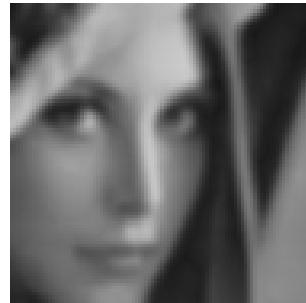


2.11. Obraz z rysunku 2.9. powiększony do dwa razy większej rozdzielczości (188 x 188 pikseli) metodą interpolacji dwuliniowej.

Można się o tym przekonać, gdy spróbuje się z obrazu o podwojonej rozdzielczości odtworzyć obraz pierwotny. Obraz poszerzony metodą najbliższego sąsiada da się odtworzyć w zmniejszonej skali praktycznie bez zmian, natomiast obraz interpolowany da obraz pierwotny zniekształcony i nieco zamazany - co pokazano na rysunku 2.12.



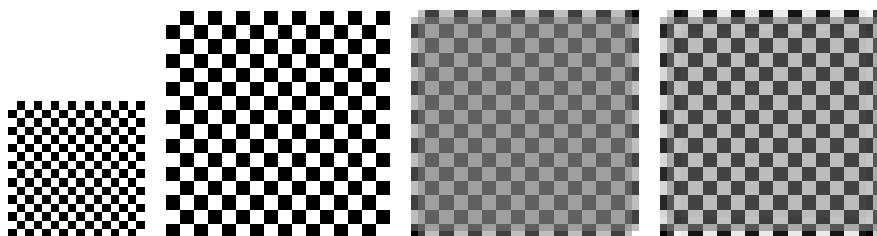
2.12. Obraz z rysunku 2.11. odtworzony z powrotem do dwa razy mniejszej rozdzielczości (94x94 pikseli).



2.13. Obraz z rysunku 2.9. dziesięciokrotnie powiększony do dwa razy większej rozdzielczości i dziesięciokrotnie z powrotem pomniejszony.

Porównanie obrazów z rysunku 2.9 i 2.12 może wnikliwemu i bystremu Czytelnikowi unaocznić zniekształcenia, jakie powstają przy stosowaniu skalowania interpolacyjnego, jednak ze względu na to, że bardzo subtelne zniekształcenia obrazu mogą być trudne do wykrycia - na rysunku 2.13. pokazano efekt **dziecięciokrotnego** przeskalowania obrazu o rozmiarach 94 x 94 piksele na 188 x 188 - i na odwrót. Teraz już zniekształcenia obrazu dają się naprawdę zauważyc gołym okiem!

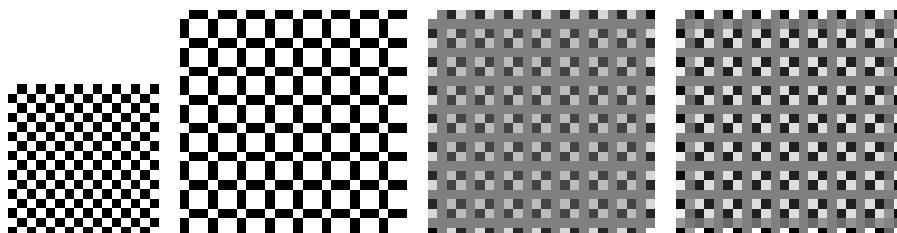
Zniekształcenia obrazu są jeszcze bardziej dotkliwe, kiedy rozmiary obrazu przed i po przeskalowaniu nie są w stosunku do siebie prostymi wielokrotnościami (jak to miało miejsce w przypadku rozważanych na rysunkach 2.9 ÷ 2.13 obrazach „Lena”). Zniekształcenia, jakie przy tym powstają, najłatwiej będzie prześledzić na przykładzie sztucznego obrazka, tak dobranego, by powstające zniekształcenia były szczególnie dotkliwie zauważalne. Rozważmy w związku z tym rysunek 2.14. Na rysunku tym widoczny jest sztuczny obraz „szachownicy” o rozmiarach 16 x 16 pikseli oraz wynik skalowania tego obrazu do rozmiaru 32 x 32 piksele za pomocą trzech metod: najbliższego sąsiada, interpolacji dwuliniowej i interpolacji dwukubicznej.



Rys. 2.14. Wynik skalowania sztucznego obrazu o rozdzielczości 16 x 16 do rozdzielczości 32 x 32.

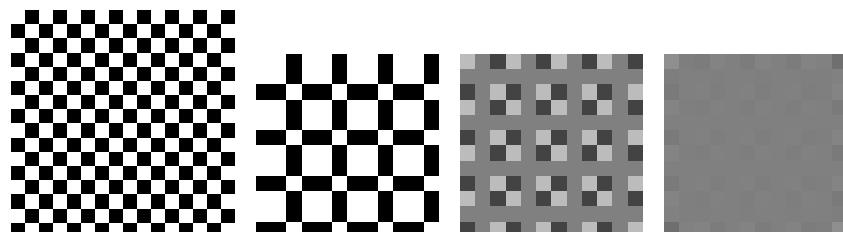
Zgodnie z przewidywaniem zniekształcenia geometryczne na rozważanym obrazie nie wystąpiły, chociaż na skutek interpolacji pogorszył się wyraźnie kontrast

między czarnymi i białymi polami. Rozważmy teraz przypadek przeskalowania tego samego obrazu o rozmiarach  $16 \times 16$  pikseli do rozdzielczości  $24 \times 24$  piksele. Wynik pokazano na rysunku 2.15.



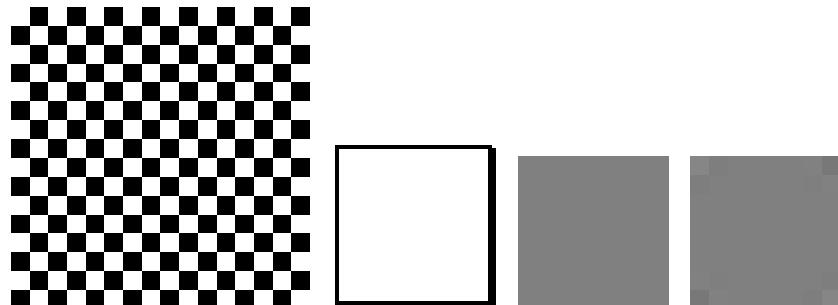
Rys. 2.15. Wynik skalowania sztucznego obrazu o rozdzielczości  $16 \times 16$  do rozdzielczości  $24 \times 24$ .

Znieksztalcenia obrazu są teraz bardzo duże, co szczególnie szokująco wygląda na obrazie przeskalowanym metodą najbliższego sąsiada. Z podanych przykładów mógł wynikać wniosek, że znieksztalcenia pojawiają się wyłącznie przy próbach powiększania obrazu. Niestety, znieksztalcenia wynikają także w przypadku braku współmierności rozmiarów obrazu przed i po skalowaniu i dają o sobie znać również przy zmniejszaniu obrazu.



Rys. 2.16. Wynik skalowania sztucznego obrazu o rozdzielczości  $16 \times 16$  do rozdzielczości  $12 \times 12$  za pomocą trzech metod: najbliższego sąsiada, interpolacji dwuliniowej i interpolacji dwukubicznej.

Na rysunku 2.16 pokazano (w większej nieco skali, dla poprawnego uwidocznienia szczegółów) skutki przeskalowania sztucznego obrazu o rozmiarach  $16 \times 16$  pikseli do mniejszego rozmiaru  $12 \times 12$  pikseli (znowu, w kolejności od lewej do prawej pokazano skutki skalowania za pomocą trzech metod: najbliższego sąsiada, interpolacji dwuliniowej i interpolacji dwukubicznej). Efekt jest z pewnością daleki od oczekiwania. Nie jest prawdziwe także przypuszczenie, że współmierność, a dokładniej - prosta wielokrotność rozmiarów obrazu przed i po przeskalowaniu stanowi uniwersalną receptę na wszystkie kłopoty ze skalowaniem. Rozważmy bowiem na rysunku 2.17 efekty skalowania testowego obrazu o rozdzielczości  $16 \times 16$  pikseli do rozmiaru  $8 \times 8$  pikseli.



Rys. 2.17. Wynik skalowania sztucznego obrazu o rozdzielczości  $16 \times 16$  do rozdzielczości  $8 \times 8$  za pomocą trzech metod: najbliższego sąsiada, interpolacji dwuliniowej i interpolacji dwukubicznej.



Rys. 2.18. Obraz „Lena” kolejno: oryginał ( $512 \times 512$ ), przeskalowany do  $341 \times 341$ , przeskalowany do  $154 \times 154$ , odtworzony w pierwotnym rozmiarze ( $512 \times 512$ ).

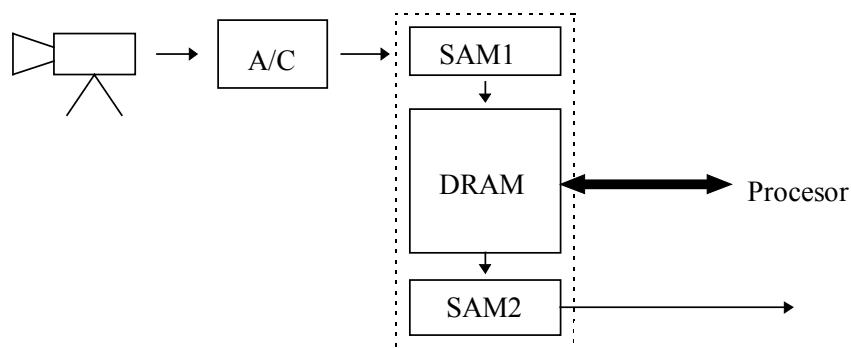
Próba skalowania znowu zakończyła się niepowodzeniem - z tego powodu, że zmniejszona rozdzielcość nie pozwoliła na odtworzenie tekstury oryginalnego obrazu.

Opisane wyżej efekty pojawiają się także podczas skalowania rzeczywistych obrazów, ale są tam znacznie mniej dokuczliwe, ponieważ zdolność naszego wzroku do podświadomego korygowania niedoskonałości oglądanych obrazów powoduje, że nawet trochę zniekształcone (na przykład procesem skalowania) obrazy - postrzegamy jako zupełnie zadowalające. Patrzmy na przykład na rysunek 2.18. Są tam cztery obrazy: oryginalny obraz „Lena” w rozdzielcości 512x512 pikseli, ten sam obraz przeskalowany do rozmiaru 341x341, ten sam obraz przeskalowany do rozmiaru 154x154 i obraz zrekonstruowany z tego ostatniego obrazu przeskalowanego. W obydwu przypadkach skalowania dokładna analiza (oparta na przykład na próbie rekonstrukcji obrazu w oryginalnej wielkości i obliczeniu sumy kwadratów różnic pomiędzy wartościami tych samych pikseli przed i po skalowaniu) pozwala wykazać spore zniekształcenia obrazu - a tymczasem gołym okiem trudno cokolwiek zauważać (dla ułatwienia oceny i ewentualnych porównań wszystkie trzy obrazy zostały pokazane w tej samej wielkości na rysunku).

## 2.3 Urządzenie do wprowadzania obrazu

### 2.3.1 Ogólny schemat systemu wprowadzania obrazu

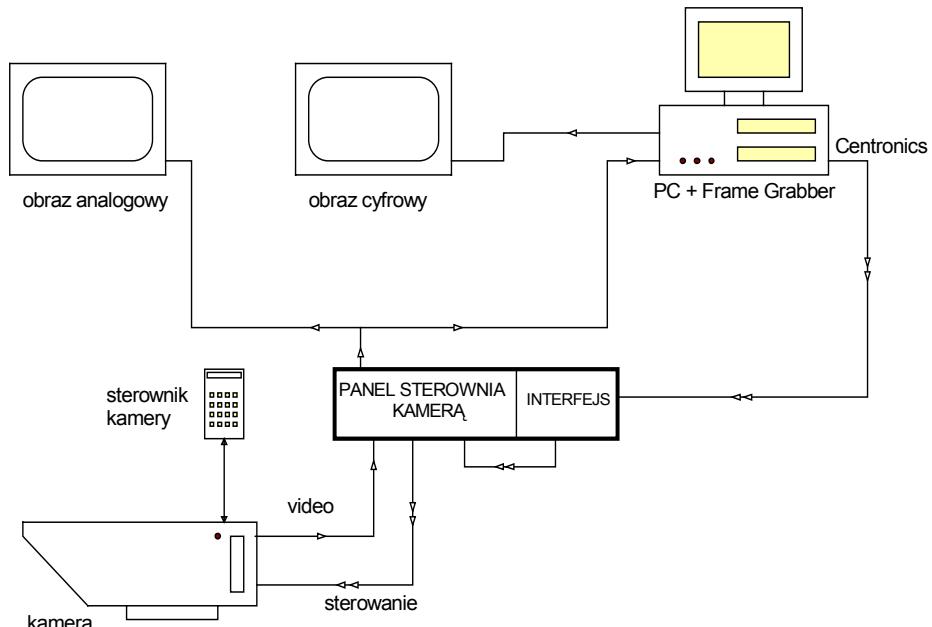
Schemat przykładowego układu służącego do pozyskiwania obrazów cyfrowych i do wprowadzania ich do systemu przetwarzającego (komputera) w największym uproszczeniu pokazuje rysunek 2.19.



Rys. 2.19. Przykładowa struktura stanowiska do wprowadzania obrazu do systemu komputerowego.

Jak widać z podanego na rysunku 2.19 schematu, sygnał z kamery po przetworzeniu w przetworniku analogowo-cyfrowym trafia do pamięci buforowej, z której jest w miarę potrzeby wydobywany i przetwarzany za pośrednictwem

odpowiedniego procesora. Dokładniejszy schemat wejściowego fragmentu systemu pokazuje z kolei rysunek 2.20. Na rysunku tym, obok obiektów już wyżej omówionych i przedstawionych, pokazane są elementy kontroli (monitory) i sterowania.

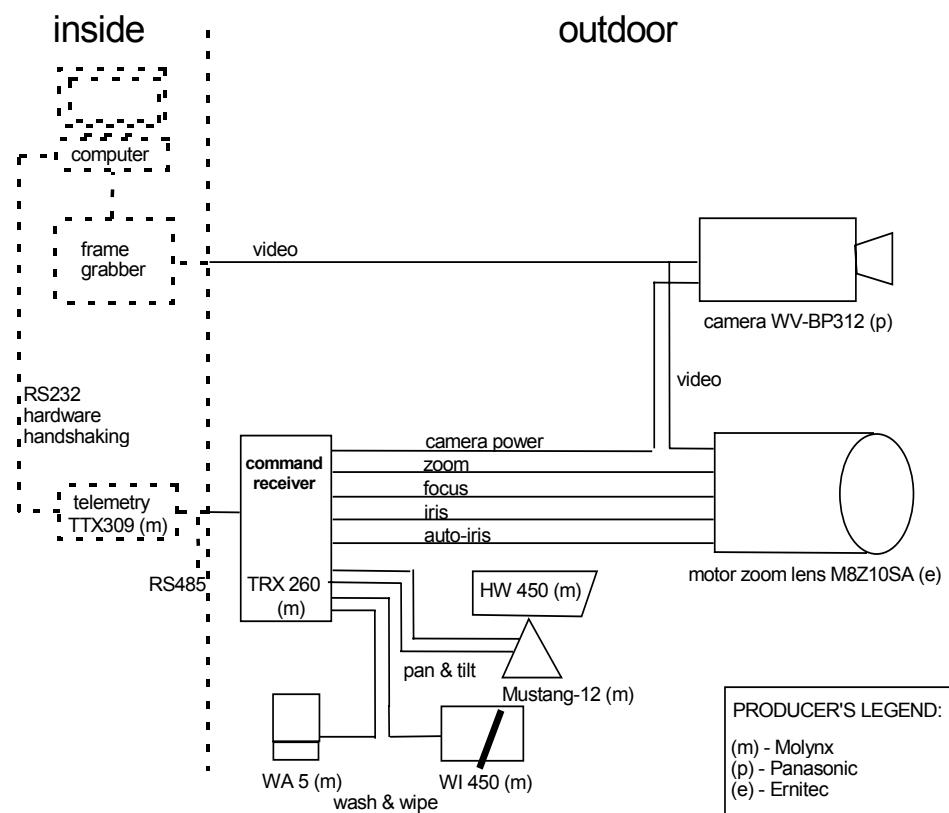


Rys. 2.20. Układ wprowadzania obrazu wraz z elementami kontrolnymi i sterującymi.

Obok sterowania optyką kamery (obiektywem w ramach funkcji *zoom*, *autofocus*, itd.) celowe jest także niekiedy sterowanie kamerą w sensie jej fizycznego przemieszczania, celem swobodnego wybierania kadru podlegającego procesowi analizy i przetwarzania. Stosowane w Katedrze Automatyki AGH rozwiązanie tego typu przedstawiono na rysunku 2.21.

W systemach analizujących obrazy w czasie rzeczywistym dla potrzeb kontroli lub sterowania praktycznie nie stosuje się rozwiązań współpracujących z kamerami kolorowymi. Jednak dla kompletu opisu omawianych tu zagadnień trzeba dodać kilka uwag także i na temat kamer barwnych. Klasycznym podejściem do analizy i przetwarzania obrazów barwnych jest wykorzystywanie modelu **RGB** wywodzącego się najpierw z fotografii barwnej, potem z barwnej telewizji, a potem z barwnej grafiki komputerowej. Genezy modelu RGB należy szukać w trójchromatycznej teorii widzenia, zgodnie z którą dowolną barwę można uzyskać w wyniku zmieszania trzech barw podstawowych. O tym, że barwami tymi są barwy R, G i B (czerwona (*red*), zielona (*green*) i błękitna (*blue*)), zadecydowały względy techniczne. Barwy te dość łatwo uzyskać na błonie fotograficznej a także na ekranie monitora, pokrytego odpowiednio rozmieszczeniami

plamkami luminoforów, które pobudzone trzema strumieniami elektronów świecą światłami o barwach: czerwonej, zielonej i niebieskiej. W modelu RGB gama dostępnych barw jest reprezentowana przez sześcią jednostkę w układzie współrzędnych kartezjańskich. Model RGB jest modelem **addytywnym**. Barwy różne od podstawowych uzyskuje się w wyniku przestrzennego sumowania strumieni światła o barwach podstawowych, jakimi świecą plamki luminoforów. Proporcje strumieni światła są określone przez wartości sygnałów sterujących podawanych na monitor, z reguły znormalizowanych do zakresu [0, 1]. Stosując jednakowe wysterowania uzyskuje się poziomy szarości, od czarnego do białego. Tak więc wszystkie poziomy szarości występują w modelu RGB na przekątnej łączącej wierzchołki (0,0,0) i (1,1,1).



Rys. 2.21. Struktura systemu pozyskiwania obrazów cyfrowych z ruchomą kamerą.

Model RGB koncepcyjnie jest bardzo prosty i ze względu na liniowość umożliwia dość łatwe wykonywanie obliczeń związanych z wyznaczeniem barw obiektów. Korzystając z modelu RGB nie można jednak zapomnieć o kilku problemach, które są istotne ze względów praktycznych.

- Sześciian RGB jest modelem dyskretnym. Poszczególne barwy są reprezentowane w sześciianie RGB za pomocą punktów rozmieszczonych wewnętrz i na powierzchni sześciianu.
- Zależnie od liczby bitów ‘m’ przeznaczonych do reprezentowania barwy piksela, liczba dostępnych barw jest ograniczona do  $2^m$ . Przykładowo - rozmieszczenie barw w sześciianie RGB przy 8 bitach na piksel (R - 3bity, G - 3bity, B - 2bity) ukazuje ograniczoną liczbę różnych barw, jaką można dysponować przy przejściu na przykład od barwy R do barwy B. Model RGB reprezentuje pewien podzbiór barw widzialnych.
- Możliwość odtwarzania barw jest ograniczona przez prawa opisujące addytywne mieszanie barw. Sześciian RGB nie jest przestrzenią jednorodną, jeśli chodzi o percepcję barw. Jednakowym zmianom wzdłuż dowolnej ścieżki w sześciianie RGB nie zawsze odpowiadają takie same zmiany w odczuciu człowieka.

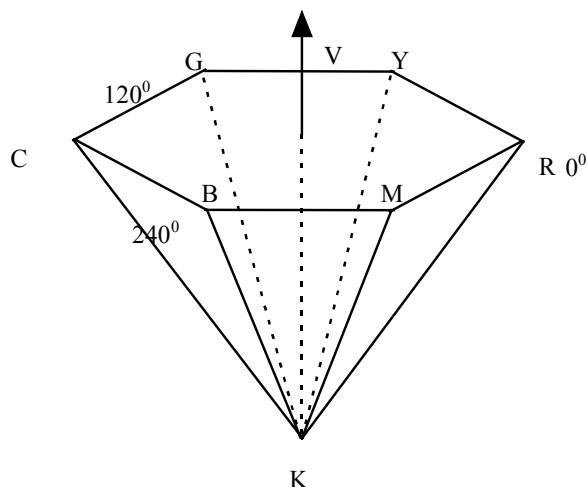
Z wymienionych powodów model barw RGB nie jest zbyt wygodny przy interaktywnej pracy z komputerem, a więc wtedy, kiedy użytkownik ma możliwość bezpośredniego określania potrzebnej barwy. Wpływ zmian wartości poszczególnych składowych R, G, B na końcową barwę nie zawsze jest oczywisty. Z tego względu dla potrzeb komputerowego przetwarzania obrazów zostały opracowane inne modele barw, których podstawową zaletą miało być ułatwienie odbioru odpowiedniej barwy. Jednym z tych modeli jest model **HSV**<sup>13</sup>. W tym modelu określa się: odcień barwy (*Hue*), nasycenie (*Saturation*) i wartość (*Value*).

Bryłą, w której są zawarte wszystkie barwy reprezentowane w modelu HSV, jest ostrosłup sześciokątny (rys. 2.22), który został skonstruowany w następujący sposób. Jeżeli popatrzy się na sześciian RGB z zewnątrz wzdłuż przekątnej, na której leżą poziomy szarości, to obserwowany kontur bryły będzie sześciokątem foremnym, w którego wierzchołkach będą się znajdować barwy R, Y, G, C, B, M. W środku sześciokąta będzie znajdować się barwa biała. Wnętrze będzie wypełnione odcieniami barw o różnym nasyceniu. Konstrukcja bryły polega na utworzeniu osi prostopadłej do sześciokąta i przechodzącej przez jego środek. Na osi tej są reprezentowane różne poziomy szarości. Po ustaleniu na tej osi punktu, któremu zostaje przypisana barwa czarna, można już skonstruować odpowiedni ostrosłup foremny. Osi głównej są przypisane wartości od 0 dla wierzchołka do 1 w środku podstawy. Każdy przekrój ostrosłupa prostopadły do osi głównej zawiera wszystkie odcienie barw o różnych nasyceniach i o jasności odpowiadającej na osi. Na obwodzie sześciokąta znajdują się barwy nasycone. Każdemu odcieniowi barwy jest przypisany kąt liczony przeciwnie do ruchu wskazówek zegara, poczynając od półprostej łączącej środek sześciokąta

---

<sup>13</sup> Istnieją pewne **odmiany modelu HSV** - model HSB, w którym B oznacza jasność (*Brightness*), i model HSI, w którym I oznacza intensywność (*Intensity*).

z wierzchołkiem reprezentującym barwę czerwoną. Wzdłuż każdej półprostej łączącej środek sześciokąta z barwą nasyconą leżą barwy nienasycone o tym samym odcieniu co barwa nasycona. Im bliżej środka sześciokąta, tym mniejsze nasycenie barwy. Skala nasycenia barwy zmienia się od 0 do 100%.



Rys. 2.22. Model HSV.

Korzystanie z modelu HSV umożliwia wybieranie barw w sposób zbliżony do stosowanego przy malowaniu. Można najpierw określić potrzebny odcień barwy, następnie, dodając białej barwy - zmniejszyć jej nasycenie i wreszcie, dodając czarnej barwy - dobrać jej jasność. Oczywiście kolejność postępowania może być też odwrotna i możliwe jest iteracyjne dochodzenie do ostatecznej barwy.

Model HSV stanowi pewne przybliżenie rzeczywistości, nie zawsze zgodne z odczuciami psychofizycznymi. Model ten skonstruowano przy założeniu, że wszystkie odcienie barw są nasycone przy tej samej wartości jasności (dla  $V=1$ ). Model HSV stanowi przestrzenne przedstawienie barwnych światel z wnętrza sześcianu RGB. Organizacja modelu bryły barw jest systematyczna, ale nie ma cech przestrzeni liniowej. O ile model RGB ma bezpośrednie odzwierciedlenie w realizacji technicznej, o tyle model HSV - nie. Dlatego często konieczne jest przechodzenie z jednego z tych modeli na model RGB i odwrotnie (zwłaszcza w systemach interaktywnych, kiedy użytkownik korzysta z modelu HSV, a ogląda obraz wyświetlany na monitorze RGB). Należy dodać, że do przyspieszenia procesu translacji między modelami opracowano specjalizowane układy konwersji. Na przykład firma Data Translation opracowała układy scalone do konwersji  $RGB \leftrightarrow HSV^{14}$ .

---

<sup>14</sup> Curran L. „Chip set speeds color image processing” Electronic Design, January 12 1989, s. 147-149.

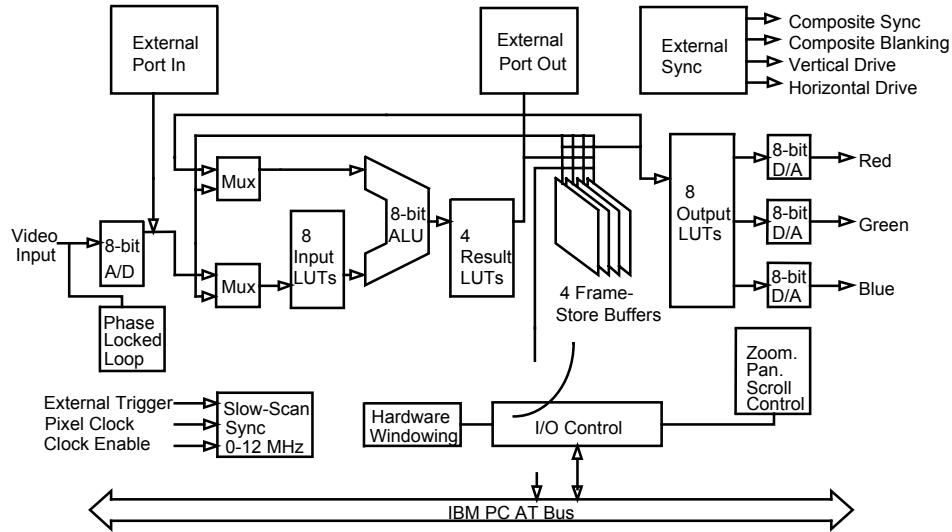
### 2.3.2 Przetwornik obrazu - *Frame Grabber*

Proces pozyskiwania obrazu cyfrowego rozpoczyna się od pobrania obrazu z kamery przez kartę *Frame Grabber*. Karta ta dokonuje omówionych w poprzednim rozdziale czynności próbkowania obrazu (co zamienia ciągły analogowy sygnał z kamery na matrycę dyskretnych pikseli i tworzy odpowiedni raster) i konwersji analogowo-cyfrowej wartości luminancji (stopnia szarości) poszczególnych punktów obrazu cyfrowej (co wprowadza kwantyzację skali jasności). Jednocześnie z pobraniem obrazu i jego zamianą na postać cyfrową następuje wizualizacja pobranego obrazu na monitorach: analogowym i cyfrowym. Obrazy te mogą być podstawą do regulacji i sterowania pracą całego systemu. Sterowanie to może przebiegać w sposób ręczny lub automatyczny.

Zazwyczaj systemy typu *Frame Grabber* wyposażone są dodatkowo w prosty program analizy ostrości i jasności wprowadzanego obrazu i na tej podstawie na wyjściu równoległym komputera (*Centronics*), do którego może być dołączony interfejs kamery, pojawiają się odpowiednie długości impulsy sterujące pierścieniami obiektywu. Jeśli kamera jest do tego odpowiednia przygotowana - następuje w tym momencie automatyczna korekta warunków rejestracji obrazu. Po wykonaniu przez silniki pierścieni obiektywu kamery wymaganego obrotu - następuje ponowne pozyskanie obrazu i jego kolejna ocena. Proces ten może być automatycznie kontynuowany aż do uzyskania obrazu cyfrowego, który automatyka kamery uzna za dostatecznie ostry i wystarczająco naświetlony.

Kluczową rolę w pozyskiwaniu obrazu odgrywa system próbkowania i analogowo-cyfrowego przetwarzania sygnału z kamery, czyli po prostu karta *Frame Grabber*. Karta taka często zawiera dodatkowo cały układ cyfrowego procesora obrazu, czego przykładem jest używany w Katedrze Automatyki *Frame Grabber DT 2862*. Jego schemat pokazano na rysunku 2.23.

Działanie systemu *Frame Grabber* (na przykładzie DT 2862) jest następujące. Do modułu wejściowego wprowadzany jest sygnał telewizyjny (w standardzie CCIR), stąd jego nazwa: moduł przyjęcia kadru. Moduł główny, zwany buforem kadru (ang. *frame buffer*), to pamięć zdolna przyjąć obraz wejściowy. Podstawową cechą wyróżniającą bufor kadru jest to, że zawartość pamięci może być zapisana bądź odczytana w czasie 1/25 sekundy. Umożliwia to przyjęcie obrazu niemal natychmiast, aby praca systemów przetwarzających obrazy odbywała się w czasie rzeczywistym, z opóźnieniem tylko jednego kadru. Dzięki takiemu rozwiązaniu pamięć obrazu może być bezpośrednio adresowana przez moduł wyświetlający, co pozwala między innymi wysyłać bezpośrednio zawartość pamięci na ekran monitora wyjściowego (przez przetwornik cyfrowo-analogowy). Inne metody adresowania pozwalają uzyskać efekt zoom (pomniejszanie lub powiększanie obrazu), oraz *scroll* (przesuwanie) i *pan* (tzw. panoramiczne – zmiana kąta widzenia).



Rys. 2.23. Schemat blokowy przykładowego *frame-grabbera* (DT 2862 firmy DATA TRANSLATION).

Jeśli w używanym systemie *Frame Grabber* występuje moduł przetwarzania obrazu (zwany ALU - *Arithmetical and Logical Unit*), to wykonuje on na macierzy jasności poszczególnych pikseli obrazu operacje niskiego poziomu (arytmetyczne i logiczne). Pozwala to włączyć pewne elementy procesu przetwarzania obrazu bezpośrednio do etapu jego pozyskiwania.

Zadaniem modułu wyjściowego karty *Frame Grabber* jest odczyt z pamięci przetworzonego obrazu, zamiana na sygnał analogowy i wyprowadzenie tego sygnału na monitor TV. Oczywiście z powyższym systemem współpracują odpowiednie pakiety programowe.

### 2.3.3 Kamera

W obecnie stosowanych komputerowych systemach wizyjnych budowa kamery jest zazwyczaj oparta o scalone analizatory obrazu, a w szczególności - o przetworniki CCD. Ich parametry mają zasadniczy wpływ na cały proces przetwarzania obrazu. Wstępny etap obróbki sygnału video we współczesnych systemach wizyjnych czasu rzeczywistego wymaga od układu interfejsu kamery rozwiązania gwarantującego osiągnięcie:

- wymaganej rozdzielczości,
- odpowiednio dokładnej reprezentacji punktów obrazowych,
- zachowania proporcji obrazu w buforze pamięci,
- dopasowania architektury do potrzeb dalszych etapów potokowej obróbki obrazu (z uwzględnieniem problemu międzyliniowości),
- spełnienia wymogów czasu rzeczywistego.

Jedną z podstawowych cech przetwornika CCD jest jego rozdzielczość. W systemach wizyjnych, jak już wyżej opisano, stosuje się szereg rozdzielczości: od najmniejszych rzędu 128x128 (czasami nawet 64x64), aż po 1024x1024, a nawet więcej<sup>15</sup> punktów w obrazie. Stosowana rozdzielczość obrazów cyfrowych związana jest także ze sposobem organizacji bufora pamięci i dlatego jest ona najczęściej potęgą liczby 2 (np. 256x256, 512x512, itp.), gdyż to upraszcza adresację. Często jednak wybór rozdzielczości kamery przez użytkownika związany jest z posiadaną przez niego platformą sprzętową (np. dla komputerów klasy PC stosowane są niekiedy rozdzielczości systemów wprowadzania obrazu zgodne z rozdzielczością stosowanych w tych komputerach typowych kart graficznych, np. 640x480, 800x600 lub tp.).

Profesjonalne systemy o dużych rozdzielczościach z reguły wykorzystują kamery CCD z wyjściem cyfrowym, często w dedykowanym standardzie, który nie stwarza większych problemów przy połączeniu z resztą systemu. Niestety koszt takich kamer (około 5000\$ dla kamer o 256 poziomach szarości i rozdzielczości 512x512) jest zbyt wysoki, aby mogły być stosowane powszechnie w praktycznych aplikacjach. Zastosowanie standardowych kamer video z wyjściem analogowym wymaga natomiast zastosowania przetwornika A/C, który zamieni jeden ze standardów analogowych (NTSC, PAL) w ciąg próbek cyfrowych obrazu (np. 8 bitów). Przyjęcie jednego ze standardów ogranicza osiąganą rozdzielczość systemu w pionie:

- w przypadku systemów europejskich (PAL) do 575 linii,
- w przypadku systemów amerykańskich i japońskich (NTSC) do 483 linii.

Jeżeli chodzi o rozdzielczość w poziomie, to w przypadku analizatorów CCD z tzw. przesuwem ramki (*FT- Frame Transfer*) osiągane są rozdzielczości rzędu 500-750, a w przypadku analizatorów z tzw. przesuwem międzykolumnowym (*IT- Interline Transfer*) zawierają się w granicach 350-450. Należy też dodać, że rozdzielczość pionowa w przypadku tych ostatnich (IT), jest również w granicach 350, a wymagana przez standard liczbę linii w pionie osiąga się przez przesyłanie tego samego obrazu jako dwie oddzielne ramki (*frame*).

Przy pracy w systemach analizujących obrazy ruchome jest jeszcze jeden problem. Czas akwizycji pola czy ramki jest zbyt długi, by obraz poruszającego się obiektu był wyraźny. Stosuje się więc układ migawek, które uaktywniają przetwornik CCD na znacznie krótszy czas, niż czas wybierania całego obrazu (choć osobnym problemem jest zapewnienie odpowiedniego oświetlenia kadru przy tak bardzo krótkich czasach). W części kamer video migawka otwiera się

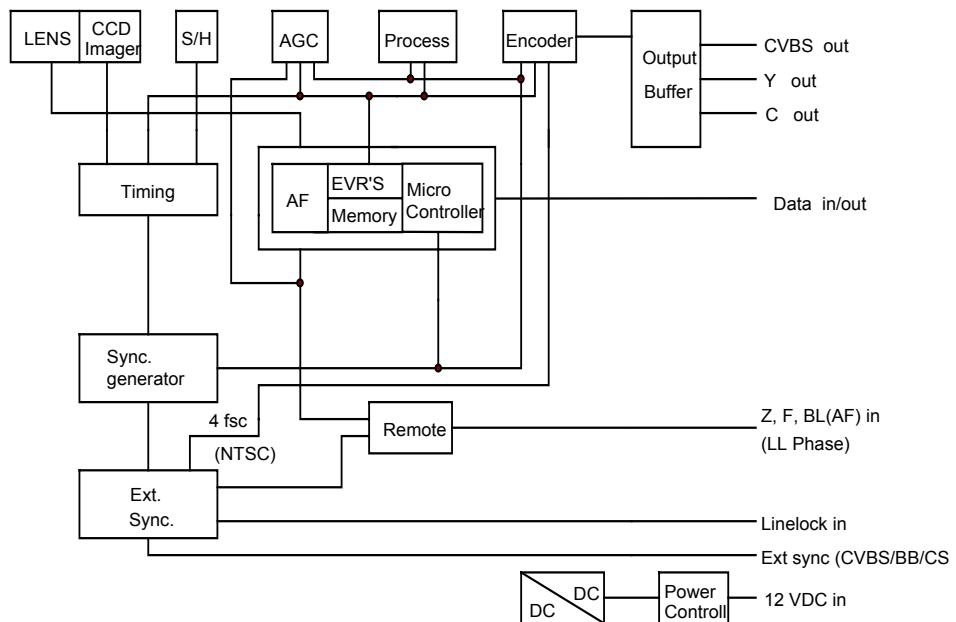
---

<sup>15</sup> Należy dodatkowo stwierdzić, że koszt i stopień złożoności systemu pracującego w czasie rzeczywistym o rozdzielczości powyżej 512 x 512 jest bardzo duży z tytułu wymaganych dużych mocy obliczeniowych i dlatego stosowane obecne rozwiązania systemów wizyjnych rzadko przekraczają rozdzielczość ograniczoną standardem PAL.

jednokrotnie na każde pole a nie obraz. Nie można zatem zastosować w systemie cyfrowym wygodnej techniki buforowania, ponieważ poszczególne pola obrazu nie zostały “uchwycone” w tej samej chwili. Prowadziłoby to do zniekształceń analizowanych krawędzi obrazu. Ten problem jest nie do rozwiązania i stosowanie tych kamer do analizy obiektów ruchomych praktycznie dwa razy obniża osiąganą rozdzielcość obrazu w pionie, zmuszając system do analizy pól, a nie ramek.

Sposób działania migawki jest więc kluczową sprawą w systemach analizujących ruch, ale należy również zwracać uwagę na typ używanego przetwornika CCD. Nic nie wnosi bowiem wyeliminowanie podwójnego wyzwalania migawki w jednej ramce, jeżeli przetwornik CCD wymaganą przez standard PAL czy NTSC międzyliniowość osiąga przez mechaniczne przemieszczanie matrycy CCD w naprzemiennych ramkach (patenty EP65885 THOMPSON-CSF, EP150973 TOSHIBA itp.). Trzeba dodać jednak, że to rozwiązanie stosowane jest dość rzadko, więc stosunkowo łatwo jest znaleźć kamerę wolną od wskazanej wady.

Należy też dodać, że większość dostępnych kamer analogowych w swej dokumentacji nie precyzuje ani typu użytego przetwornika, ani opisanych problemów związanych z migawką. Dobór więc odpowiedniej kamery dla systemów analizujących ruch nie jest więc sprawą trywialną ani prostą, gdyż wymagać może samodzielnych badań mających na celu identyfikację wchodzących w rachubę kamer.



Rys. 2.24. Schemat budowy typowej kamery video dla zastosowań w przetwarzaniu obrazów.

Użyte na rysunku oznaczenia mają następującą interpretację:

LENS - zespół soczewek, CCD Imager - przetwornik obrazowy typu CCD,

S/H - układ próbkujuco - pamiętajacy, (ang. *sample/hold*),

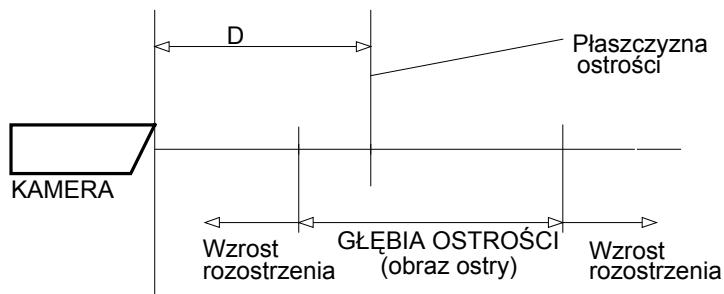
AGC - automatyczna regulacja wzmacnienia, (ang. *automatic gain control*).

AF - układ automatycznej regulacji ostrości (ang. *autofocus*),

EVR - (ang. *electronic variable resistors*), Remote - blok zdalnego sterowania.

Na rysunku 2.24 przedstawiono schemat blokowy używanej w Katedrze Automatyki AGH kamery **Bischke CCD SFI 5612p**. Wybór tej kamery dla systemów przetwarzania obrazów podyktowany został między innymi możliwością komputerowego sterowania silnikami poruszającymi pierścieniem regulującym długość ogniskowej i ostrość.

Bardzo istotnym składnikiem każdej kamery jest układ optyczny (w skrócie nazywany dalej obiektywem), którego zadaniem jest uzyskanie ostrego obrazu rozważanych obiektów. Zagadnieniem struktury obiektywu nie będziemy się tu zajmowali, warto jednak podkreślić, że ogromnie istotnym warunkiem uzyskania dobrej jakości elektronicznego przetwarzania obrazu jest wcześniejsze uzyskanie jego dobrej ostrości w systemie optycznym. Większość systemów soczewek odtwarza obraz ostry tylko w pewnym zakresie położień, określonym głębią ostrości, co przedstawia rys. 2.25.



2.25. Schemat obszaru ostrości obrazu dla typowej kamery wideo.

Jeżeli obiekt znajduje się poza obszarem ograniczonym powierzchniami ostrości, jest on rozmyty. Stopień rozstrojenia zależy od odległości od tych powierzchni. Wraz ze wzrostem tej odległości obiekty są coraz bardziej nieostre. Odległość D pomiędzy obiektem i kamerą jest możliwa do obliczenia z podanego niżej wzoru, jeżeli znane są parametry systemu optycznego i współczynnika rozproszenia:

$$D = \frac{Fv}{v - F - \sigma n}$$

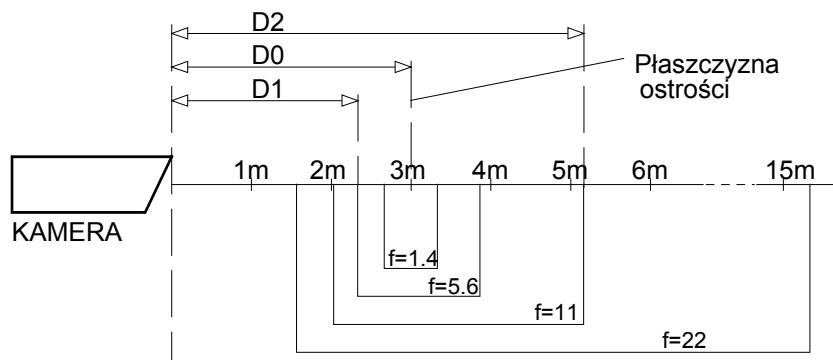
gdzie: v - odległość między kamerą, a obrazem

n - ilość soczewek w układzie optycznym

F - ogniskowa

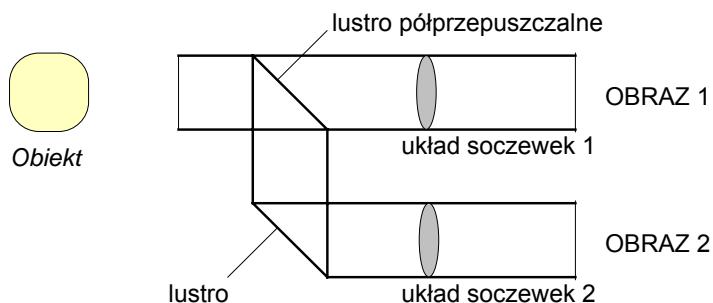
$\sigma$  - współczynnik rozproszenia, opisujący jak bardzo obraz jest rozstrojony

Przykładowe wartości liczbowe ustalające położenie i zakres głębi ostrości dla różnych wartości względnego otworu przysłony  $f$  dla typowego obiektywu telewizyjnego pokazano na rysunku 2.26.



Rys. 2.26. Przykładowy zakres głębi ostrości dla obiektywu o ogniskowej 50mm.

Do automatycznego ustawiania ostrości obiektywów kamer TV używa się zazwyczaj układów automatyki opisanych w następnym podrozdziale. Jednak w prostych systemach wystarczyć może prosty układ dalmierza o strukturze podobnej do schematu podanego na rysunku 2.27.

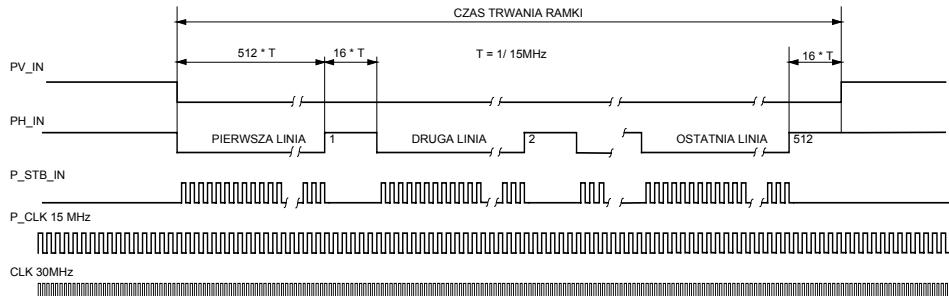


Rys. 2.27. Prosty system umożliwiający ręczne ustawianie stopnia ostrości obrazu.

Układ ten umożliwia automatyczne umieszczenie przez układ optyczny kamery badanego obiektu w pobliżu płaszczyzny ostrości (a przynajmniej w obszarze zaliczanym do głębi ostrości), co jest warunkiem uzyskania dobrej jakości obrazu. Brak spełnienia tego warunku prowadzi do uzyskania obrazu, który będzie trudny i kłopotliwy w dalszym przetwarzaniu.

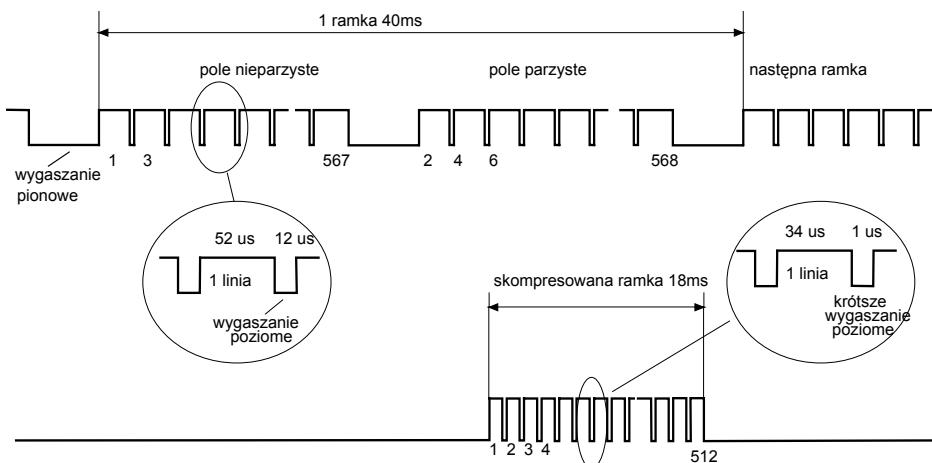
Po optycznym zarejestrowaniu obrazu następuje jego elektroniczna analiza zgodnie z zasadami używanego standardu telewizyjnego. Telewizyjna analiza obrazu – to specyficzny rodzaj przetwarzania optoelektronicznego, w którym zmienemu w czasie i w przestrzeni rozkładowi światel przekazywanej sceny jest

przyporządkowany wzajemnie jednoznacznie elektryczny przebieg, zależny jedynie od czasu. Strukturę sygnału produkowanego przez kamerę TV prezentuje rysunek 2.28.



Rys. 2.28. Struktura sygnału produkowanego przez kamerę TV.

Aby zapewnić kwadratowy kształt pola obrazu cyfrowego należy wykorzystać jedynie 3/4 długości każdej linii pochodzącej ze standardowej kamery (patrz rys. 2.2). Ze względu na typowe pojemności pamięci i ich sposób zorganizowania (np. podział na kolumny i wiersze) wygodnie jest przyjąć podział obrazu analizowanego na 512 linii podzielonych na 512 punktów (okno analizowane). W przypadku stosowania standardowych kamer (PAL) powoduje to dalsze zmniejszenie pola obrazu z 575 linii widocznych na ekranie do 512 linii. Dla zachowania kwadratowego pola obrazu zmniejszona długość analizowanej linii wynosi:  $3/4 \times 512/575 \times 52 \mu\text{s}$  ( $52 \mu\text{s}$  - to standardowy czas trwania widocznej części jednej linii). W przypadku gdy użytkownik dysponuje kamerą o gorszej rozdzielczości niż  $N_{ver} \times N_{hor}$  ( $767 \times 575$  dla PAL), reprezentacje sąsiednich punktów obrazowych będą pochodząły z tych samych elementów aktywnych CCD, co zubaża pozyskiwane dane obrazowe.

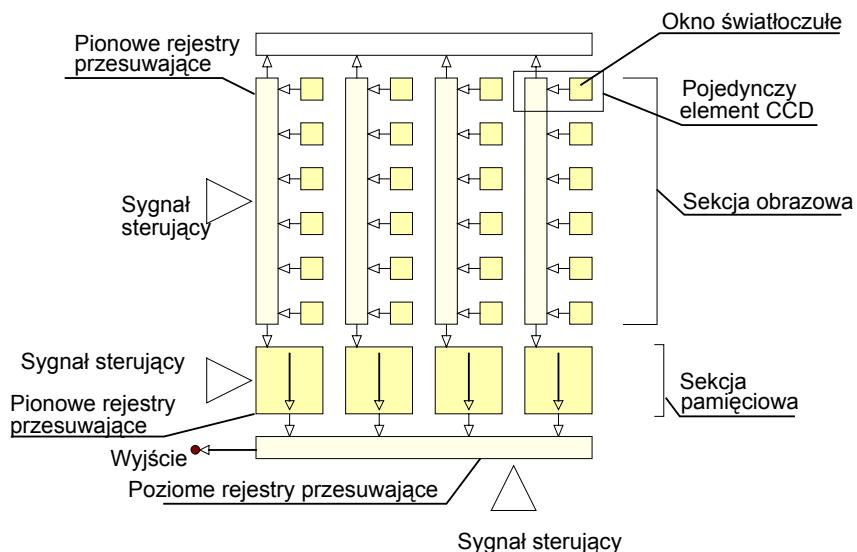


2.29. Typowy sygnał z kamery TV (u góry) i sygnał specjalnie preparowany do przetwarzania komputerowego (u dołu).

Sygnal z typowej kamery TV jest niekorzystny z punktu widzenia techniki komputerowego przetwarzania obrazów. Wynika to z faktu, że sygnał taki budowany jest zwykle na zasadzie tak zwanego wybierania międzyliniowego. Chodzi o to, że w przypadku najpopularniejszych rozwiązań przetworników CCD (tj. IT lub FT) wymogi standardu PAL osiągane są kilkoma metodami (np. przesuwanie centrum kolekcji ładunku czy odczytywanie różnych kombinacji wierszy w kolejnych ramkach), w związku z czym czas trwania jednego kadru (1/25 sekundy) odpowiada przesyłaniu nie całego obrazu, tylko jednego półobrazu - na przykład najpierw półobrazu złożonego z samych nieparzystych linii obrazu, a potem to samo odpowiednio dla linii parzystych. Taki sposób akwizycji obrazu wymaga zwykle jego specjalnego przekodowywania przed operacją konwersji obrazu do postaci cyfrowej. Zasadę takiej konwersji pokazano na rysunku 2.29.

### 2.3.4 Przetwornik CCD

Przetwarzanie obrazu optycznego na sygnał wizyjny jest dokonywane zgodnie z zasadami określonego systemu telewizyjnego i zachodzi w urządzeniach, które – dla odróżnienia od innych przetworników podobnego rodzaju – dogodnie jest ująć wspólną nazwą **wizyjnych przetworników analizujących** lub elektrooptycznych analizatorów obrazu.



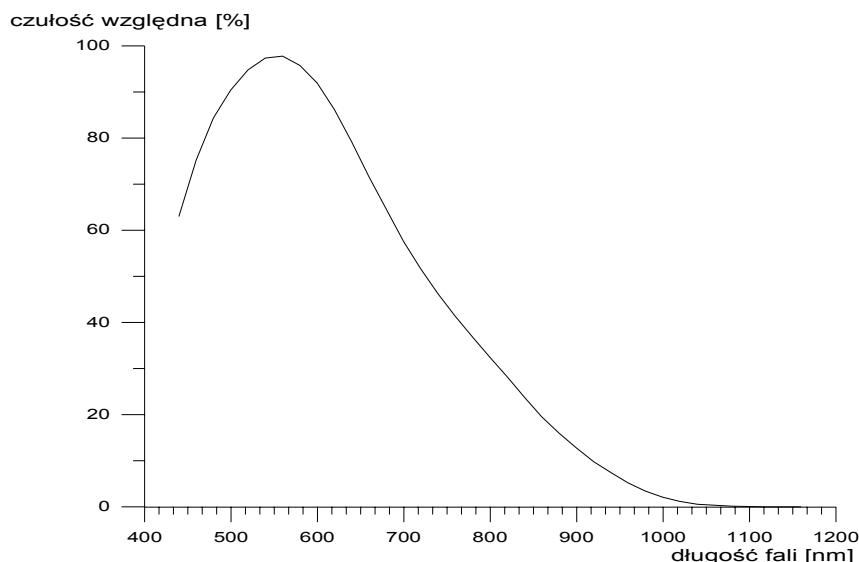
Rys. 2.30. Ogólna budowa przetwornika CCD typu INTER - LINE TRANSFER.

Obecnie najpowszechniej stosowany w komercyjnych kamerach wideo jest przetwornik CCD (ang. *Charge-Coupled Device*) – przyrząd o sprzężeniu ładunkowym. Wykonywany jest w postaci jednego układu scalonego, którego matryca obrazowa zawiera setki tysięcy elementów światłoczułych (np. przetworniki typu

FIT–CCD składają się z nie mniej niż 450000 elementów światłoczułych). Ogólna struktura jednego z rodzajów przetwornika obrazowego CCD (typu INTER–LINE TRANSFER) przedstawiona została na rys. 2.30. Składa się on z części transportującej i części obrazowej – światłoczułej.

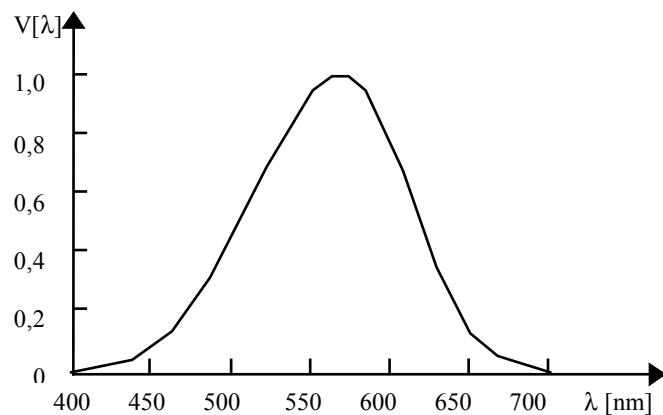
Aby na wyjściu przetwornika uzyskać sygnał elektryczny będący ładunkową reprezentacją obrazu optycznego skupionego na części światłoczułej, należy odpowiednio dobrą sygnały sterujące. Wytworzone w poszczególnych komórkach porcje ładunku elektrycznego, których wielkość jest proporcjonalna do natężenia padającego na poszczególne komórki światła, są przemieszczane najpierw w lewo, do linii zbierającej ładunki z poszczególnych elementów światłoczułych, a następnie w dół do głównej części transportującej. Z głównej części transportującej sygnał wizyjny, odpowiadający obrazowi skanowanemu, kolejno linia po linii wychodzi na zewnątrz w postaci sygnału elektrycznego.

Zakres przetwarzanego promieniowania stosowanych przetworników CCD wynika z zastosowania w nich krzemu (Si) jako podstawowego materiału półprzewodnikowego. Rozciąga się on od 300 nm (ultrafiolet) do 1300 nm (bliska podczerwieni), z maksimum w okolicach 750 nm (rys. 2.31). W celu dostosowania tej charakterystyki do czułości spektralnej oka ludzkiego (350÷750 nm z maksymalną wartością 555 nm) stosuje się odpowiednie filtry optyczne.



Rys. 2.31. Wykres czułości spektralnej typowego elementu CCD.

Dla porównania na rysunku 2.32. przedstawiono wykres wzajemnej czułości oka dla różnej długości fal świetlnych. Porównanie rysunków 2.31 i 2.32 pozwala zorientować się, jak odmienne są sposoby „widzenia” tych samych obrazów przez oko ludzkie i przez system wyposażony w kamerę z przetwornikiem CCD.

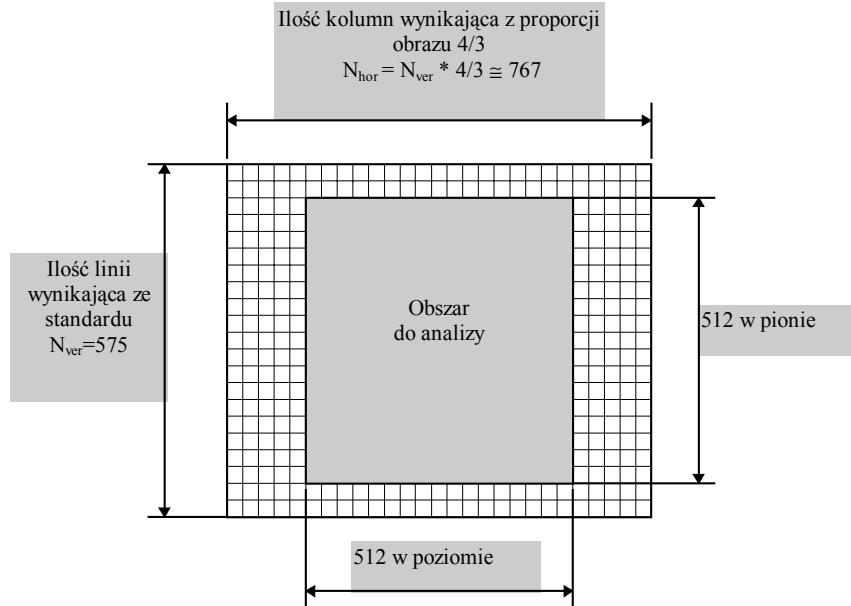


Rys. 2.32. Krzywa względnej widmowej czułości oka.

Przetwornik CCD najlepiej pracuje w zakresie temperatur od –66 do ok. 70° C. Jest to spowodowane właściwym dla wszystkich półprzewodników procesem ciągłej generacji termicznej par nośników wewnątrz materiału. Powyżej pewnej temperatury (dla krzemu ok. 70° C) proces ten zagraża działaniu przetwornika w ten sposób, że liczba elektronów wygenerowanych pod wpływem promieniowania świetlnego (a więc niosąca interesujące nas informacje) będzie znikoma w stosunku do ilości nośników wytworzonych pod wpływem temperatury. Grozi to całkowitą utratą informacji o obrazie. Fizycznym ograniczeniem szybkości działania przetwornika CCD jest z kolei maksymalna prędkość, z jaką w danym materiale mogą poruszać się nośniki ładunku. W celu jej zwiększenia strukturę wyposaża się w tzw. kanał zagrzebany (*ang. buried channel*). Zwiększenie prędkości przemieszczania się nośników jest wtedy możliwe ze względu na to, że w głębi półprzewodnika jest znacznie mniejsza liczba defektów (w stosunku do warstw przypowierzchniowych), których występowanie jest jedną z podstawowych przyczyn ograniczających prędkość nośników. Praktycznym ograniczeniem szybkości działania przetwornika jest także występowanie w strukturach CCD szkodliwych pojemności, niemniej spotykane są już przetworniki z częstotliwościami rzędu gigaherców.

Ponieważ struktura CCD jest praktycznie pozbawiona szumów, wpływ na dynamikę przetwornika mają układy z nim współpracujące. Z powodu niewielkiego poboru mocy i niewielkiej liczby kontaktów w układach z przetwornikami CCD ich niezawodność jest bardzo wysoka. Producenci oceniają ją na 200 000 do 500 000 godzin bezawaryjnej pracy.

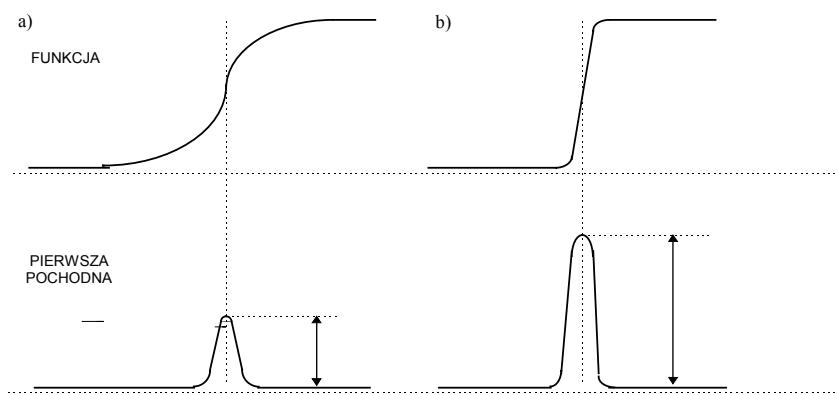
Obraz dostarczany przez układ optyczny kamery i przetwornik CCD jest z reguły najlepszy w centralnej swej części, natomiast bywa zniekształcony w pobliżu marginesów. Z tego powodu wybierając obszar, który poddajemy przetwarzaniu analogowo-cyfrowemu celem uzyskania obrazu cyfrowego, lokalizuje się go w centrum obrazu telewizyjnego (rys. 2.33).



Rys. 2.33. Lokalizacja obszaru analizy cyfrowej na obrazie TV.

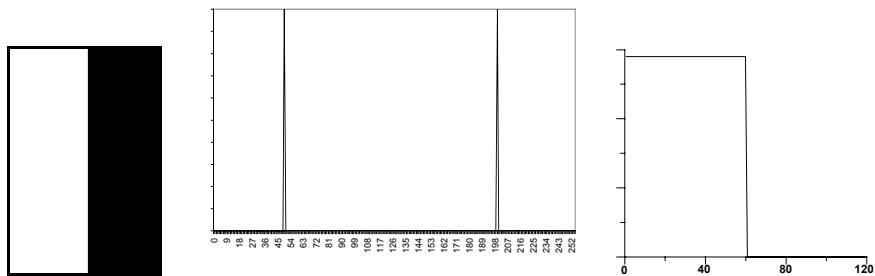
### 2.3.5 Automatyka kamery

Dla zapewnienia sobie właściwej jakości obrazu cyfrowego trzeba zadbać o to, by obraz dostarczany przez kamerę TV był odpowiednio naświetlony i odpowiednio kontrastowo ustawiony. Można tu bazować na ręcznej regulacji, ale stosunkowo łatwo można skonstruować algorytm automatycznego ustawiania ostrości obrazu, oparty na analizie pochodnej obrazu w obszarach odpowiadających automatycznie wykrywanym krawędziom obiektów. Zasada działania tego algorytmu podana jest na rysunku 2.34.

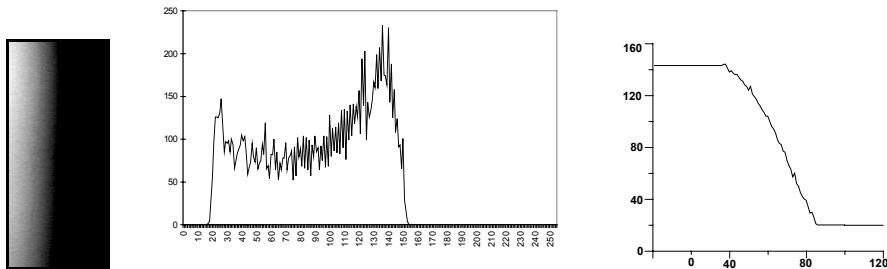


Rys. 2.34. Przykładowe przebiegi funkcji zmian stopni szarości pikseli oraz ich pochodne, dla obrazu nieostrego (a) i obrazu ostrego (b).

Przebieg i profil krawędzi wybranych obiektów na obrazie przedstawiono na rysunkach 2.35 i 2.36 - odpowiednio dla obrazu ostrego i dla obrazu nieostrego. Na rysunkach tych pokazano także histogramy<sup>16</sup> odpowiednich obrazów.



Rys. 2.35. Idealna krawędź, jej profil linii i histogram.



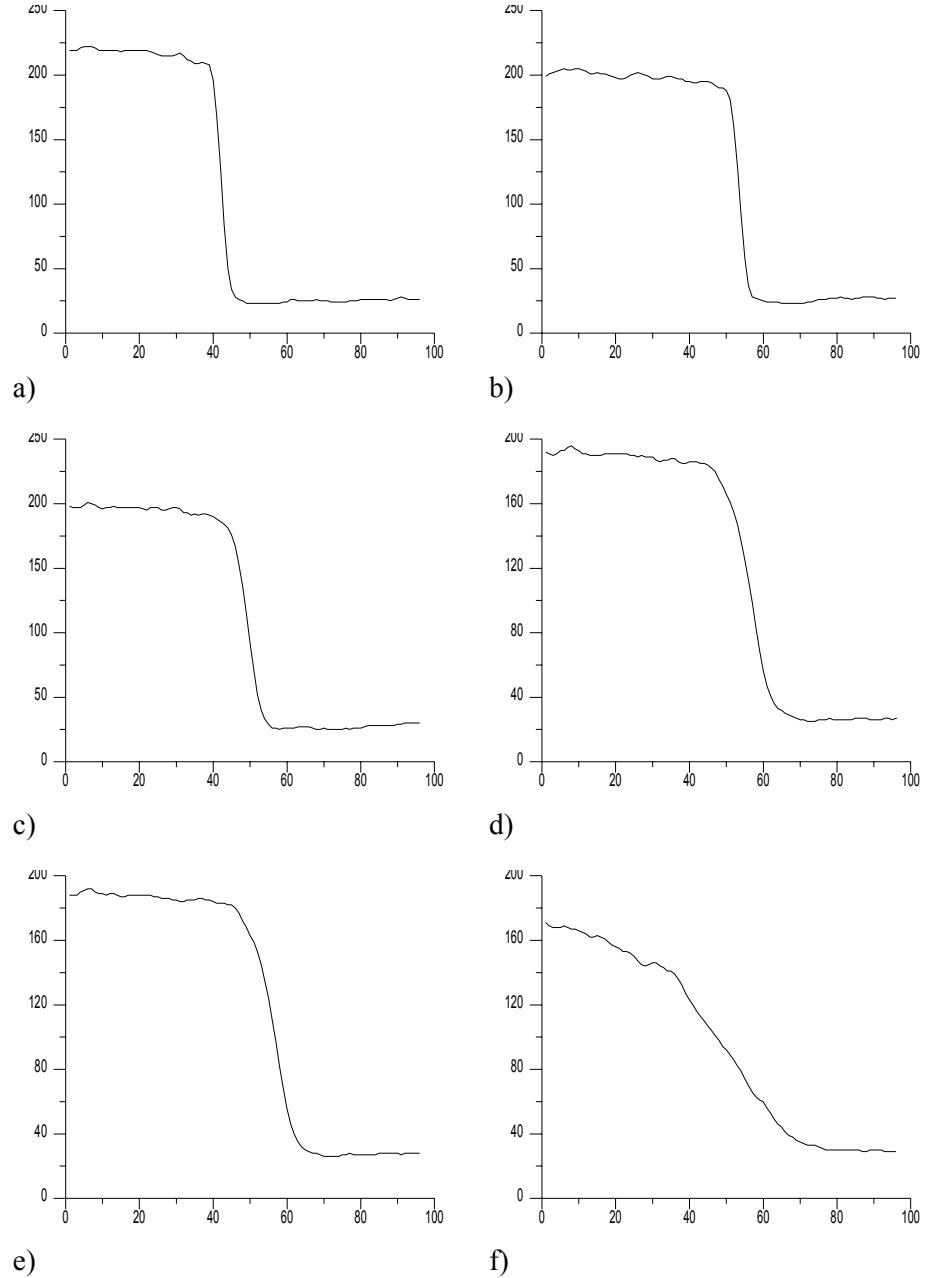
Rys. 2.36. Nieostra krawędź, jej profil linii i histogram.

Przy ocenie ostrości obrazu na podstawie histogramu lub przekroju linii zasadnicze znaczenie ma wybór takiego fragmentu obrazu, aby rozmycie krawędzi wpływało z jak największą wagą na kształt histogramu. Kąt pomiędzy analizowaną linią a krawędzią powinien być jak najbardziej zbliżony do kąta prostego, ze względu na pewne rozmycie występujące nawet w subiektywnie najostrzejszych krawędziach. Zależność kształtu profilu linii od wspomnianego kąta przedstawiona jest na rys. 2.37.

Należy tu rozróżnić dwa podejścia do wyboru i analizy fragmentu obrazu: globalne i lokalne. Podejście globalne stosujemy, gdy pozyskujemy po raz pierwszy obraz, którego jakość należy poprawić. Chodzi o to, aby wybrać np. jeden z kilku możliwych obiektów, będący najbardziej miarodajnym. Natomiast podejście lokalne stosujemy, gdy mamy już zlokalizowany np. właściwy obiekt i określamy, jak zachowuje się jego krawędź w wyniku zmian nastawienia ostrości. Aby przeprowadzić właściwą analizę globalną należy więc określić na wstępie, jak duży może być obszar pośredni (rozostrzona krawędź) pomiędzy dwoma obszarami, przy maksymalnej nieostrości obrazu. Eksperymenty wykazały jednak dość

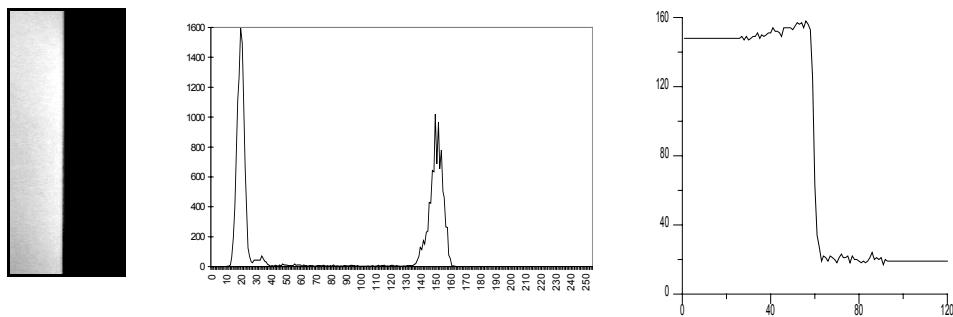
<sup>16</sup> Pojęcie histogramu i elementy analizy obrazu oparte na technice histogramowej omówione będą dokładniej w rozdziale 3.

dużą rozbieżność tego parametru w zależności od ogniskowej i od konstrukcji obiektywu.

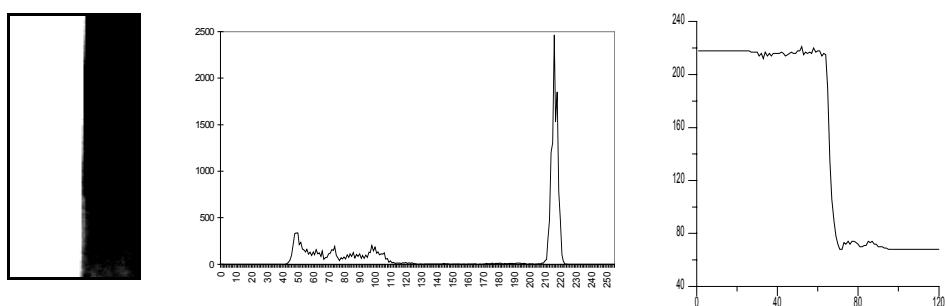


Rys. 2.37. Zależność kształtu profilu linii od kąta pomiędzy linią i krawędzią:  $90^\circ$  (a),  $45^\circ$  (b),  $20^\circ$  (c),  $10^\circ$  (d),  $5^\circ$  (e),  $2^\circ$  (f). Wyniki dla obrazu ostrego.

Rodzaj oświetlenia ma duży wpływ na intensywność obrazu, a co się z tym wiąże - na takie parametry jak przysłona i głębia ostrości, czas migawki oraz nastawienie poziomu bieli, itd. Dla wzroku ludzkiego płynne zmiany oświetlenia nie są tak zauważalne jak dla systemu wizyjnego, ponieważ człowiek wrażliwy jest na różnicę poziomów (kontrast), a nie na ich absolutną wartość. Zilustrowano to na rys. 2.38 i rys. 2.39, na których pokazano histogramy oraz profile linii obrazu ostrego, który był oświetlany światłem dziennym, a następnie światłem sztucznym o wyraźnie innym natężeniu światła. Różnica pomiędzy wynikami testów jest duża. Widać wyraźne „przesunięcie” histogramu w kierunku wartości wyższych, czyli rozjaśnienie obrazu. Efektem niepożądanym jest utrata „piku czerni” ze względu na pojawiające się odbicia i nierównomierność pokrycia powierzchni. Jak widać, w takim przypadku analiza obrazu metodą profilu linii jest korzystniejsza - kształt linii jest mniej zależny od rodzaju oświetlenia.



Rys. 2.38. Fragment obrazu ostrego oświetlonego światłem dziennym, jego histogram i przekrój linii.

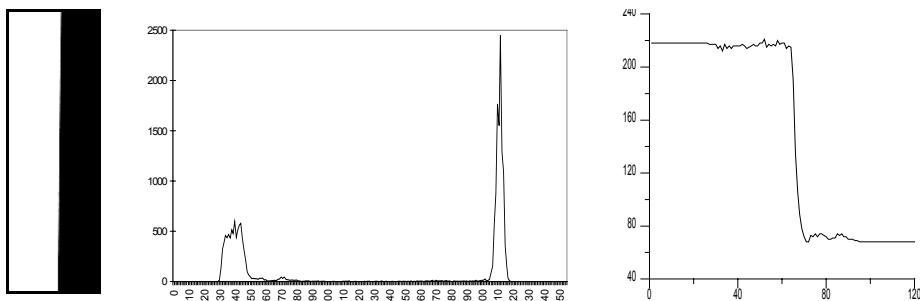


Rys. 2.39. Fragment obrazu ostrego oświetlonego światłem sztucznym, jego histogram i przekrój linii.

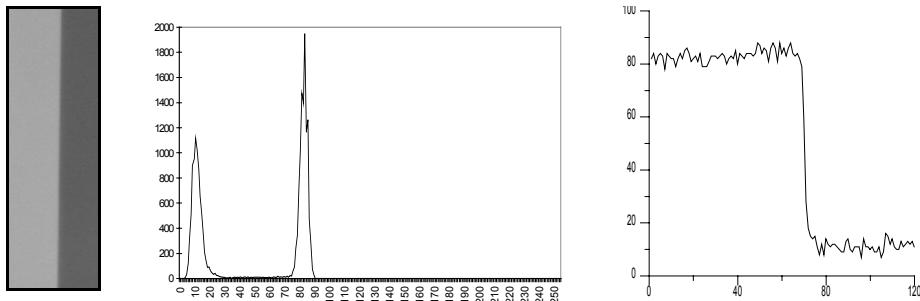
Przeprowadzano również próby mające na celu określenie wpływu nierównomiernego i zmieniającego się oświetlenia oraz cieni na możliwości oceny ostrości obrazu. Obiekt (czarny kwadrat) przesuwany był w obrębie obszaru nierównomiernie oświetlonego, gdzie występowały również cienie. Ostrość nastawiona była właściwie. Histogramy fragmentów zawierających zawsze ten sam fragment krawędzi różnią się od siebie, jednak wydaje się, że można dopuścić pewne

zmiany oświetlenia podczas analizy kolejnych obrazów w trakcie „dochodzenia” do ostrości. Cechy obrazu ostrego można było odróżnić w każdym przypadku.

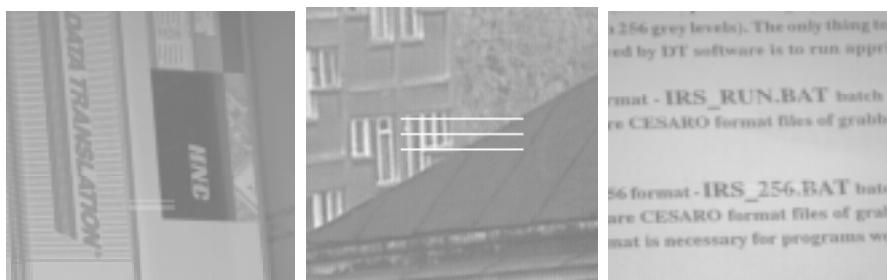
Na rysunkach 2.40 i 2.41 przedstawiono wpływ przysłony na histogram obrazu ostrego. Wraz ze zwiększeniem przysłony (od 2 do 11) widać „przesuwanie” się histogramu w kierunku wartości mniejszych i jednocześnie zmniejszanie kontrastu obrazu („piki” zbliżają się do siebie). Jednocześnie zwiększenie przysłony powoduje zwiększenie głębi ostrości. Zbyt mała wartość przysłony powodowała natomiast prześwietlenie obrazu i krawędzie stawały się niewykrywalne.



Rys. 2.40. Wykrywanie krawędzi dla obiektywu kamery otwartego do wartości „Przysłona 2”.



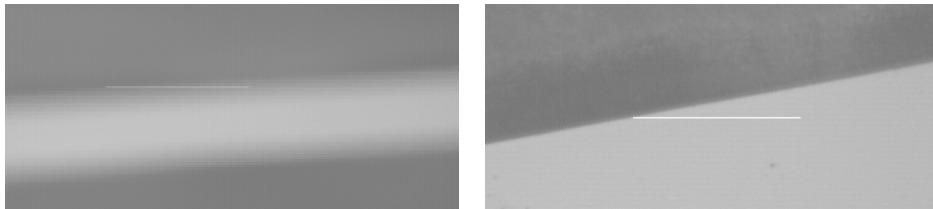
Rys. 2.41. Wykrywanie krawędzi dla obiektywu kamery zamkniętego do wartości „Przysłona 11”.



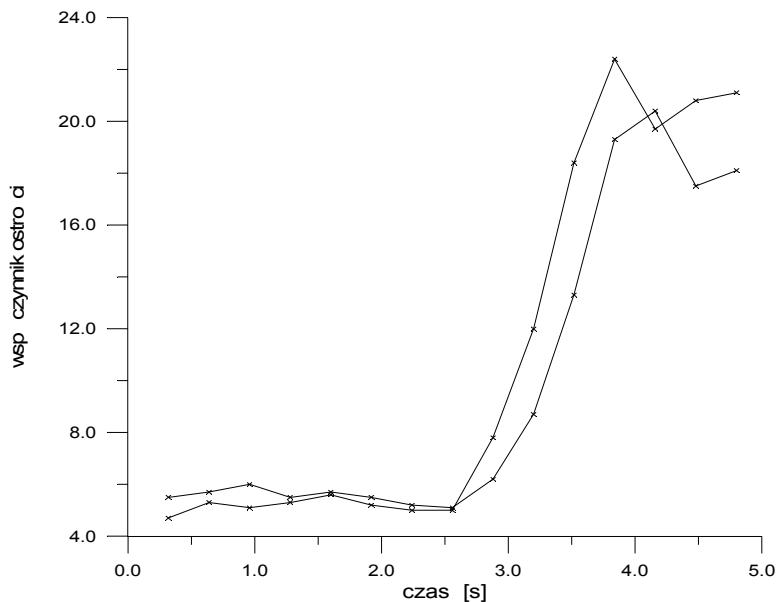
Rys. 2.42. Rozmieszczenie linii, według których badana jest pochodna sygnału przy automatycznym ustawianiu ostrości obrazu.

Na rysunku 2.42. pokazano lokalizację automatycznie wybieranych linii obrazu, wzdłuż których wybierane są funkcje opisane wyżej, stanowiące podstawę procesu automatycznego wyostrzania.

Efekty procesu automatycznego wyostrzania pokazano na rysunku 2.43. Rysunek ten sygnalizuje jednak pewien problem, jaki może się pojawić przy realizacji procesu automatycznego wyostrzania obrazu. Po pierwsze badania wykazały, że krawędź ostrego obrazu rzeczywistego nie jest funkcją skokową. Dlatego nie możemy poszukiwać obrazu, który posiada maksymalną różnicę pomiędzy dwoma sąsiadującymi punktami, gdyż może się zdarzyć, iż obraz ten nie będzie obrazem najbliższym. Przejście pomiędzy płaszczyznami o różnych odcieniach występuje dla obrazów ostrych na ogólnie na trzech do pięciu pikselach. Ocena ostrości obrazu, poprzez poszukiwanie tylko jednej największej różnicy, może doprowadzić do błędnych wniosków, np. ze względu na występujące zakłócenia.



Rys. 2.43. Obraz na początku i na końcu procesu automatycznego wyostrzania fragmentu obrazu .

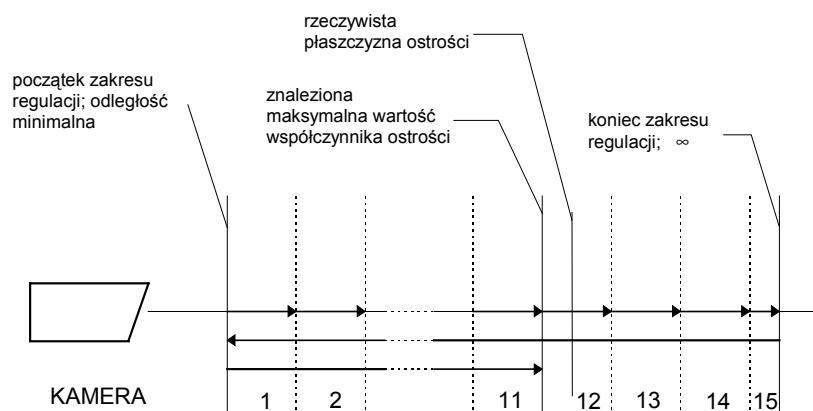


Rys. 2.44. Przebieg procesu automatycznego wyostrzania obrazu.

Po drugie - w miarę poruszania pierścieniem ostrości obiekty na obrazie przesuwają się nieznacznie. Analizowana krawędź znajduje się więc w coraz to innym miejscu na tym odcinku. Sposób przemieszczania się krawędzi jest zależny od odległości obiektu od kamery, położenia obiektu na obrazie, ogniskowej obiektywu. Nie udało się znaleźć zależności opisujących te przemieszczenia i uwzględnić ich w algorytmie, w celu zmniejszania ilości danych poprzez stopniowe ograniczanie analizowanego fragmentu obrazu. Nie można też lokalizować krawędzi ani obiektu do którego ona należy, ponieważ cechy tej krawędzi zmieniają się dynamicznie wraz z regulacją ostrości. Dopiero w końcowej fazie regulacji ostrości można wyróżnić obiekty lub wyznaczyć cechy krawędzi, w początkowej fazie ze względu na duże rozmycie krawędzi oraz interferencję jest to niemożliwe. Wyznaczony fragment obrazu musi posiadać więc rezerwę taką, aby analizowana krawędź nie znalazła się w trakcie regulacji poza nim.

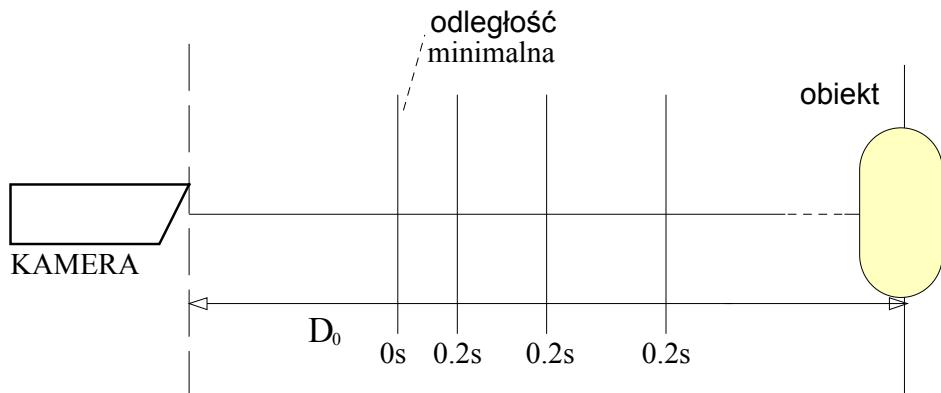
Trzeba mocno podkreślić, że proces uzyskiwania przez kamerę właściwej ostrości obrazu jest procesem rozłożonym w czasie (rys. 2.44), co oznacza, że dla szybko poruszających się obrazów konieczne jest sterowanie nadążne elementami nastawnymi optyki kamery.

Sposób ogniskowania obiektu kamery w trakcie tego procesu pokazano na rysunku 2.45. Jak wspomniano na wstępie, przy wyborze konkretnej kamery dość istotną rolę odgrywa możliwości sterowania silnikami zmiany parametrów obiektywu, co umożliwia automatyczne ustawianie ostrości. Podeczas projektowania systemów wizyjnych, których zadaniem jest m.in. śledzenie obiektów, powinno się jednak określić ich maksymalne prędkości, przy których system będzie w stanie je kontrolować. Prędkości te zależą od możliwości kamery, efektywności algorytmu oraz możliwości obliczeniowych komputera analizującego sekwencję obrazów i sterującego kamerą, a w przypadku zamocowania kamery np. na platformie obrotowo - uchylnej (por. rys. 2.21) - także od możliwości ruchowych tego stanowiska.



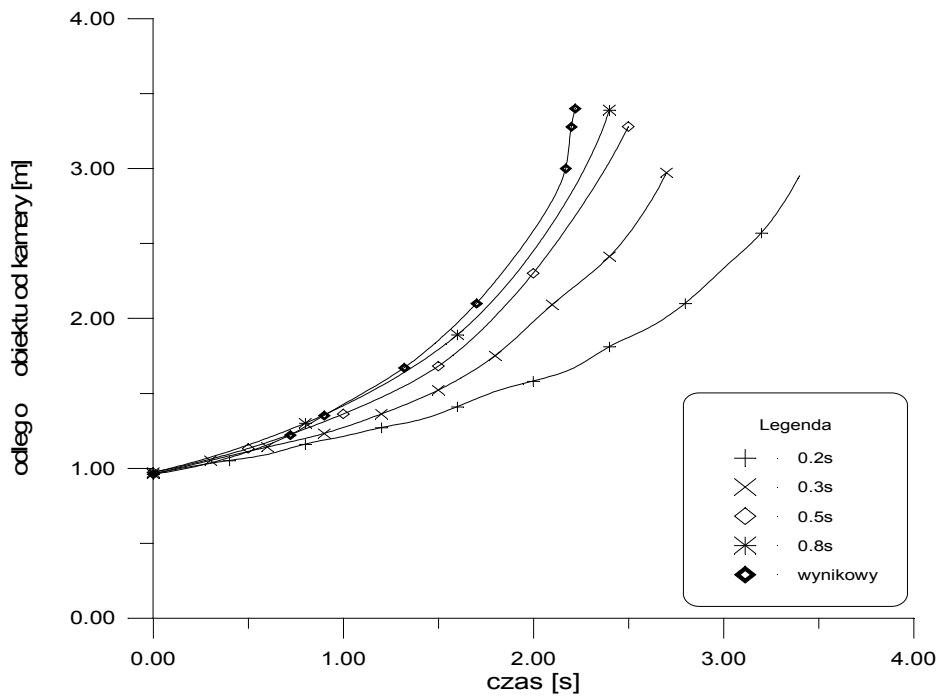
Rys. 2.45. Parametry wyostrzania obrazu.

Pomiary skuteczności automatycznego ustawiania przez kamerę ostrości obrazu mogą być wykonane na stanowisku, którego schemat pokazuje rysunek 2.46.

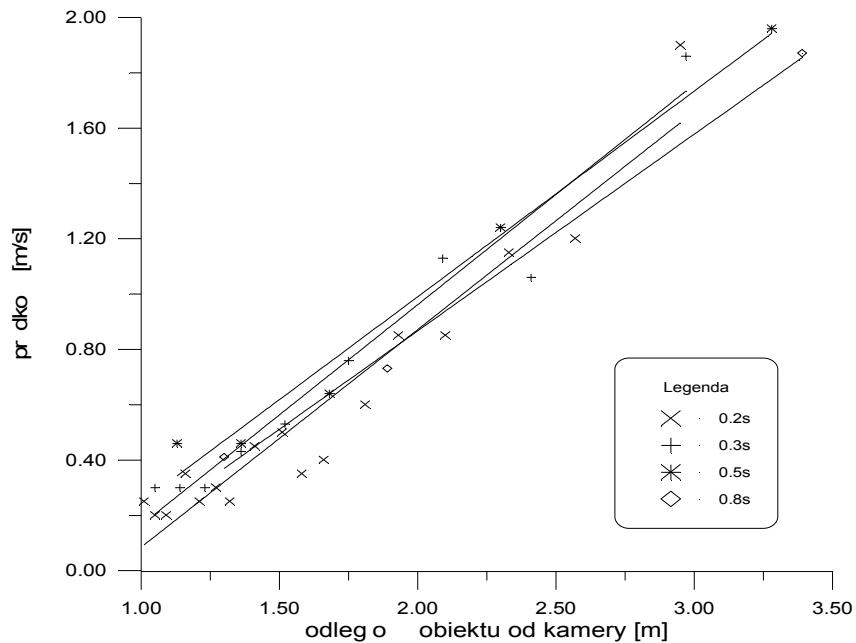


Rys. 2.46. Układ do badania dynamiki kamery sterowanej komputerem oraz badania prędkości obiektów śledzonych przez kamerę.

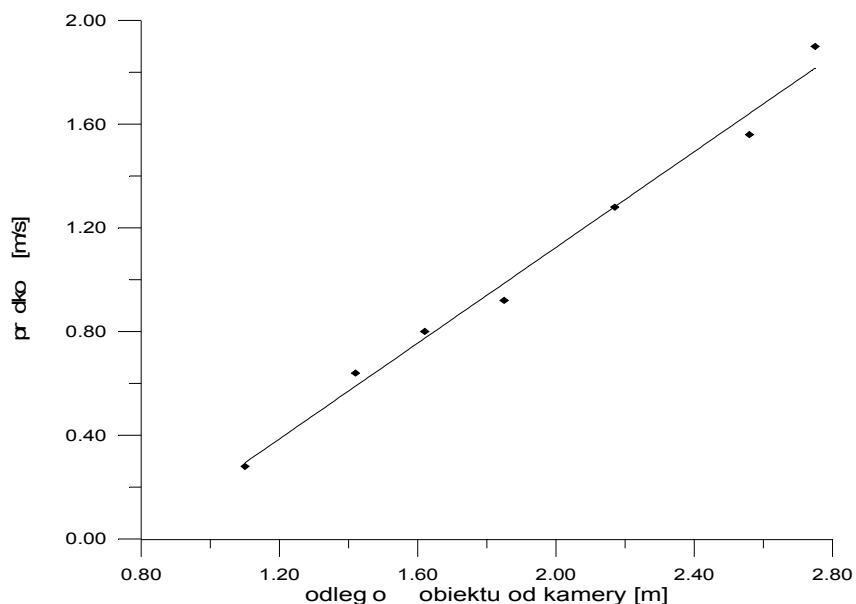
Na wstępie znaleziona może być zależność odległości płaszczyzny ostrości od kamery od czasu obracania pierścieniem ostrości (rys. 2.47).



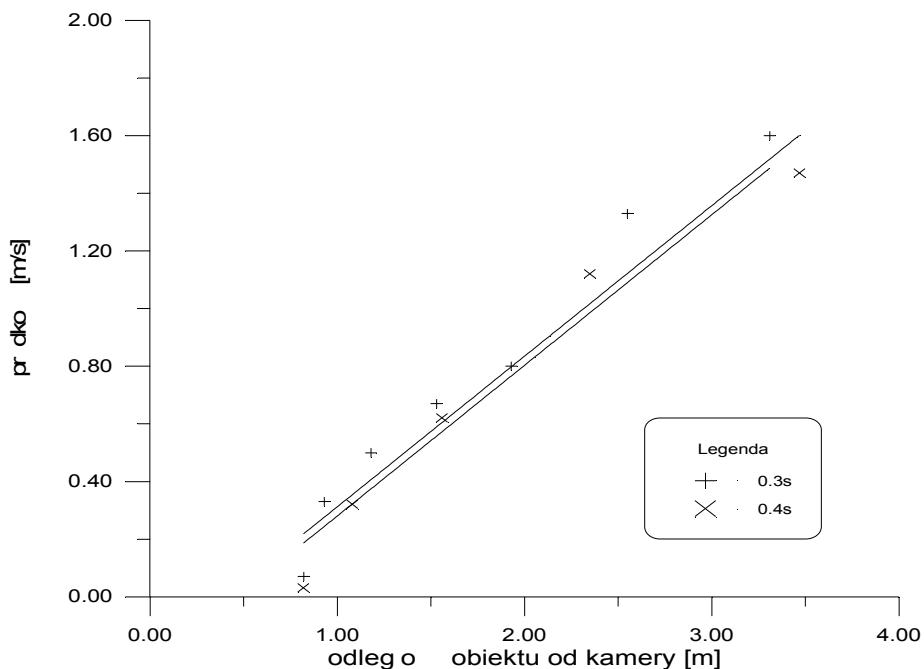
Rys. 2.47. Zależność odległości płaszczyzny ostrości od kamery, od czasu obracania pierścieniem ostrości.



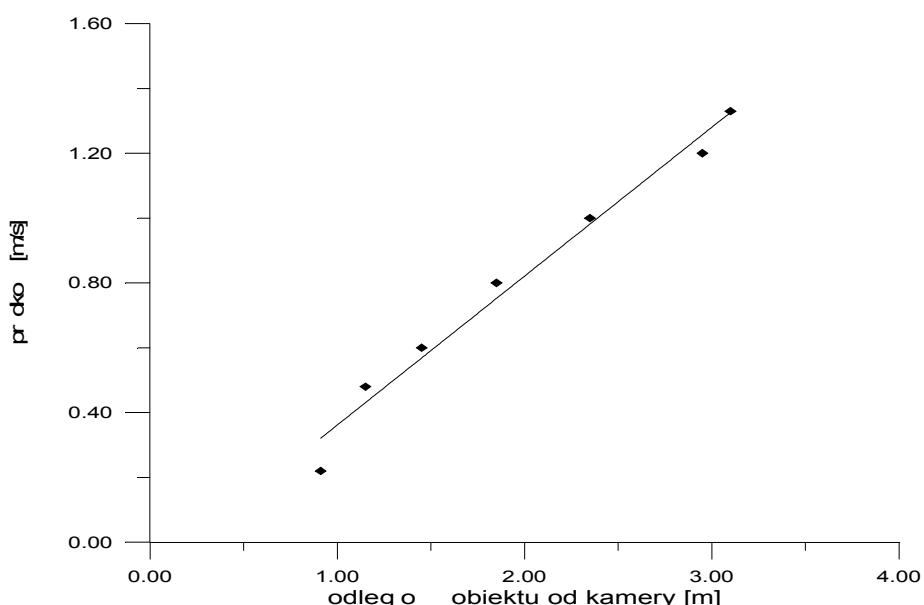
Rys. 2.48.a. Zależność maksymalnej możliwej prędkości obiektu (którego ostrość może być utrzymana) w zależności od odległości od kamery. Wyniki przy sterowaniu impulsami prostokątnymi o długościach 0.2s, 0.3s, 0.5s, 0.8s.



Rys. 2.48.b. Zależność maksymalnej możliwej prędkości obiektu (którego ostrość może być utrzymana) w zależności od odległości od kamery, przeliczona na pracę ciągłą silnika sterującego. Prosta regresji:  $y = 0.92x - 0.72$ .



Rys. 2.49.a. Zależność maksymalnej prędkości z jaką może się poruszać obiekt (którego wielkość w kadrze jest utrzymywana) od jego odległości od kamery, przy sterowaniu krótkimi impulsami o zmiennej długości (0.3s, 0.4s).

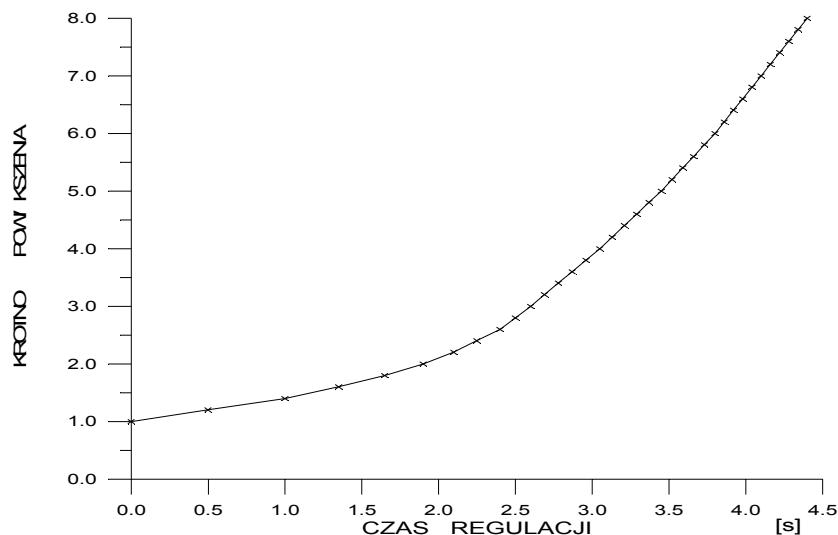


Rys. 2.49.b. Zależność jak na rys. 2.49.a, przeliczona na pracę ciągłą silnika regulującego ogniskową. Prosta regresji:  $y = 0.46x - 0.1$ .

Na tej podstawie określona może być maksymalna prędkość przesuwania płaszczyzny ostrości w zależności od oddalania się obiektu od kamery. Przedstawione jest to na rysunkach 2.48.a i 2.48.b. Rysunek 2.48.a przedstawia tę zależność przy poruszaniu silnikami kamery za pomocą krótkich sygnałów o różnej długości (0.2s, 0.3s, 0.5s, 0.8s), czyli jest to sytuacja podobna do tej, jaka ma miejsce w rzeczywistym dochodzeniu do ostrości w kilku lub kilkunastu krokach i śledzeniu obiektu. Rysunek 2.48.b pokazuje, z jaką prędkością może się przesuwać obiekt w przypadku ciągłej pracy silnika.

Podobne pomiary mogą być przeprowadzone w celu określenia możliwości utrzymania stałej wielkości obiektu w kadrze (test systemu *zoom*). Rysunek 2.49 przedstawia zależność maksymalnej prędkości, z jaką może się przemieszczać obiekt (którego wielkość na obrazie jest niezmienna) od odległości tego obiektu od kamery. Wykres na rysunku 2.49.a, dotyczy sterowania silnika zmieniającego ogniskową obiektywu krótkimi sygnałami (symulacja pracy kamery w rzeczywistym systemie), a wykres na rysunku 2.49.b dotyczy pracy ciągłej silnika. Należy zaznaczyć, iż podczas zmian ogniskowej obiektywu, wykonywana była również korekta ostrości.

W systemach śledzących obiekty na obrazie i utrzymujących ich stałą wielkość w kadrze wykorzystywana jest funkcja ZOOM obiektywu. Jej charakterystyka regulacyjna dana jest na rysunku 2.50.



Rys. 2.50. Zależność krotności powiększenia od czasu regulacji.

Należy podkreślić, że opisane wyżej eksperymenty dotyczące prędkości obiektów wykonane zostały dla sytuacji najbardziej niekorzystnych (długie ogniskowa obiektywu, słabe oświetlenie). W rzeczywistości warunki takie występują bardzo rzadko. Przykładowo, dla krótkich ogniskowych obiektywu i dużych wartości

przysłon, głębia ostrości jest duża. Może wystąpić sytuacja gdy wszystkie obiekty znajdujące się w odległości większej niż kilka metrów są ostre. W takiej sytuacji silnik ostrości obiektywu nie jest uruchamiany. Dopuszczalne prędkości obiektów poruszających się blisko kamery nie są duże, jednak rosną dość znacznie dla obiektów bardziej odległych. W doświadczeniach obiekty poruszały się wzduż osi optycznej obiektywu kamery. W przypadku śledzenia obiektów poruszających się w innym kierunku bierzemy pod uwagę jedynie składową prędkości równoległą do osi obiektywu kamery; prędkości obiektów mogą być więc większe.

Podsumowując te ogólne rozważania warto pamiętać konkretnie dane techniczne jednej z kamer często używanych dla cyfrowych systemów widzijnych.

Tabela 2.1. Wybrane parametry techniczne kamery typu CCD-SFI-5612P

Przetwornik	CCD, 1/2 cala, max. 568000 pikseli	(V)×(H)=625×909
	typ HQ-HAD Interline Transfer Sensor	(440000 el. obrazu)
min. oświetlenie	0.03 lx (*1)	
obiektyw: /jasność/ogniskowa	F=1,4 8.5 ÷ 68 mm	(8-krotny zoom)
migawka elektroniczna	1/50 ÷ 1/10000 s	8 podzakresów
zakres regulacji ostrości	od 1.1 m do $\infty$	
system telewizyjny	CCIR, 625 linii, 50Hz, PAL	
wyjście video	1 V <sub>pp</sub> / 75 $\Omega$ Y/C (S-VHS)	
wrażliwość spektralna	250÷1000 nm	
współczynnik korekcji gamma	0.45	
stosunek sygnału do szumu	> 46dB	
układ automatyki ostrości	TTL-Autofocus	
rozdzielcość obrazu (linie TV)	564 (H)	
zasilanie / pobór mocy	12 V / 4.8 VA	
zakres temperatur pracy	-30° ÷ 60° C	
wilgotność względna	do 95 %	
wymiary	85×96×265 mm	
masa	1150g	

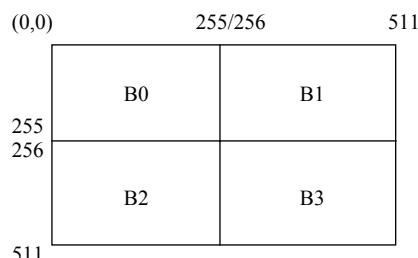
## 2.4 Działanie Frame Grabbera

Jak już było mówione, system typu Frame Grabber dokonuje wszystkich czynności związanych z przetwarzaniem obrazu do postaci cyfrowej, a także pozwala na zapisanie obrazu do pamięci RAM komputera. Taki jest typowy schemat postępowania, poprzedzający zwykle bardziej zaawansowane czynności związane z przetwarzaniem i analizą obrazu. Czasem jednak system Frame Grabber musi

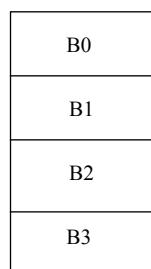
zapisać obraz do pamięci masowej komputera w postaci pliku. Warto w związku z tym wiedzieć, że plik ten wyposażany jest wtedy w nagłówek zawierający 18 bajtów. Zawartość tego nagłówka jest następująca:

- Pole 1 (1 bajt) = 0
- Pole 2 (1 bajt) = 0
- Pole 3 (1 bajt) = 131
- Pole 4 (5 bajtów) = 255, 0, 255, 0, 255
- Pole 5 (10 bajtów) - specyfikacja obrazu
  - 2 bajtów - współrzędna X początku obrazu
  - 2 bajtów - współrzędna Y początku obrazu
  - 2 bajtów - szerokość obrazu
  - 2 bajtów - wysokość obrazu
  - 1 bajt - liczba bitów na piksel
  - 1 bajt - bajt opisu obrazu
    - 5 bit umiejscowienie (0 - przy lewym dolnym rogu, 1 - przy górnym lewym rogu)
    - pozostałe bity są zarezerwowane

Przy zapisie w pamięci RAM z kolei obraz podzielony jest na cztery obszary (patrz rys. 2.51), w pamięci komputera jest to reprezentowane zgodnie z rys. 2.52. Znajomość tych formatów potrzebna jest w związku z ewentualnym samodzielnyim pisaniem programów przetwarzających obrazy - trzeba wtedy wiedzieć, w jakich obszarach RAM szukać określonych pikseli.



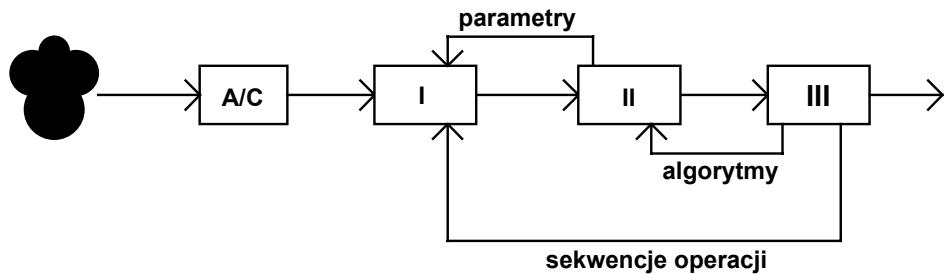
Rys 2.51. Podział obrazu na cztery obszary.



Rys 2.52. Sposób pamiętania czterech obszarów obrazu w pamięci.

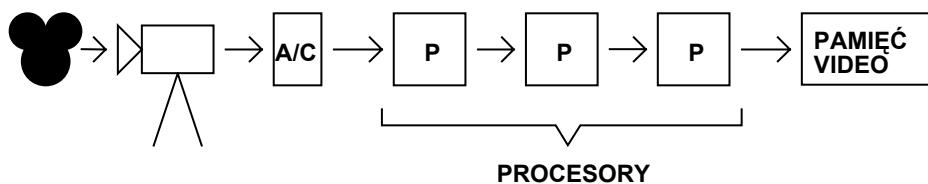
## 2.5 Poziomy dalszej obróbki obrazów

W algorytmach obróbki obrazów można wyróżnić kilka poziomów przetwarzania i analizy. Najczęściej przyjmuje się trzy poziomy (rys. 2.53). Najniższy poziom analizy obrazów ( I ), nazywany często obróbką wstępna (ang. *processing*) sygnału wizyjnego. Ma on na celu eliminację zakłóceń, wydobywanie obiektu z tła, detekcję krawędzi, ustalanie poziomów szarości obiektu na podstawie histogramu, równoważenie histogramu, itp. Średni poziom analizy obrazów ( II ) dokonuje segmentacji obrazu, lokalizacji obiektów, rozpoznania kształtu obiektu i wyróżnienia cech charakterystycznych tego kształtu. Najwyższy poziom ( III ) to analiza złożonej sceny w sensie analizy ruchu obiektu, bieżącego sterowania obiektem, zadawanie parametrów do obróbki i analizy obrazów na pozostałe poziomy, itp.

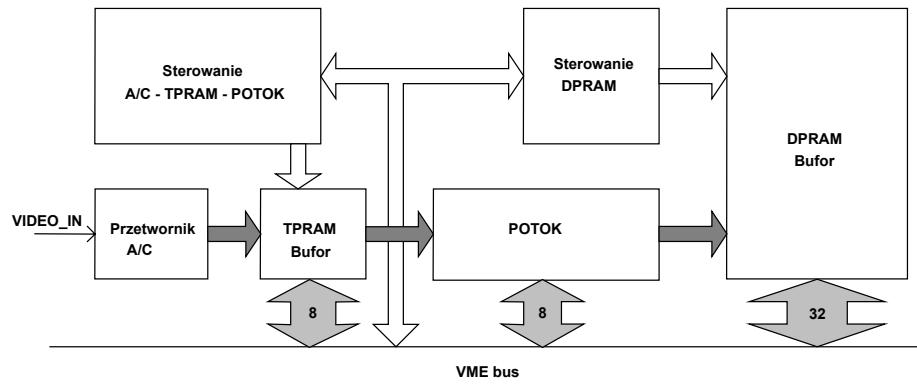


Rys. 2.53. Poziomy analizy obrazów.

Przedstawiona na rys. 2.53 struktura zadań systemu wizyjnego ukazuje ponadto występujące powiązania (sprzężenia) pomiędzy różnymi poziomami systemu wizyjnego. Wyniki II i III etapu przetwarzania obrazu mogą mieć wpływ na parametry i kolejność wykonywanych operacji na I najniższym poziomie oraz na algorytmy przetwarzania na II poziomie. Zadania systemu wizyjnego często bywają realizowane w strukturach wieloprocesorowych (rys. 2.54), które częściej organizowane są w postaci architektury potokowej (rys. 2.55), w której centralną rolę odgrywa potok specjalizowanych procesorów wykonujących w sposób szybki kolejne etapy procesu przetwarzania obrazu.



Rys. 2.54. Przykładowa struktura wieloprocesorowego (potokowego) systemu przetwarzania obrazów.



Rys. 2.55. Potokowa architektura wieloprocesorowego systemu do przetwarzania obrazów.

### 3 Klasyczne metody komputerowego przetwarzania obrazu

#### 3.1 Podział i ogólna charakterystyka algorytmów przetwarzania obrazu

W bogatej literaturze dotyczącej techniki *computer vision* zdefiniowana została wielka różnorodność algorytmów przetwarzania obrazu. Możliwych sposobów przekształcenia jednego obrazu w inny jest nieskończoność wiele, jednak większość nie posiada znaczenia praktycznego. Niemniej pozostała część przekształceń, mogących przynieść praktyczne efekty, jest na tyle liczna, że warto sklasyfikować ją na grupy ze względu na posiadane cechy. W tym rozdziale zajmiemy się czterema podstawowymi grupami przekształceń:

- przekształcenia geometryczne;
- przekształcenia punktowe (bezkontekstowe);
- przekształcenia kontekstowe (filtry konwolucyjne, logiczne i medianowe);
- przekształcenia widmowe (wykorzystujące transformację Fouriera);
- przekształcenia morfologiczne.

Na **przekształcenia geometryczne** składają się przesunięcia, obroty, odbicia i inne transformacje geometrii obrazu (rys. 3.1). Przekształcenia te wykorzystywane są do korekcji błędów wnoszonych przez system wprowadzający oraz do operacji pomocniczych. Będą one w tym rozdziale omówione skrótnie.



Rys. 3.1. Obraz „Lena” - oryginalny i po prostych przekształceniach geometrycznych.

Cechą charakterystyczną **punktowych przekształceń** obrazu jest to, że poszczególne elementy obrazu (punkty) modyfikowane są niezależnie od stanu elementów sąsiadujących. Innymi słowy - jeden punkt obrazu wynikowego otrzymywany

jest w wyniku wykonania określonych operacji na pojedynczym punkcie obrazu wejściowego. Dzięki takiej prostej regule operacje jednopunktowe mogą być wykonywane stosunkowo łatwo i szybko nawet na bardzo dużych obrazach. Najprostszymi operacjami punktowymi są: utworzenie negatywu, rozjaśnienie lub zaciemnienie obrazu wybranych punktów obrazu. Przekształcenia jednopunktowe (inaczej znane jako operacje anamorficzne) wykonywane są zwykle z zastosowaniem operacji LUT<sup>17</sup>, wykorzystującej z góry przygotowane tablice korekcji.

Przekształcenia przy pomocy **filtrów** polegają na modyfikacji poszczególnych elementów obrazu w zależności od stanu ich samych i ich otoczenia. Ze względu na tę kontekstowość operacje filtracji mogą wymagać dość długiego czasu (żeby wyprodukować jeden punkt obrazu wynikowego trzeba poddać określonym przekształceniom zarówno odpowiadający mu punkt obrazu źródłowego, jak i kilka (do kilkudziesięciu!) punktów z jego otoczenia. Jednak przekształcenia wchodzące w skład filtracji obrazu są algorytmicznie proste i regularne, a ponadto mogą być wykonywane na wszystkich punktach obrazu równocześnie bez konieczności uwzględniania przy przetwarzaniu jednego piksela wyników przetwarzania innych pikseli. Powoduje to, że do celów filtracji obrazów coraz częściej i coraz częściej stosuje się specjalizowane procesory dokonujące współbieżnego przetwarzania wszystkich punktów obrazu na drodze czysto sprzętowej.

**Przekształcenia widmowe** są pod wieloma względami podobne do filtracji kontekstowych, z tą jedynie różnicą, że kontekstem używanym w operacjach jest w istocie cały obraz. Technika przekształceń widmowych polega na tym, że najpierw w oparciu o cały obraz obliczane jest (z użyciem transformacji Fouriera) dwuwymiarowe widmo obrazu, następnie widmo to podlega określonej modyfikacji (na przykład usuwane są składowe o wysokich częstotliwościach) a następnie dokonywana jest rekonstrukcja obrazu (z użyciem odwrotnej transformacji Fouriera). Taki sposób przetwarzania obrazu pozwala na wyjątkowo precyzyjne kontrolowanie skutków dokonywanych transformacji, jednak z punktu widzenia jakości obrazów nie wnosi istotnie nowej jakości, a wiąże się z ogromnym obciążeniem obliczeniowym komputera (wykonywanie transformacji Fouriera). Zagadnienia przekształceń widmowych omawiane będą w oddzielnym rozdziale (rozdz. 4).

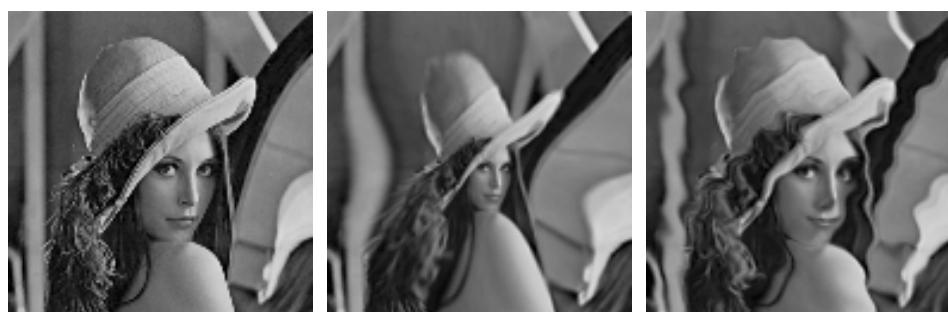
**Przekształcenia morfologiczne** różnią się od filtrów tym, że dany element obrazu nie jest modyfikowany zawsze, ale tylko wtedy, gdy spełniony jest zadany warunek logiczny. Przekształcenia morfologiczne wykonywane są zazwyczaj iteracyjnie, aż do zaistnienia zadanego warunku logicznego (zazwyczaj braku dalszych zmian w przetwarzanym obrazie). Zagadnienia przekształceń morfologicznych również omawiane będą w oddzielnym rozdziale (rozdz. 5).

---

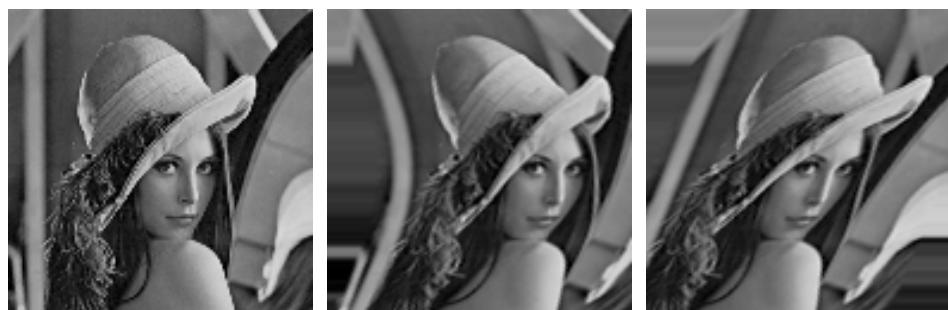
<sup>17</sup> Operacja ta będzie dalej dokładnie opisana.

Jak wynika z przytoczonego omówienia - asortyment pozostających do dyspozycji metod przetwarzania obrazów jest bardzo szeroki. Co więcej, z wielu przypadkach złożenie wielu różnych typów przekształceń obrazu daje ten sam wynik końcowy. W innych przypadkach pozornie identyczne sekwencje przekształceń dostarczają szokująco odmiennych wyników końcowych. Świadczy to o wielkiej złożoności problemów przetwarzania obrazu, którym teraz przyjrzymy się dokładniej w kolejnych podrozdziałach.

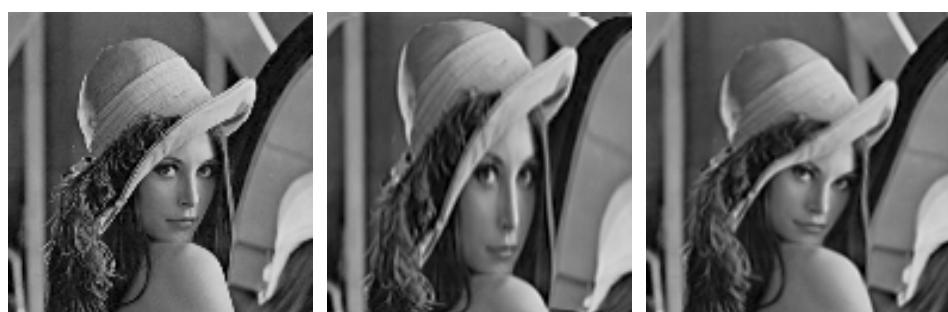
### 3.2 Przekształcenia geometryczne



Rys. 3.2. Obraz „Lena” i jego lokalne znieksztalcenia geometryczne.

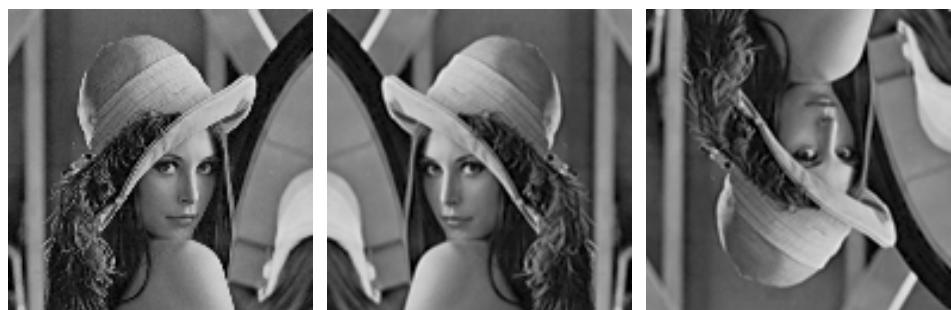


Rys. 3.3. Obraz „Lena” i jego globalne znieksztalcenia geometryczne.



Rys. 3.4. Obraz „Lena” i jego przekształcenia geometryczne polegające na lokalnej zmianie skali.

Przekształcenia geometryczne pełnią rolę pomocniczą w trakcie przetwarzania obrazu. Przykłady wybranych przekształceń geometrycznych znanego już z rozdziału 2 obrazu „Lena” pokazano na rysunkach 3.1 do 3.6, jednak nie należy rozważanej tu klasy przekształceń utożsamiać wyłącznie z tymi przykładami, gdyż możliwości prostych geometrycznych transformacji obrazu jest nieporównanie więcej.



Rys. 3.5. Obraz „Lena” i jego przekształcenia geometryczne polegające na obrotach.



Rys. 3.6. Obraz „Lena” i jego totalna deformacja geometryczna.

Przekształcenia geometryczne najczęściej wykorzystywane są do korekcji błędów geometrii obrazu takich, jak zniekształcenia poduszkowe, beczkowate i trapezowe. Źródłem takich zniekształceń są najczęściej niskiej jakości układy optyczne stosowane w kamerach video. Przekształcenia geometryczne wykorzystywane też mogą być do niwelowania niedostatków systemu akwizycji obrazu.

Przekształcenia geometryczne mogą występować jako samodzielne transformacje, ale mogą także być także wykorzystywane do wspomagania innych rodzajów przekształceń i analiz. Przykładem takiej pomocniczej roli przekształceń geometrycznych jest obracanie obrazu w trakcie wyliczania średnic Fereta (patrz rozdział 6).

### 3.3 Przekształcenia punktowe

#### 3.3.1 Przekształcenia oparte na przetwarzaniu pojedynczych punktów

Jak wspomniano wyżej, przekształcenia punktowe realizowane są zwykle w taki sposób, że wymagane operacje wykonuje się na poszczególnych pojedynczych punktach źródłowego obrazu, otrzymując w efekcie pojedyncze punkty obrazu wynikowego. Operacje te charakteryzuje się następującymi cechami:

- Modyfikowana jest jedynie wartość (np. stopień jasności) poszczególnych punktów obrazu. Relacje geometryczne pozostają bez zmian.
- Jeżeli wykorzystywana jest funkcja ściśle monotoniczna (rosnąca lub malejąca), to zawsze istnieje operacja odwrotna, sprowadzająca z powrotem obraz wynikowy na wejściowy. Jeżeli zastosowana funkcja nie jest ściśle monotoniczna, pewna część informacji jest bezpowrotnie tracona.
- Operacje te mają za zadanie jedynie lepsze uwidocznienie pewnych treści już zawartych w obrazie. Nie wprowadzają one żadnych nowych informacji do obrazu.

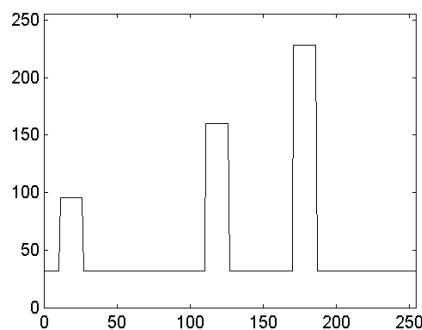
Bezpośrednio widocznym efektem przekształceń punktowych jest więc zawsze zmiana skali jasności obrazu bez zmiany geometrii widocznych na obrazie obiektów (rys. 3.7).



Rys. 3.7. Obraz „Lena” i jego przekształcenia w dziedzinie jasności.

Mimo bardzo prostego matematycznego charakteru przekształcenia punktowe bardzo radykalnie modyfikują subiektywne wrażenie, jakie uzyskujemy oglądając obraz. Czasem prowadzi to do krańcowego zniekształcenia obrazu, czasem jednak pozwala wykryć lub uwypuklić pewne cechy obrazu praktycznie niewidoczne, gdy się ogląda obraz oryginalny. Najprostszym przykładem takiego przekształcenia uwypuklającego pewne cechy obrazu może być transformacja polegająca na uwidocznieniu na wynikowym obrazie wyłącznie niektórych po-

ziomów szarości źródłowego obrazu - z pominięciem wszystkich innych. Na rysunku 3.8. pokazano przykładową regułę, według której takie przekształcenie może być wykonane. Widoczna na rysunku zależność podaje na pionowej osi wartości  $L'(m,n)$ , jakie zostaną przypisane pikselom na obrazie wynikowym, gdy na obrazie źródłowym odpowiednie piksele będą miały wartości  $L(m,n)$  podane na osi poziomej.



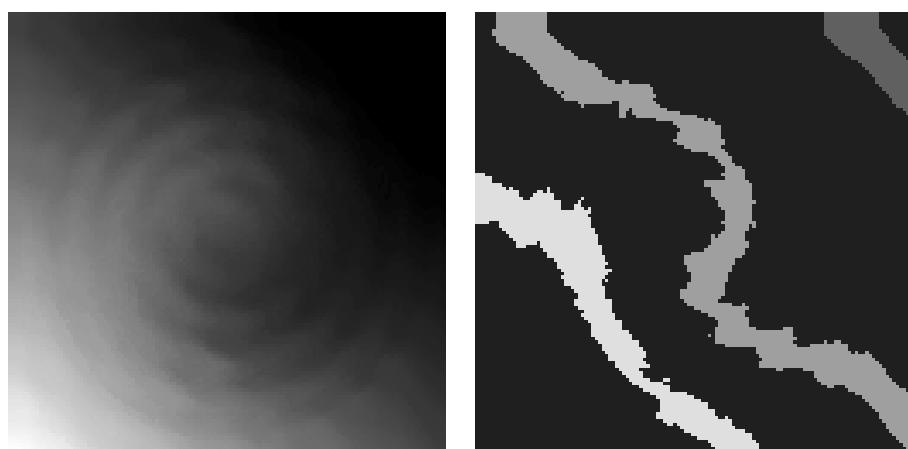
Rys. 3.8. Zależność pomiędzy jasnością pikseli obrazu przekształconego i obrazu źródłowego.

Jak widać w rozważanym przekształceniu spośród wielu stopni szarości wejściowego obrazu wybrano jedynie trzy konkretne wartości i tylko te trzy wartości są wiernie odtworzone na obrazie wynikowym. Wszystkie pozostałe stopnie szarości zostały zlikwidowane (dokładniej - zostały zamienione na wartości 0, co odpowiada barwie czarnej). Efekt takiego przekształcenia na obrazie „Lena” jest dość nieciekawy (rys. 3.9).



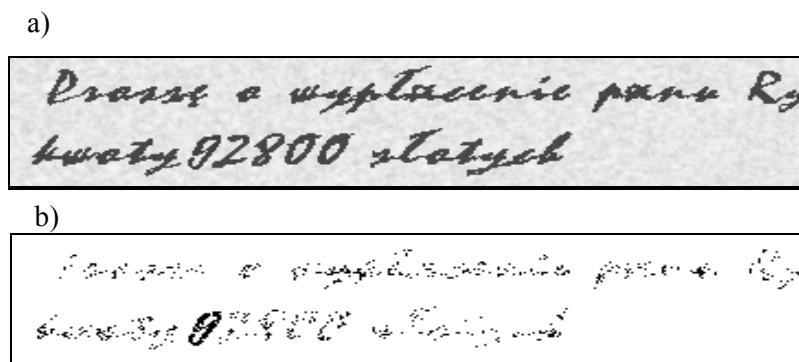
Rys. 3.9. Obraz „Lena” przed i po przekształceniu z wykorzystaniem zależności przedstawionej na rysunku 3.8.

Wobec zdecydowanie paskudnego wyglądu obrazu po przekształceniu wydobywającym poszczególne „poziomice” szarości można by przypuszczać, że przekształcenie takie nie ma sensu. Wniosek taki byłby jednak błędny - czasem takie „wyłuskanie” poziomic pozwala uwidoczyć szczegóły obrazu zdecydowanie ukryte czy mało widoczne. Rozważmy rysunek 3.10. Widoczny na nim sztuczny obraz w pełnej gamie szarości (po lewej) nie daje szczególnie czytelnego zarysu pokazanej na nim struktury. Tymczasem wykonanie na tym obrazie tej samej operacji, co na obrazie 3.9, ujawnia pewne właściwości i prawidłowości rozważanego obrazu, bardzo trudne do dostrzeżenia na obrazie źródłowym.



Rys. 3.10. Sztuczny obraz przed i po przekształceniu z wykorzystaniem zależności pokazanej na rysunku 3.8.

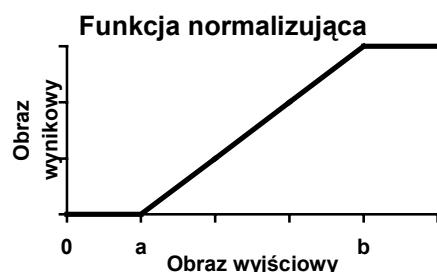
Na tej zasadzie można wykrywać i ujawniać subtelne różnice w obrazach na przykład w kryminalistyce (patrz rysunek 3.11), a także w diagnostyce medycznej (wykrywanie i sztuczne podkreślanie subtelnych różnic poszczególnych narządów i tkanek na zobrazowaniach rentgenowskich i ultrasonograficznych).



Rys. 3.11. a) Obraz tekstu, zawierającego fałszerstwo; b) obraz przekształcony z wykorzystaniem zależności pokazanej na rysunku 3.8. do ujawnienia fałszerstwa czeku.

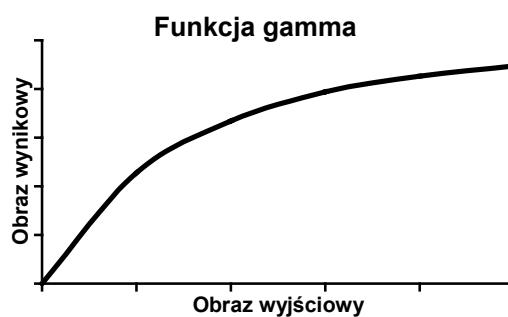
Wadą omówionych wyżej odwzorowań punktowych, wykorzystujących głównie arbitralnie określane przez badacza przekodowanie stopni szarości obrazu, jest to, że reguła przekształcenia (na przykład taka, jak pokazana na rysunku 3.12) musi być każdorazowo wymyślona przez osobę analizującą obraz - a to w ogólnym przypadku wcale nie jest łatwe. Źródłowy obraz zawiera punkty mające 256 rozróżnialnych poziomów szarości - skąd się w tej sytuacji dowiedzieć, które poziomy należy wzmacnić, a które osłabić? Możliwych kombinacji może być bardzo wiele i można znaczną część życia spędzić na bezowocnym poszukiwaniu tej właściwej. Dla ogólnego polepszenia jakości obrazu (bez określania z góry, czego się na obrazie szuka) można użyć następujących dwóch wygodnych przekształceń:

- **Normalizacja** - polegająca na sprowadzeniu przedziału zmian wartości punktów wyjściowego obrazu do pewnego, ustalonego zakresu. Operacja ta zazwyczaj poprzedza lub kończy inne przekształcenia obrazu (w szczególności arytmetyczne).



Rys. 3.12. Funkcja normalizująca.

- **Modulacja (korekcja) gamma** - mająca za zadanie redukcję nadmiernego kontrastu obrazu wyjściowego. Wykorzystywana funkcja ma postać:  $x \rightarrow x^\gamma$ , gdzie:  $\gamma$  - stały wykładnik, zazwyczaj liczba naturalna.



Rys. 3.13. Funkcja gamma.

Pewną pomoc w znalezieniu na obrazie cech trudno dostrzegalnych gołym okiem można uzyskać próbując przetwarzania obrazu z wykorzystaniem pewnych metod arytmetycznych.

### 3.3.2 Przekształcenia oparte na arytmetycznym przeliczaniu pojedynczych punktów

Na drodze arytmetycznej można wykonywać dowolne przekształcenia punktowe. Zakłada się wtedy, że dla każdej pary dyskretnych współrzędnych  $m$  i  $n$  (gdzie  $m \in [0, M-1]$  i  $n \in [0, N-1]$ ), wskazujących jeden piksel na obrazie źródłowym, obliczana jest pewna funkcja  $\Psi$ , w wyniku czego piksele obrazu wynikowego  $L'(m,n)$  są obliczane na podstawie pikseli obrazu źródłowego  $L(m,n)$  poprzez wykonanie operacji wynikającej z właściwości funkcji  $\Psi$ .

$$L'(m,n) = \Psi(L(m,n))$$

Przykładami najprostszych arytmetycznie realizowanych przekształceń punktowych mogą być:

- dodanie do obrazu liczby  $\chi$  (dodatniej lub ujemnej - patrz rys. 3.7 b)

$$L'(m,n) = L(m,n) + \chi$$

po wykonaniu przekształcenia może być konieczna normalizacja obrazu  $L'(m,n)$  celem zapewnienia warunku

$$L(m,n) \in \mathcal{N}$$

gdzie  $\mathcal{N}$  oznacza zbiór liczb całkowitych z przedziału  $[0, 2^B-1]$ , a  $B$  jest przyjętą liczbą bitów dla reprezentacji jednego punktu obrazu. Przekształcenie to przesuwa cały obraz w kierunku jaśniejszej lub ciemniejszej części przyjętej palety stopni szarości, co często pozwala zauważać obiekty lub szczegóły uprzednio niedostrzegalne gołym okiem.

- przemnożenie obrazu przez liczbę  $\gamma$  (dodatnią lub ujemną - patrz rys. 3.7 c),

$$L'(m,n) = \gamma L(m,n)$$

po wykonaniu przekształcenia może być konieczna normalizacja. Przekształcenie to zwiększa lub zmniejsza zróżnicowanie stopni szarości na obrazie. Zwiększenie zróżnicowania zwykle jednak okupione jest utratą części informacji w następstwie procesu niezbędnej normalizacji.

- zastosowanie funkcji potęgowej w celu podwyższenia kontrastu w obszarze dużych wartości  $L(m,n)$  (oznacza to generalne przyciemnienie obrazu z silnym różnicowaniem najciemniejszych partii);

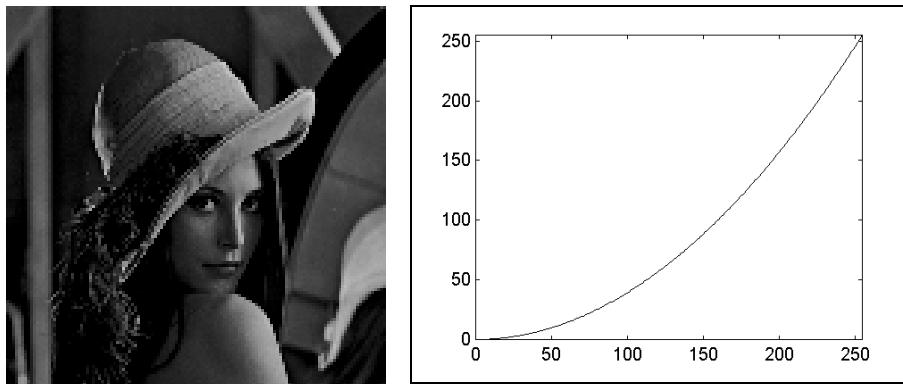
najczęściej stosuje się przypadki  $k = 2$  lub  $k = 3$ , patrz rys. 3.14 - 3.17).

$$L'(m,n) = (L(m,n))^k$$

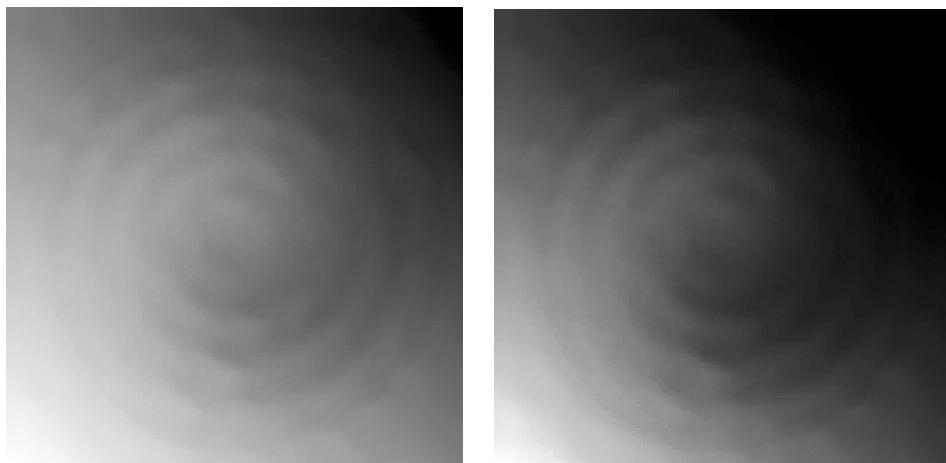
Funkcja potęgowa musi być odpowiednio wyskalowana celem zapewnienia warunku

$$L(m,n) \in \mathcal{N}$$

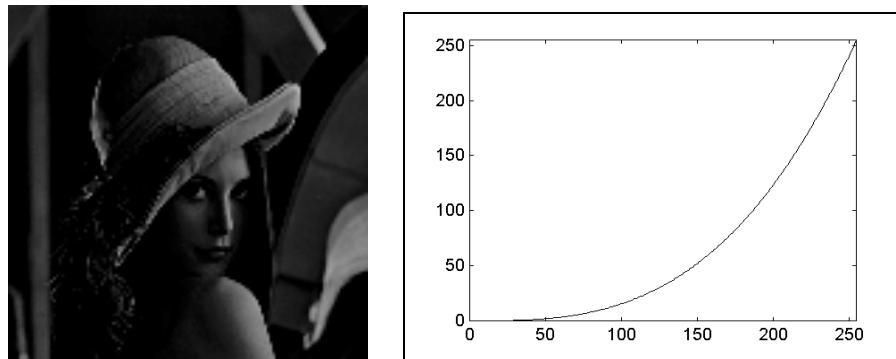
gdzie  $\mathcal{N}$  oznacza zbiór liczb naturalnych z przedziału  $[0, 2^B - 1]$ , a  $B$  jest przyjętą liczbą bitów dla reprezentacji jednego punktu obrazu.



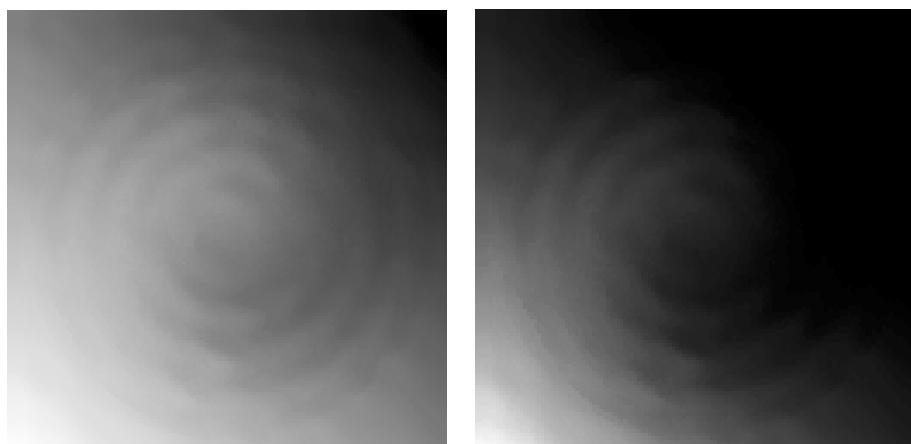
Rys. 3.14. Wynik przekształcenia obrazu „Lena” za pomocą funkcji  $L'(m,n) = (L(m,n))^2$  (po lewej) i wykres odpowiednio wyskalowanej funkcji kwadratowej (po prawej).



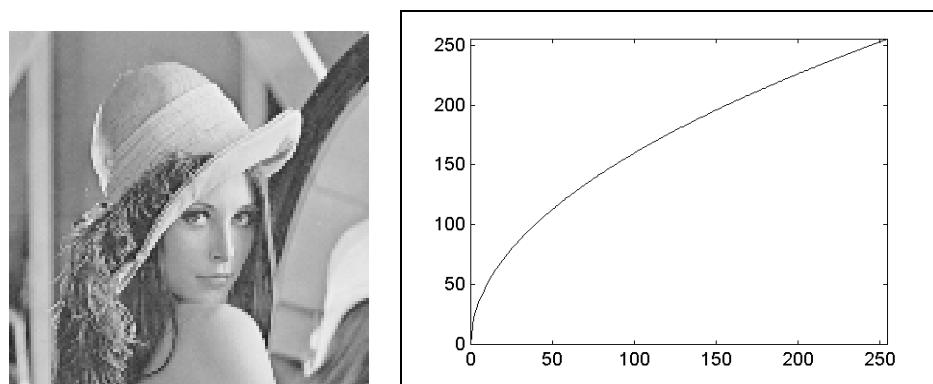
Rys. 3.15. Wynik przekształcenia sztucznego obrazu za pomocą funkcji  $L'(m,n) = (L(m,n))^2$ . Po lewej obraz oryginalny, po prawej przekształcony za pomocą odpowiednio wyskalowanej funkcji kwadratowej.



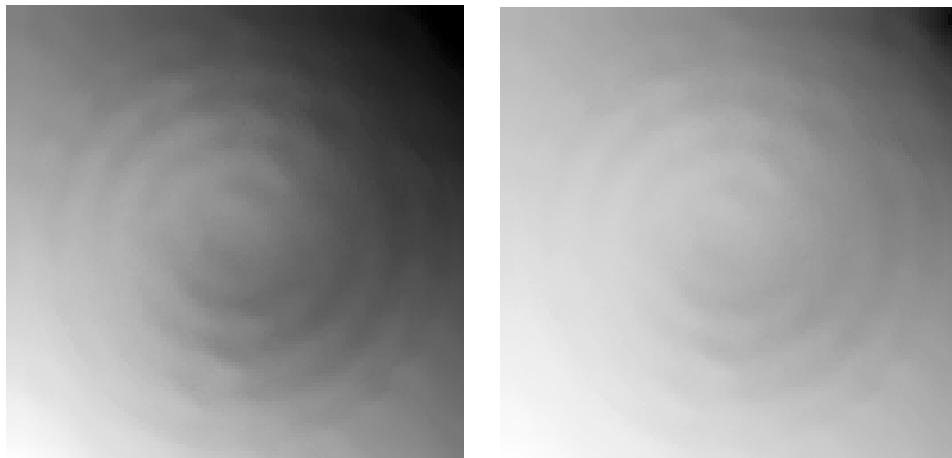
Rys. 3.16. Wynik przekształcenia obrazu „Lena” za pomocą funkcji  $L'(m,n) = (L(m,n))^3$  (po lewej) i wykres odpowiednio wyskalowanej funkcji sześciennnej (po prawej).



Rys. 3.17. Wynik przekształcenia sztucznego obrazu za pomocą funkcji  $L'(m,n) = (L(m,n))^3$ . Po lewej obraz oryginalny, po prawej - przekształcony za pomocą odpowiednio wyskalowanej funkcji sześciennnej.



Rys. 3.18. Wynik przekształcenia obrazu „Lena” za pomocą funkcji **pierwiastkowej** (po lewej) i wykres odpowiednio wyskalowanej funkcji **pierwiastkowej** (po prawej).



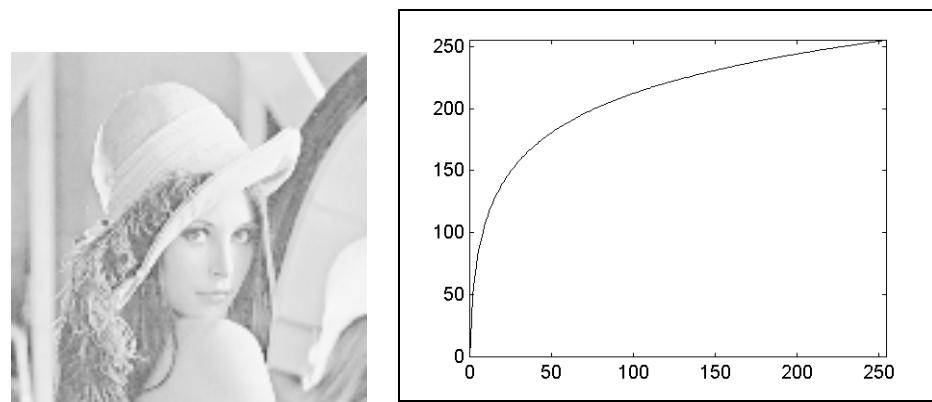
Rys. 3.19. Wynik przekształcenia sztucznego obrazu za pomocą funkcji *pierwiastkowej*.

Po lewej obraz oryginalny, po prawej - przekształcony za pomocą odpowiednio wyskalowanej funkcji.

- zastosowanie funkcji pierwiastkowej w celu podwyższenia kontrastu w obszarze małych wartości  $L(m,n)$  (oznacza to generalne rozjaśnienie obrazu z silnym różnicowaniem najjaśniejszych partii; patrz rysunki 3.18 i 3.19)

$$L'(m, n) = \sqrt{L(m, n)}$$

Funkcja pierwiastkowa musi być odpowiednio wyskalowana.

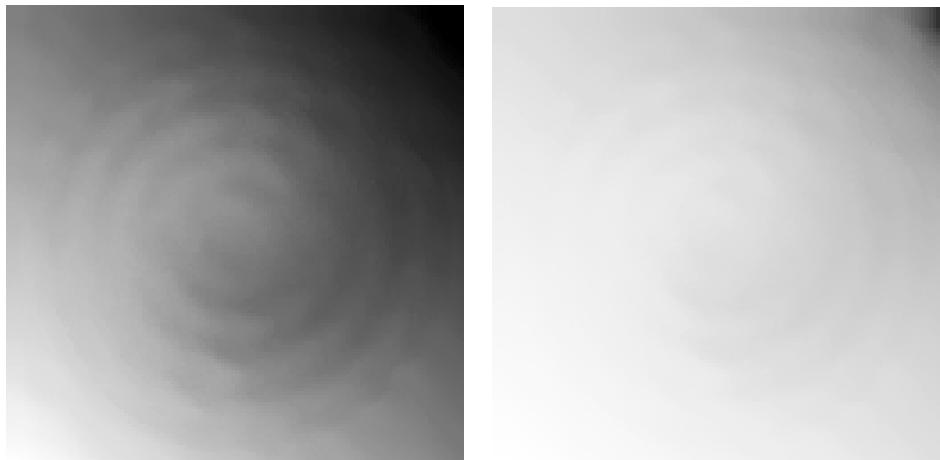


Rys. 3.20. Wynik przekształcenia obrazu „Lena” za pomocą funkcji *logarytmicznej* (po lewej) i wykres odpowiednio wyskalowanej funkcji *logarytmicznej* (po prawej).

- zastosowanie funkcji logarytmicznej w celu podwyższenia kontrastu w obszarze małych wartości  $L(m,n)$  (oznacza to generalne silne rozjaśnienie obrazu z bardzo silnym różnicowaniem najjaśniejszych partii; patrz rysunki 3.20 i 3.21)

$$L'(m,n) = \log(L(m,n) + 1)$$

Funkcja logarytmiczna musi być odpowiednio wyskalowana.



Rys. 3.21. Wynik przekształcenia sztucznego obrazu za pomocą funkcji **logarytmicznej**.

Po lewej obraz oryginalny, po prawej - przekształcony za pomocą odpowiednio wyskalowanej funkcji.

### 3.3.3 Realizacja przekształceń punktowych z użyciem LUT

Na pozór może się wydawać, że wykonanie operacji - na przykład - logarytmowania wszystkich punktów obrazu może być bardzo pracochłonne. Tymczasem w rzeczywistości wszystkie wymienione czynności mogą być wykonane bardzo szybko, dzięki powszechnemu stosowaniu w przekształceniach punktowych obrazów operacji typu LUT<sup>18</sup>. W operacji tej do przekształcania wartości poszczególnych punktów obrazu używa się przygotowanych a priori tabel przekodowania, inaczej zwanych tablicami korekcji. Możliwość przygotowania tabeli przekodowania wynika z faktu, że przy ograniczonej i dyskretnej skali jasności obrazu dla każdego piksela obrazu źródłowego ulokowanego w punkcie  $(m, n)$  ( $m \in [0, M-1]$ ,  $n \in [0, N-1]$ ) zachodzi warunek

$$L(m,n) \in \mathcal{N}$$

gdzie  $\mathcal{N}$  oznacza zbiór liczb całkowitych z przedziału  $[0, 2^B - 1]$ , a  $B$  jest przyjętą liczbą bitów dla reprezentacji jednego punktu obrazu. Zbiór  $\mathcal{N}$  zawiera skończoną i na ogół niewielką liczbę wartości, można więc dla każdej z tych wartości  $x \in \mathcal{N}$  z góry obliczyć wartość funkcji  $\Psi(x)$  a następnie zbudować tabelę (patrz Tabela 3.1), w której zestawione będą wartości funkcji  $\Psi(x)$  dla wszystkich wartości  $x$  należących do przedziału  $[0, 2^B - 1]$ .

---

<sup>18</sup> ang. *Look Up Tables*.

Tabela 3.1. Struktura tablicy przekodowania obrazu, wykorzystywanej w operacji LUT

stara wartość	nowa wartość
00000000	X1
00000001	X2
.....	...
11111111	X256

Przy posługiwaniu się tabelą przekodowań wykonywanie operacji punktowych (zwanych też operacjami anamorficznymi) jest bardzo proste i szybkie. Wartość piksela obrazu źródłowego  $L(m,n)$  staje się adresem (numerem wiersza w tabeli), zaś wartość odpowiedniego piksela dla obrazu wynikowego  $L'(m,n)$  jest po prostu odczytywana z tabeli i może być natychmiast wykorzystana - niezależnie od stopnia złożoności funkcji  $\Psi(x)$ . Dlatego można sobie pozwalać przy budowie odwzorowań punktowych nawet na bardzo złożone przekształcenia  $\Psi(x)$ , mając świadomość, że podczas efektywnego wykonywania na obrazie ich realna złożoność obliczeniowa jest bardzo mała.

### 3.3.4 Wyrównywanie histogramu

Często stosowanym przekształceniem jednopunktowym jest **wyrównanie histogramu**. Polega ono na takim przekształceniu jasności poszczególnych punktów obrazu, aby ilość punktów o jasności leżącej w każdym z równych przedziałów histogramu była (w przybliżeniu) taka sama. Transformacja ta zostanie niżej dokładniej omówiona, ponieważ ma duże znaczenie praktyczne. Zanim to jednak nastąpi - konieczne jest wyjaśnienie, czym jest histogram obrazu.

Histogram obrazu jest bardzo prostą i bardzo użyteczną funkcją, pozwalającą w sposób globalny scharakteryzować obraz. Formalnie można określić histogram jako funkcję  $h(i)$  zdefiniowaną w następujący sposób:

$$h(i) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} p(i / (m, n)) \quad i = 0, 1, \dots, 2^B - 1$$

gdzie

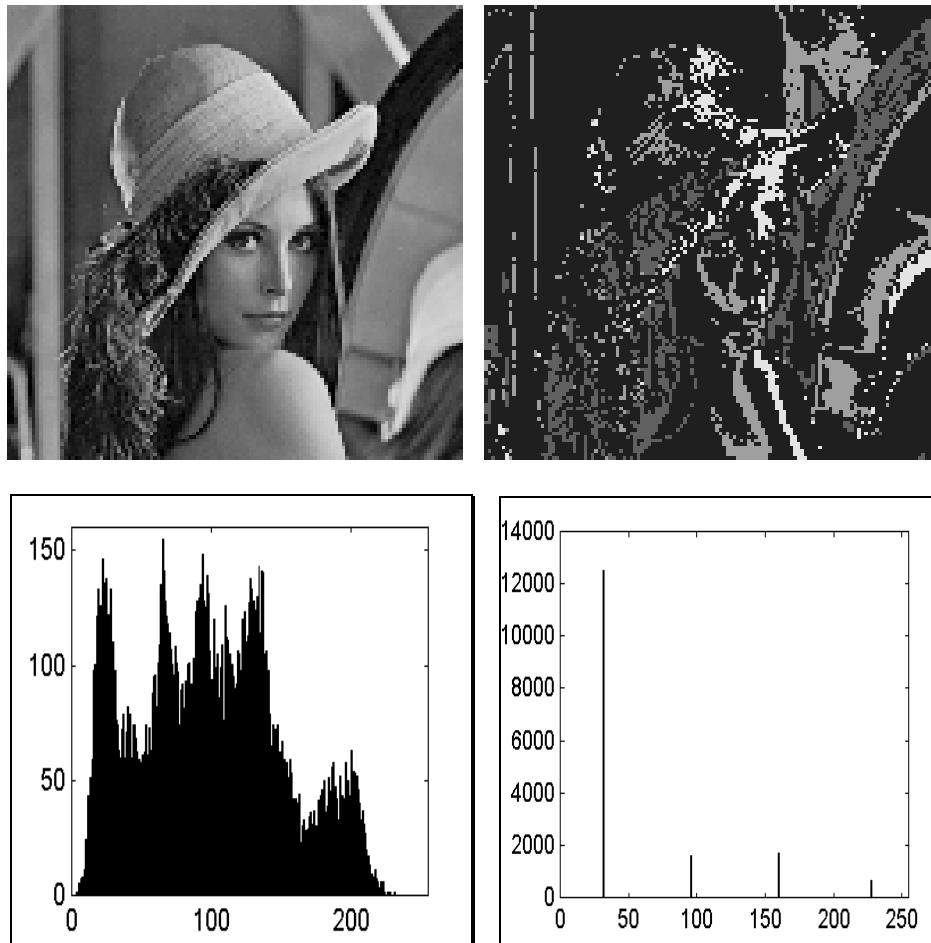
$$p(i / (m, n)) = \begin{cases} 1 & \text{gdy } L(m, n) = i \\ 0 & \text{w przeciwnym przypadku} \end{cases}$$

W praktyce oznacza to, że poszczególne składowe funkcji  $h(i)$  oznaczają po prostu liczbę pikseli o jasności  $i$  występujących na obrazie. Można to przedstawić w postaci tabeli (patrz Tabela 3.2) lub w postaci wykresu, na którego poziomej osi odkładane są kolejne możliwe wartości stopni szarości  $i = 0, 1, \dots, 2^B - 1$ , a wartości funkcji  $h(i)$  pokazuje się w postaci słupków ustawionych w punktach

odpowiedających kolejnym wartościom  $i$  (patrz rysunek 3.22). Wyskalowanie osi pionowej takiego wykresu pozwala na ustalenie, jaka jest liczba punktów o określonej szarości, co umożliwia dokonanie pewnych ilościowych oszacowań dotyczących struktury i zawartości obrazu.

Tabela 3.2. Przedstawienie histogramu w postaci tabelarycznej

poziomy szarości	ilość pikseli
00000000	K1
00000001	K2
.....	...
11111111	K256



Rys. 3.22. Przykładowe obrazy (u góry) i ich histogramy u dołu.

Analizując histogram można uzyskać wiele użytecznych informacji na temat rozważanego obrazu. Na przykład można zauważyc, że wartość  $h(i)$  może być zerowa dla wielu wartości  $i$ , co oznacza, że dostępne poziomy kwantowania wykorzystane są nieefektywnie. Może to występować na obrazach, w których pewne poziomy szarości usunięto (na przykład przypomniany na rysunku 3.14 obraz „Lena”, na którym zachowano tylko niektóre „poziomice” szarości) albo może to wystąpić w sytuacji, kiedy dynamika obrazu jest mała i zakres dozwolonych szarości nie jest poprawnie wykorzystany. Na ogół, jeżeli brakuje niektórych poziomów szarości w histogramie, to przyczyną są niedoskonałości przetwornika analogowo - cyfrowego i jest to zwykle sytuacja niekorzystna.

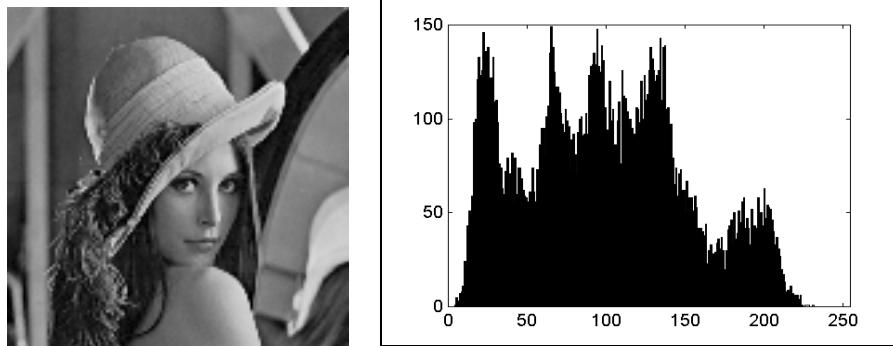
Takiemu mankamentowi obrazu można zaradzić dokonując operacji równoważenia histogramu. Operacja ta polega na zmianie położenia (wzdłuż poziomej osi, odpowiadającej stopniom szarości poszczególnych pikseli) kolejnych słupków, zawierających zliczenia liczby pikseli o danej szarości. Kryterium, jakie jest stosowane przy tym przesuwaniu jest następujące. Jeśli założymy, że dla pewnych liczb całkowitych  $m$  i  $n$  należących do dziedziny funkcji  $h(i)$  spełniony jest warunek  $h(m) > 0$  i  $h(n) > 0$  i równocześnie  $h(i) = 0$  dla wszystkich  $m < i < n$ , to wówczas należy tak przemieszczać punkty  $m$  i  $n$ , by minimalizować wartość  $Q$  wyznaczaną ze wzoru:

$$Q = \left| \frac{\sum_{i=0}^{2^B-1} h(i)}{2^B - 1} - \frac{h(m)}{n - m} \right|$$

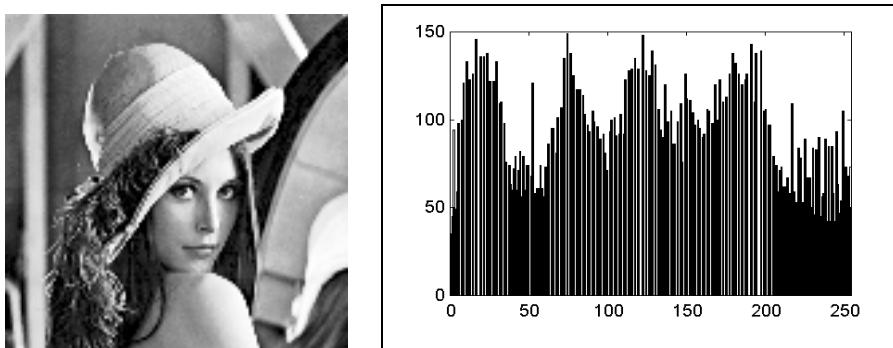
Oczywiście przemieszczanie punktów  $m$  i  $n$  polega w istocie na tym, że zmienia się (za pomocą odpowiednich operacji typu LUT) stopnie szarości określonych punktów na obrazie. Intuicyjnie można powiedzieć, że przy równoważeniu histogramu należy zwiększyć różnicę jasności pomiędzy tymi pikselami na obrazie, które mają jasności często występujące. Wszystko to brzmi w sposób dosyć skomplikowany, ale w istocie programy dokonujące równoważenia histogramu działają w sposób dosyć prosty, zaś efekt ich działania można obejrzeć na rysunku 3.23 i 3.24. Rysunek 3.23 przedstawia oryginalny obraz „Lena” i jego histogram.

Natomiast na rysunku 3.24 widoczny jest obraz po operacji równoważenia histogramu. Porównując rysunki 3.23 i 3.24 łatwo można zauważyc, że po wyrownaniu histogramu wysokie prążki, symbolizujące na histogramie punkty obrazu o takich szarościach, które często występują na obrazie - zostały odsunięte od siebie. W wyniku takiego zabiegu są one wyraźniej odróżnialne na obrazie, co poprawia jego czytelność. Z kolei te poziomy szarości, które na obrazie rzadko występują (niskie prążki histogramu) - zostały ze sobą utożsamione. Oznacza to,

że została utracona część informacji, jednak obraz generalnie stał się po tym zbiegu bardziej wyrazisty i czytelny.



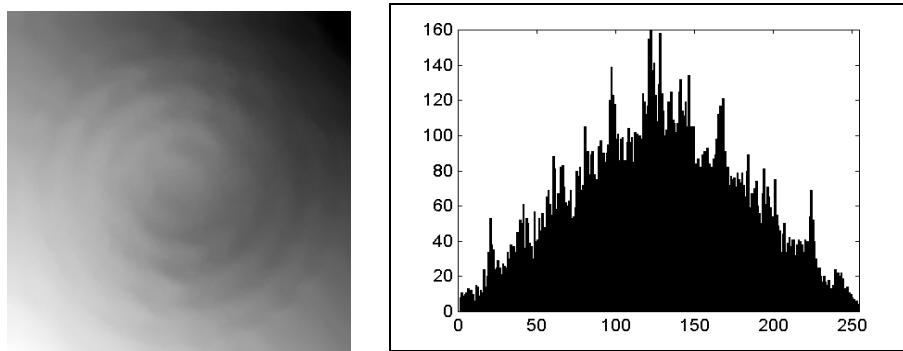
Rys. 3.23. Oryginalny obraz „Lena” i jego histogram.



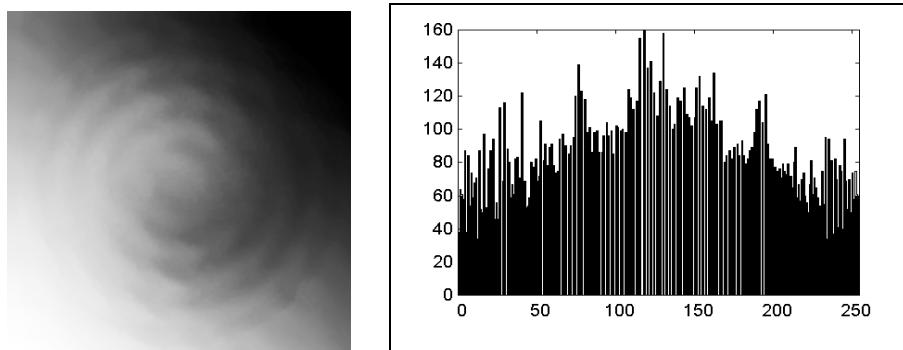
Rys. 3.24. Obraz „Lena” po wyrównaniu histogramu i jego histogram.

Oryginalny obraz „Lena” jest obrazem o bardzo dobrej jakości, dlatego zresztą jest on tak chętnie i tak często stosowany do prezentacji różnych aspektów przetwarzania obrazów - zarówno w tej książce, jak i w całej dość bogatej literaturze światowej, poświęconej tej problematyce. Ta dobra jakość obrazu źródłowego powoduje jednak, że wyrównywanie histogramu wnosi przy obrazie „Lena” mało zauważalną poprawę jego jakości. Bywają jednak obrazy o kiepskiej jakości, w których proces wprowadzania obrazu do komputera doprowadził do istotnego zubożenia jego dynamiki. Przy takich obrazach równoważenie histogramu może poprawić jakość obrazu w sposób naprawdę radykalny. Rozważmy najpierw sztuczny obraz pokazany na rysunku 3.25 i jego histogram. Histogram tego obrazu przypomina swoim kształtem trójkąt (co oznacza, że w obrazie zdecydowanie dominują średnie odcienie szarości). Tymczasem podany wyżej warunek minimalizacji wskaźnika  $Q$ , będący istotą procesu równoważenia histogramu, zakłada, że idealny histogram powinien mieć w ogólnym zarysie kształt zbliżony

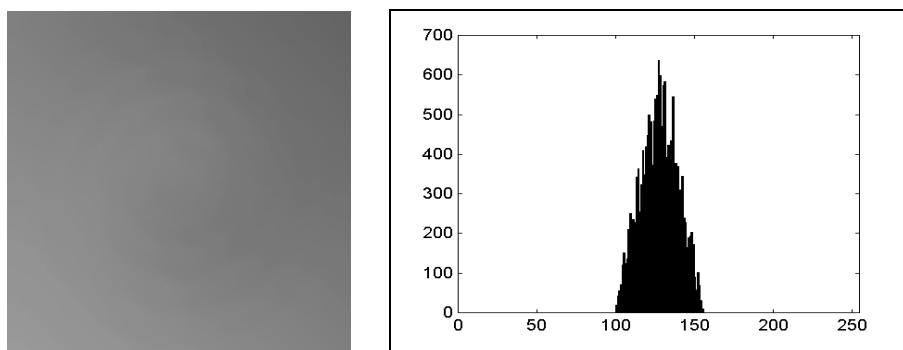
do prostokąta (żadne wydzielone obszary w dziedzinie szarości nie powinny mieć przewagi nad innymi). Oglądając rysunek 3.26 można stwierdzić, na ile ten ideał udało się zrealizować - i jakie to miało skutki w odniesieniu do czytelności samego obrazu.



3.25. Sztuczny obraz i jego histogram.



Rys. 3.26. Sztuczny obraz po wyrównaniu histogramu i jego histogram.

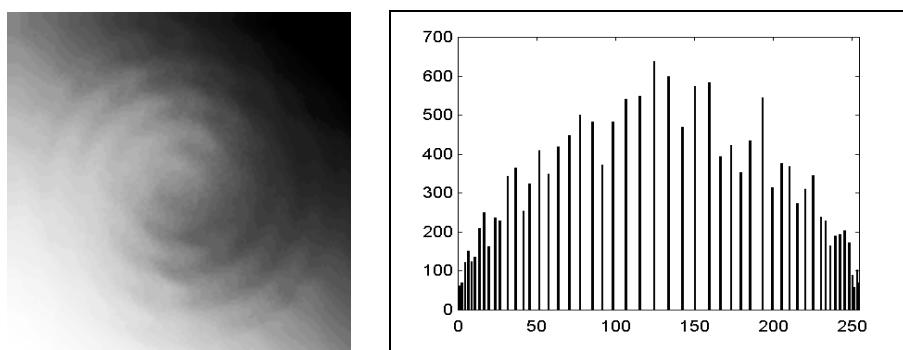


3.27. Sztuczny obraz o wadliwej skali szarości i jego histogram.

Mimo niewłaściwego kształtu histogramu obraz na rysunku 3.25 też miał niezłą jakość, więc jego wygląd po zrównoważeniu histogramu nie jest istotnie różny

od obrazu, jaki mamy przed tym zabiegiem. Obejrzyjmy jednak ten sam obraz na rysunku 3.27. Tutaj (celowo) popełniono najgorszy możliwy błąd - dopuszczono do drastycznego zawężenia zakresu dynamiki obrazu podczas jego wprowadzania. Jak widać na rysunku 3.27 - w efekcie obrazu prawie nie widać - wszystko jest równomiernie szare. Istotę zaistniałego nieszczęścia wyjaśnia rzut oka na histogram - piksele obrazu przyjmują wartości pewnego wąskiego zakresu i cały przedział możliwych szarości - od intensywnej czerni do absolutnej bieli, wykorzystany jest tylko w niewielkim stopniu.

Jeśli teraz, dla tego fatalnego obrazu zastosuje się operację wyrównywania histogramu - efekt będzie zauważalny (patrz rys. 3.28).



Rys. 3.28. Sztuczny obraz z wadliwą skalą szarości po wyrównaniu histogramu.

Tymczasem warto zauważyć, że obraz na rysunku 3.28 zawiera w istocie **mniej** informacji, niż obraz na rysunku 3.27, ponieważ podczas równoważenia histogramu doszło do utożsamienia stopnia szarości pewnej grupy rzadko występujących, szczególnie jasnych i szczególnie ciemnych, pikseli. Liczba rozróżnianych na obrazie poziomów szarości w wyniku tego **zmalała**, a nie wzrosła. Jednak subiektywnie czytelność obrazu radykalnie się poprawiła, bo statystycznie częściej pojawiał się na obrazie znaczący kontrast między sąsiednimi punktami.

Wyrównywanie histogramu można wykonywać globalnie (jak to pokazano wyżej) lub lokalnie. Lokalne wyrównanie histogramu polega na przeprowadzeniu takiej operacji w każdym z fragmentów obrazu niezależnie. Operacja taka niweluje, na przykład, skutki nierównomierności oświetlenia obiektu. Zagadnienie to nie będzie tu szczegółowo ilustrowane.

### 3.3.5 Punktowe operacje wykonywane na dwu obrazach

Operacjom punktowym mogą podlegać także dwa obrazy  $L_1(m,n)$  i  $L_2(m,n)$ , dając w rezultacie trzeci obraz (wynikowy)  $L'(m,n)$ . Przyjmując, że określona jest pewna dwuargumentowa (skalararna) funkcja  $\Phi$  możemy w ogólny sposób zapisać te przekształcenia w postaci:

$$L'(m,n) = \Phi(L_1(m,n), L_2(m,n))$$

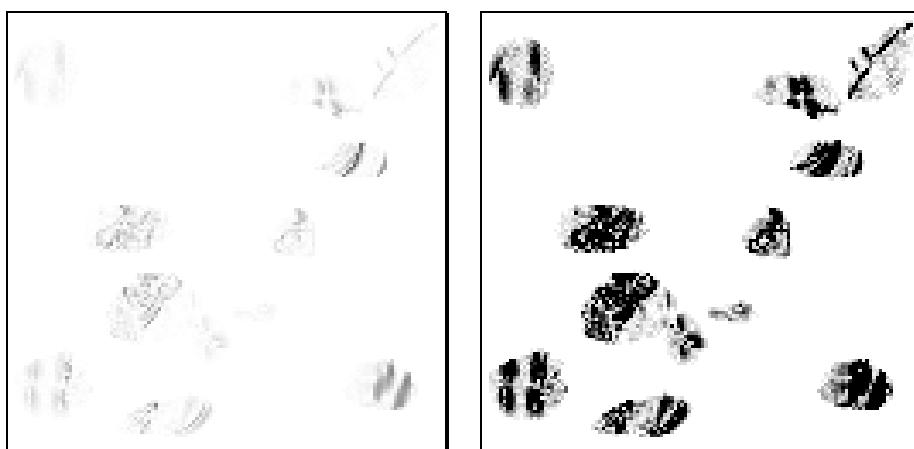
Do podstawowych dwuargumentowych operacji punktowych należą:

- dodanie dwóch obrazów,
- odjęcie dwóch obrazów,
- przemnożenie dwóch obrazów,
- kombinacja liniowa dwóch obrazów.

Przekształcenie arytmetyczne dwuargumentowe polega na przeprowadzeniu odpowiedniej operacji arytmetycznej na odpowiadających sobie punktach obrazów wyjściowych i zapisanie wynikowego elementu do obrazu końcowego (finalnego). Dodawanie dwóch obrazów wykonuje się głównie w celach trickowych („cyfrowe fotomontaże”).



Rys. 3.29. Obraz „Lena” - oryginalny i z celowo wprowadzonymi 12 zmianami. Proszę je wskazać!



Rys. 3.30. Obraz, będący różnicą między dwoma obrazami pokazanymi na rysunku 3.29.

Po lewej stronie obraz różnicowy w oryginalnej skali szarości, po prawej - przeskalowany do pełnego zakresu 0-255.

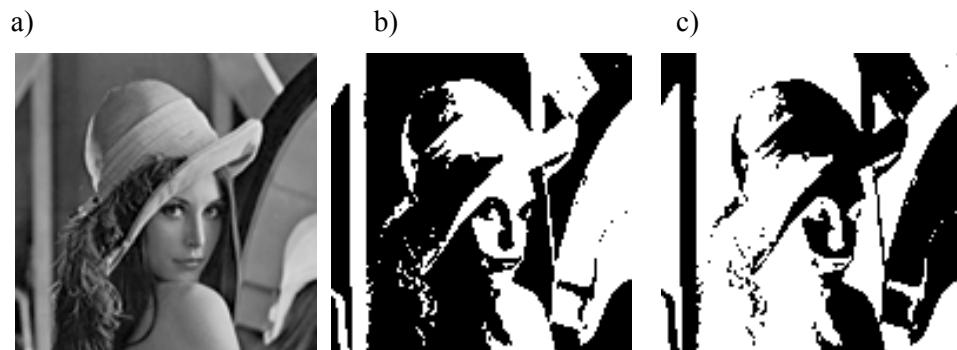
Zagadnieniu temu nie będziemy poświęcali zbyt wiele uwagi. Natomiast interesujące możliwości, szczególnie ważne z punktu widzenia zastosowań praktycznych, tkwią w operacji odejmowania obrazów. Odjęcie dwóch obrazów jest podstawowym sposobem wykrycia zmian na obrazach (na przykład ruchomych obiektów w systemach automatycznego zabezpieczenia określonych obiektów). Popatrzmy na rysunek 3.29. Pokazano na nim obraz „Lena” i ten sam obraz z pewnymi (celowo wprowadzonymi) zmianami. Przy odrobinie wysiłku można te zmiany zauważać gołym okiem, ale jeśli się te dwa obrazy odejmie od siebie (obraz różnicowy pokazano na rysunku 3.30) - zmiany staną się bardzo widoczne i natychmiast dostrzegalne.

O tym, jak subtelna jest metoda odejmowania obrazów, świadczyć może przykład pokazany na rysunku 3.31. Na rysunku tym pokazano fotomontaż obrazu „Lena” i napisu *NAPIS*. Napis jest w praktyce niewidoczny (bardzo dokładnie wpatrując się w rysunek można zauważać zarys litery *N* na denku kapelusza dziewczyny - reszta jest absolutnie niezauważalna). Tymczasem odjęcie obrazu spreparowanego od oryginalnego obrazu „Lena” natychmiast ujawnia fotomontaż.



Rys. 3.31. Obraz „Lena” zmontowany z obrazem napisu (po lewej) i sam napis ujawniony na obrazie różnicowym.

Operacja obliczania różnicy dwóch obrazów może - wraz z towarzyszącą jej normalizacją pikseli obrazu różnicowego do pełnego zakresu dopuszczalnych stopni szarości - służyć do wydobywania i podkreślania bardzo subtelnych cech obrazu. Na przykład, jeśli z oryginalnego obrazu „Lena” wydobędzie się obraz binarny, pokazany na rysunku 3.32 po lewej stronie, oraz jego negatyw (pokazany na tym samym rysunku po prawej), wówczas poprzez odejmowanie tych „masek” od oryginalnego obrazu i odpowiednie skalowanie można uzyskać efekty, które mają nawet pewien interesujący walor estetyczny (patrz rysunek 3.33).

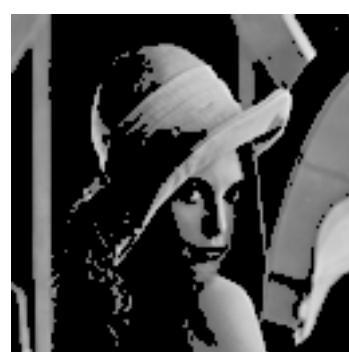


Rys. 3.32. Obraz „Lena” i „maski” wydobyte z tego obrazu za pomocą binaryzacji.



Rys. 3.33. Obrazy uzyskane po odjęciu obrazu „Lena” i masek pokazanych na rysunku 3.32.

Zdecydowanie interesujące wyniki uzyskuje się podczas mnożenia przez siebie dwóch odpowiednio dobranych obrazów. Na przykład na rysunku 3.34 można prześledzić efekt mnożenia obrazu „Lena” przez pierwszą z masek pokazanych na rysunku 3.32.



Rys. 3.34. Obraz „Lena” po przemnożeniu przez „maskę”.

Niesłychanie ciekawe możliwości wiążą się z tworzeniem kombinacji liniowych dwóch lub większej liczby obrazów. Na rysunku 3.35 pokazano efekty wyliczenia

kombinacji liniowych obrazu „Lena” i obrazu papugi (znanego w literaturze pod nazwą „Hamlet” - pewnie tak się ta papuga nazywa). Kolejno pokazano kombinacje zawierające odpowiednio 0%, 25%, 50%, 75% i 100% papugi.



Rys. 3.35. Kombinacje liniowe obrazu „Lena” i obrazu „Hamlet”.

### 3.3.6 Binaryzacja

Jedną z ważniejszych czynności punktowego przetwarzania obrazu jest binaryzacja, występująca prawie zawsze jako czynność poprzedzająca analizę obrazu a także bardzo przydatna w procesie rozpoznawania. Celem binaryzacji jest radykalna redukcja ilości informacji zawartej w obrazie. Przeprowadzenie procesu binaryzacji polega na tym, aby obraz mający wiele poziomów szarości zamienić na obraz, którego piksele mają wyłącznie wartość 0 i 1. Przykłady obrazów binarnych tworzonych na podstawie obrazu o wielu poziomach szarości były już wcześniej podawane. Teraz zajmiemy się jednak nieco dokładniej samą techniką binaryzacji.

Binaryzacja może zostać przeprowadzona na wiele sposobów. Poniżej przedstawiono najczęściej wykorzystywane metody:

- **binaryzacja z dolnym progiem:**

$$L'(m,n) = \begin{cases} 0; & L(m,n) \leq a \\ 1; & L(m,n) > a \end{cases}$$

gdzie:

$L(m,n)$  - jasność punktu w obrazie źródłowym  $L(m,n) \in [0, 2^B-1]$ ;

$L'(m,n)$  - wartość odpowiedniego punktu w obrazie wynikowym

$L'(m,n) \in \{0,1\}$ ,

$a$  - próg binaryzacji.

- **binaryzacja z górnym progiem:**

$$L'(m,n) = \begin{cases} 0; & L(m,n) \geq a \\ 1; & L(m,n) < a \end{cases}$$

- **binaryzacja z podwójnym ograniczeniem:**

$$L'(m,n) = \begin{cases} 0; & L(m,n) \leq a_1 \\ 1; & a_1 < L(m,n) \leq a_2 \\ 0; & L(m,n) > a_2 \end{cases}$$

gdzie dodatkowo:

$a_1, a_2$  - progi binaryzacji,  $a_1 < a_2$  ;

- **binaryzacja warunkowa** (binaryzacja z histerezą):

$$L'(m,n) = \begin{cases} 0; & L(m,n) \leq a_1 \\ s; & a_1 < L(m,n) \leq a_2 \\ 1; & L(m,n) > a_2 \end{cases}$$

gdzie dodatkowo:

$s$  - wartość sąsiadujących punktów,  $s \in \{0,1\}$ ;

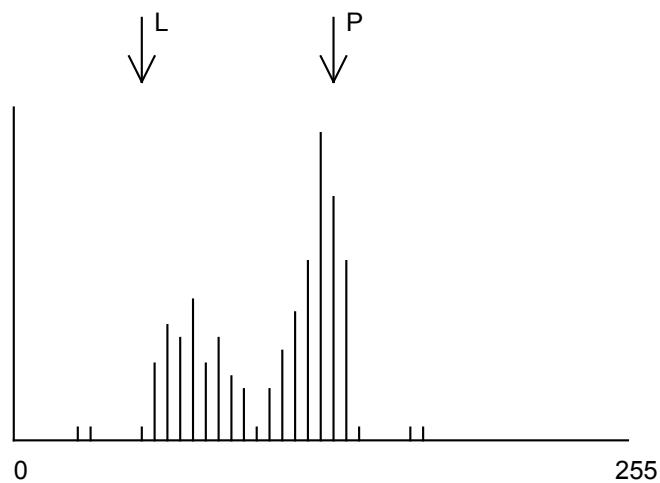
- **binaryzacja wielokryterialna**<sup>19</sup> - binaryzacja przeprowadzana jest niezależnie na wielu obszarach obrazu, znacznie różniących się poziomem jasności. Operacja ta nie będzie tu opisywana.

Przy wykonywaniu binaryzacji obrazu podstawowym problemem jest odpowiedni wybór progu binaryzacji, oznaczanego w podanych wyżej wzorach przez  $a$ . Najczęściej w celu znalezienia właściwej wartości progu  $a$  tworzy się histogram obrazu. W najprostszym (ale na szczęście dość często spotykanym) przypadku histogram taki ma postać dwugarbnej krzywej (rys. 3.36). Taki kształt histogramu oznacza, że na obrazie dają się wyróżnić dwa rodzaje stosunkowo często występujących punktów: ciemne, których szarości grupują się w postaci skupiska oznaczonego na histogramie literą L, oraz jasne, których szarości tworzą skupisko oznaczone na histogramie literą P. Zależnie od okoliczności mogą nas

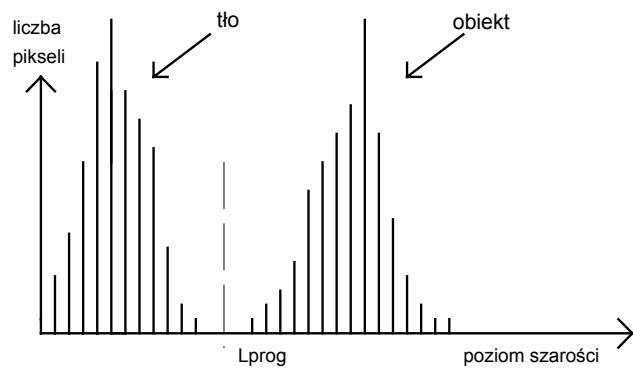
---

<sup>19</sup> ang. *multithresholding*.

interesować albo te pierwsze, albo drugie (jasne obiekty na ciemnym tle, albo ciemne obiekty na jasnym tle). Do ich wydzielenia możemy zastosować filtrację z górnym progiem lub filtrację z dolnym progiem. Niezależnie jednak od tego, która z wymienionych ewentualności istotnie ma miejsce - wartość progu  $a$  należy tak ustalić, by przypadała ona na środek „doliny” między pokazanymi skupiskami (rys. 3.37).



Res. 3.36. Przykładowy histogram obrazu ujawnia, że na obrazie występują piksele należące do dwóch klas.



Rys. 3.37. Położenie progu binaryzacji można ustalić na podstawie lokalizacji „doliny” na histogramie.

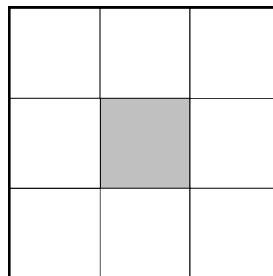
W zasadzie binaryzacja jest jednym z prostszych i mniej skomplikowanych przekształceń punktowych, można było więc skwitować ją krótko przy omawianiu operacji LUT (tą techniką binaryzację często się wykonuje) i nie zajmować się nią więcej. Binaryzacja odgrywa jednak tak ważną rolę w analizie obrazu, że poświęcono jej tu cały oddzielny rozdział. Dopiero na obrazach binarnych (i za-

zwyczaj tylko na nich) można przeprowadzić większość pomiarów oraz niektóre złożone przekształcenia. Należy o tym pamiętać.

## 3.4 Kontekstowa filtracja obrazu

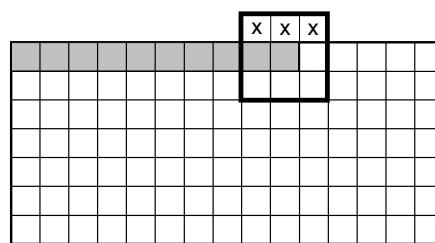
### 3.4.1 Ogólne problemy filtracji cyfrowej

Filtry cyfrowe stanowią o wiele bardziej złożone narzędzie przetwarzania obrazów, niż omawiane wyżej przekształcenia jednopunktowe. Z reguły (i w zasadzie jest to reguła pozbawiona wyjątków) filtry używane do analizy obrazów zakładają, że wykonywane na obrazie operacje będą **kontekstowe**. Oznacza to, że dla wyznaczenia wartości **jednego** punktu obrazu wynikowego trzeba dokonać określonych obliczeń na **wielu** punktach obrazu źródłowego. Zwykle polega to na wyznaczeniu wartości funkcji, której argumentami są wartości piksela o tym samym położeniu na obrazie źródłowym oraz wartości pikseli z jego otoczenia  $K$ , które w ogólnym przypadku może mieć różną formę, ale najczęściej utożsamiane jest z kwadratowym „oknem” otaczającym symetrycznie aktualnie przetwarzany punkt obrazu (rys. 3.38).



Rys. 3.38. Struktura „kontekstu”  $K$  wykorzystywane typowo podczas filtracji obrazów.

Z powodu kontekstowości wykonywanych operacji filtracja z reguły nie może dotyczyć pikseli znajdujących się bezpośrednio na brzegu obrazu (rys. 3.39), ponieważ dla odpowiedniej funkcji wieloargumentowej, będącej matematycznym zapisem reguły działania filtru, brakować będzie wartości argumentów oznaczonych symbolem  $x$ .



Rys. 3.39. Niemożność wykonania kontekstowych filtracji dla punktów położonych przy brzegu obrazu.

Mimo wskazanego ograniczenia, kontekstowe operacje filtracji obrazu są bardzo często wykorzystywane w przetwarzaniu i analizie obrazu. Operacje te - w odróżnieniu od operacji punktowych - istotnie zmieniają zawartość obrazu, w tym także geometrię widocznych na obrazie obiektów. Pozwalają one dzięki temu w efektywny sposób pozbyć się z obrazu pewnych niepożądanych obiektów (zakłóceń, szumów) lub wydobyć z niego pewne, nieuchwytnie na pierwszy rzut oka, użyteczne informacje. W praktycznych zastosowaniach filtry wykorzystywane są zazwyczaj do realizacji następujących celów:

- Stłumienie w obrazie niepożądanego szumu. Przy braku konkretnych przesłańek na temat istoty szumu realizujący tę funkcję filtr działa zazwyczaj na zasadzie lokalnych średnich. Każdemu z punktów obrazu przypisywana jest w takim przypadku średnia wartości jego otoczenia.
- Wzmocnienie w obrazie pewnych elementów zgodnych z posiadanym wzorcem. W tym przypadku dany punkt zostanie wzmocniony w stopniu zależnym od spełniania przez jego otoczenie określonych warunków.
- Usunięcie określonych wad z obrazu. Na przykład: usunięcie wad powstały wskutek zarysowania kliszy fotograficznej.
- Poprawa obrazu o złej jakości technicznej. Na przykład: obrazów nieostrych, poruszonych lub o niewielkim kontraście.
- Rekonstrukcja obrazu, który uległ częściowemu zniszczeniu. Na przykład: rekonstrukcja materiałów fotograficznych, które przez długi czas podlegały działaniom niekorzystnych warunków.

We wszystkich tych przypadkach bardzo istotne jest posiadanie wiedzy co do istoty cech obrazu, które zamierzamy wzmocnić lub stłumić. Bardzo użyteczna jest wiedza na temat sposobu powstania tych cech w analizowanym obrazie.

Z matematycznego punktu widzenia filtr jest pewną funkcją (wieloargumentową) przekształcającą jeden obraz w drugi metodą „piksel po pikselu”. Właściwości filtru wynikają wprost z analitycznych własności realizującej go funkcji. W dalszej części niniejszej pracy pojęcia filtru i realizującej go funkcji stosowane będą wymiennie. Będziemy więc wyróżniali filtry:

- liniowe (wykonujące operację filtracji w oparciu o pewną liniową kombinację wybranych pikseli obrazu wejściowego);
- nieliniowe (wykonujące operację filtracji w oparciu o pewną nielinową funkcję wybranych pikseli obrazu wejściowego).

Filtry liniowe są z reguły prostsze w wykonaniu, natomiast filtry nieliniowe mają z reguły bogatsze możliwości.

### 3.4.2 Filtry liniowe

#### 3.4.2.1 Podstawowe własności filtru liniowego

Filtры zaliczane są do liniowych, jeżeli funkcja je realizująca spełnia dwa warunki liniowości:

- jest addytywna

$$\varphi(f + g) = \varphi(f) + \varphi(g);$$

- oraz jednorodna

$$\varphi(\lambda f) = \lambda \varphi(f), \lambda \in \mathbb{R}$$

gdzie:

$\varphi$  - funkcja realizująca filtr;

$f, g$  - obrazy podlegające filtracji.

W praktyce najczęściej wykorzystywane są filtry spełniające dodatkowo warunek niezmienności względem przesunięcia:

$$\varphi(f_{\vec{h}}) = [\varphi(f)]_{\vec{h}}$$

gdzie dodatkowo:

$\vec{h}$  - wektor przesunięcia.

Spełnienie tego warunku oznacza, że każdy punkt obrazu przekształcany jest w identyczny sposób.

Filtры liniowe wykorzystywane są często, gdyż są bardzo proste w implementacji. Jednocześnie odpowiadają one intuicyjnym oczekiwaniom badaczy poszukujących odpowiednich przekształceń swych obrazów.

#### 3.4.2.2 Konwolucja

Przy rozpatrywaniu funkcji realizujących filtry liniowe wygodnie jest się posłużyć pojęciem *konwolucji*, zwanej także *splotem funkcji*. Konwolucja jest zdefiniowana następującym wzorem:

$$g(x) = (f \times h)(x) = \int_{-\infty}^{+\infty} f(x-t)h(t)dt$$

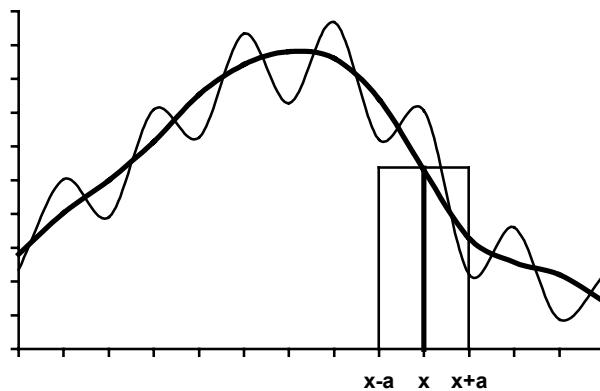
gdzie:

$f, h$  - splatane funkcje.

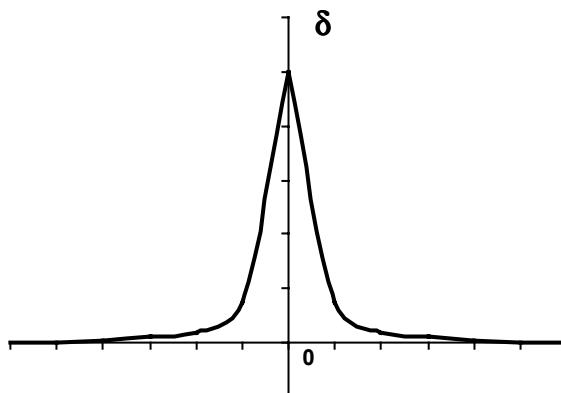
Splot ( $g$ ) jest zdefiniowany na całym  $\mathbb{R}$ , natomiast iloczyn  $f(x-t)h(t)$  jest całkowalny na całym  $\mathbb{R}$ , ale funkcja  $h$  może mieć skońzoną dziedzinę. W takim przypadku konwolucja wykorzystująca funkcję  $h$  staje się filtrem. Wspomnianą wyżej funkcję realizującą tłumienie szumów na zasadzie lokalnych średnich można realizować jako następującą konwolucję:

$$g(x) = \frac{1}{2a} \int_{-a}^a f(t)dt = (f \times h)(x), \quad \text{gdzie:} \quad h(u) = \begin{cases} \frac{1}{2a}; & u \in (-a, a) \\ 0; & u \notin (-a, a) \end{cases}$$

Rozwiązanie to posiada interpretację geometryczną pokazaną na rysunku 3.40:



Rys. 3.40. Interpretacja geometryczna zasady tłumienia szumów za pomocą lokalnych średnich.



Rys. 3.41. Przebieg przykładowej funkcji przybliżającej funkcję Diraca.

Przy rozważaniu właściwości filtrów liniowych istotne znaczenie praktyczne ma tak zwana odpowiedź impulsowa filtru, czyli reakcja filtru na impuls. Impuls taki może być wyrażony tzw. funkcją<sup>20</sup> Diraca (Rys. 3.41), która ma następujące własności:

---

<sup>20</sup> Dokładniej - jest to **dystrybucja** Diraca.

$$1. \delta_0 = 0; \quad x \neq 0$$

$$2. \int_{-\infty}^{\infty} \delta_0(x) dx = 1$$

Splatając funkcję Diraca  $\delta_0$  z inną funkcją  $h$  otrzymujemy następujący wynik:

$$(h \times \delta_0)(x) = \int_{-\infty}^{\infty} \delta_0(x-t)h(t)dt = h(x),$$

a zatem odpowiedzią filtra na impuls jest funkcja  $h$ , stanowiąca jego matrycę. Informacja ta jest szczególnie istotna w omawianym w następnym podrozdziale przypadku dyskretnym. Stanowi ona bowiem podstawę do projektowania matryc filtrów.

Konwolucja posiada pewne własności, które są bardzo pomocne w praktycznej realizacji filtrowania. Do własności tych należą, między innymi:

- **Lączność:**  $(f \times g) \times h = f \times (g \times h) = f \times g \times h$  Własność ta pozwala na rozdzielenie filtrowania dowolnie dużą matrycą na kolejne filtrowanie przy pomocy małych matryc.
- **Rozdzielność.** Pozwala to rozdzielić filtrację dwuwymiarowego obrazu jako złożenie filtracji jednowymiarowych.

### 3.4.2.3 Konwolucje dyskretne

W komputerowej analizie obrazu dziedzina funkcji  $L(m,n)$  jest dwuwymiarowa i dyskretna (nieciągła), gdyż taka jest reprezentacja obrazu. W takim przypadku wszystkie powyższe dywagacje znacznie się upraszczają. Konwolucję dla 2-wymiarowego, dyskretnego obrazu można zapisać w następujący sposób:

$$L'(m,n) = (w \times L)(m,n) = \sum_{i,j \in K} L(m-i,n-j)w(i,j)$$

Zgodnie z przyjętą wcześniej metodą reprezentacji obrazu  $L(m,n)$  jako zbioru punktów rozmieszczonych w węzłach regularnej sieci oraz ograniczeniem się do rozpatrywania otoczenia punktu o określonej średnicy filtry definiuje się jako tablice współczynników  $w(i,j)$ . Współczynniki  $w(i,j)$ , wraz z odpowiednimi elementami obrazu  $L(m-i,n-j)$ , znajdującymi się w „oknie”  $K$  rozlokowanym wokół punktu o współrzędnych  $(m,n)$  służą łącznie do obliczenia wartości funkcji  $L'(m,n)$  w danym punkcie na obrazie wynikowym. Głównym zagadnieniem wymagającym rozważenia jest problem normalizacji. Współczynniki  $w(i,j)$  wybiera się zwykle w taki sposób, by były liczbami całkowitymi. Powód takiego wyboru jest oczywisty - dla uzyskania wartości  $L'(m,n)$  dla wszystkich punktów obrazu trzeba w typowych warunkach wykonać ogromną liczbę dodawań i mnożeń (dla obrazu o typowych rozmiarach 512x512 punktów i otoczenia  $K$  obejmującego

zaledwie 9 punktów konieczne jest wykonanie aż 2.359.296 dodawań i mnożeń!). Przy takiej liczbie operacji nie jest rzeczą obojętną, czy będą to operacje wykonywane szybko i sprawnie na krótkich liczbach stałoprzecinkowych, czy też zaangażowawszy się w operacje zmiennoprzecinkowe wynikające z ułamkowych wartości  $w(i,j)$  będziemy wykonywać każde działanie kilkunastokrotnie dłużej. Jednak przy całkowitoliczbowych wartościach  $w(i,j)$  trzeba się liczyć z faktem, że po wykonaniu wymaganych działań uzyskane wartości wynikowych pikseli wyjściowego obrazu nie będą spełniały warunku normalizacji  $L'(m,n) \in [0, 2^B-1]$ . Dlatego pełna operacja filtracji konwolucyjnej musi obejmować czynność normalizacji. Dla filtrów eliminujących proste zakłócenia, dla których (co będzie pokazane niżej) wszystkie współczynniki spełniają warunek  $w(i,j) \geq 0$  możliwe jest zastosowanie stosunkowo prostej techniki normalizacji, danej wzorem:

$$L'(m,n) = \frac{1}{\sum_{(i,j) \in K} w(i,j)} \sum_{(i,j) \in K} L(m-i, n-j) w(i,j)$$

Gorzej jest w przypadku operatorów odwołujących się do współczynników  $w(i,j)$  przyjmujących zarówno wartości dodatnie, jak i ujemne. W takich przypadkach operacja normalizacji musi odwoływać się do rzeczywistych, uzyskanych po przetworzeniu obrazu wartości  $\min L'(m,n)$  oraz  $\max L'(m,n)$  i opierać się na wzorze:

$$L''(m,n) = \frac{L'(m,n) - \min L'(m,n)}{\max L'(m,n) - \min L'(m,n)} \cdot 2^B$$

Omawiając w dalszych podrozdziałach poszczególne typy i rodzaje filtrów konwolucyjnych będziemy na ogół abstrahowali od konieczności dokonywania normalizacji, zakładając, że stosowne przeskalowanie obrazu wynikowego może być łatwo wykonane przy użyciu operacji punktowych, które były już wyczerpująco omówione wcześniej.

Postać otoczenia  $K$  oraz liczba elementów wchodzących w jego skład są swobodnie wybierane przez projektantów filtrów konwolucyjnych. Ogólnie można powiedzieć, że im większy rozmiar otoczenia  $K$ , tym bardziej radykalne działanie filtra. Przykładowo, jeżeli założymy (co się najczęściej spotyka) rozpatrywanie otoczenia  $K$  punktu  $(m,n)$  w postaci kwadratowego okna o wielkości  $3 \times 3$ , to tablica współczynników  $w(i,j)$  przyjmie następującą postać:

$$\begin{bmatrix} w(1,1) & w(1,0) & w(1,-1) \\ w(0,1) & w(0,0) & w(0,-1) \\ w(-1,1) & w(-1,0) & w(-1,-1) \end{bmatrix}$$

co zazwyczaj zapisuje się też w uproszczonej postaci:

$$\begin{bmatrix} w_{11} & w_{10} & w_{1-1} \\ w_{01} & w_{00} & w_{0-1} \\ w_{-11} & w_{-10} & w_{-1-1} \end{bmatrix}$$

Filtr ma wtedy pewne wygodne właściwości (na przykład rozważany punkt  $L'(m,n)$  może być umieszczony dokładnie w środku stosowanego okna filtracji, co jest niemożliwe w przypadku filtrów o rozmiarach  $2 \times 2$ ), a jednocześnie „głębokość” filtracji nie jest tak uciążliwie duża jak w przypadku filtrów o większych rozmiarach okna.

Wymienione wyżej powody sprawiają, że w zdecydowanej większości przypadków stosuje się właśnie wariant kwadratowego okna o wymiarach  $3 \times 3$  i dlatego ten przypadek przeanalizujemy dokładniej. Używanie macierzy współczynników, w której indeksy przyjmują dwa i to zarówno dodatnie, jak i ujemne wartości - jest formalnie eleganckie, ale z praktycznego punktu widzenia raczej niewygodne. W związku z tym dla szybkiego i wygodnego opisywania kolejnych filtrów, w których wykorzystywać będziemy macierze konwolucji - wprowadzimy notację uproszczoną, wykorzystującą jedynie kolejne numery współczynników:

$$\begin{bmatrix} w_1 & w_2 & w_3 \\ w_4 & w_5 & w_6 \\ w_7 & w_8 & w_9 \end{bmatrix}$$

W tym przypadku funkcja realizująca proces filtracji (z użyciem konwolucji) może być zapisana w następujący sposób:

$$\begin{aligned} L'(m,n) = & w_1 L(m-1, n-1) + w_2 L(m-1, n) + w_3 L(m-1, n+1) \\ & + w_4 L(m, n-1) + w_5 L(m, n) + w_6 L(m, n+1) \\ & + w_7 L(m+1, n-1) + w_8 L(m+1, n) + w_9 L(m+1, n+1) \end{aligned}$$

Odpowiednio dobierając współczynniki  $w(i,j)$  będziemy teraz budowali i badali filtry o różnych właściwościach.

#### 3.4.2.4 Filtry dolnoprzepustowe

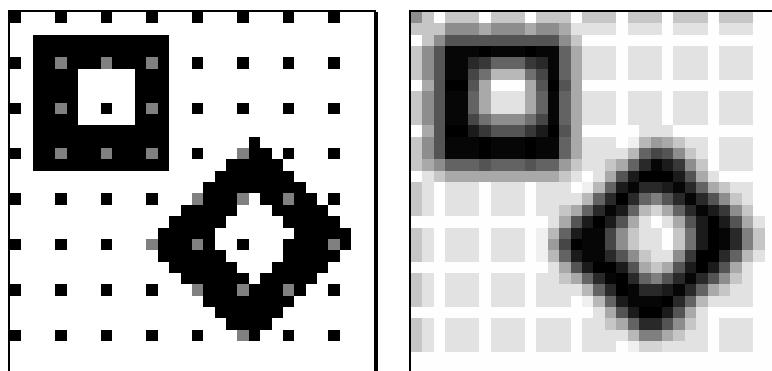
Najbardziej typowe zastosowanie filtracji polega oczywiście na usuwaniu zakłóceń z obrazu. Przy tego typu zastosowaniu korzystne jest używanie z prostego filtra uśredniającego, którego macierz konwolucji

$$\begin{bmatrix} w_1 & w_2 & w_3 \\ w_4 & w_5 & w_6 \\ w_7 & w_8 & w_9 \end{bmatrix}$$

ma postać:

1	1	1
1	1	1
1	1	1

Efekt działania takiego filtra można prześledzić na rysunku 3.42. Jak łatwo zauważyc, filtr uśredniający usuwa drobne zakłócenia z obrazu - znikają (a raczej ulegają znacznemu osłabieniu ale i pewnemu „rozmażaniu”) pojedyncze czarne punkty na jasnym tle lub jasne plamki na tle ciemnym, wygładzane są drobne „zawirowania” krawędzi obiektów, usuwane mogą być efekty falowania jasności zarówno w obszarze samych obiektów, jak i w obszarze tła, itp. Filtr omawianego typu ma jednak także zdecydowanie niekorzystne działanie, ponieważ powoduje pewne „rozmycie” konturów obiektów i pogorszenie rozpoznawalności ich kształtów.



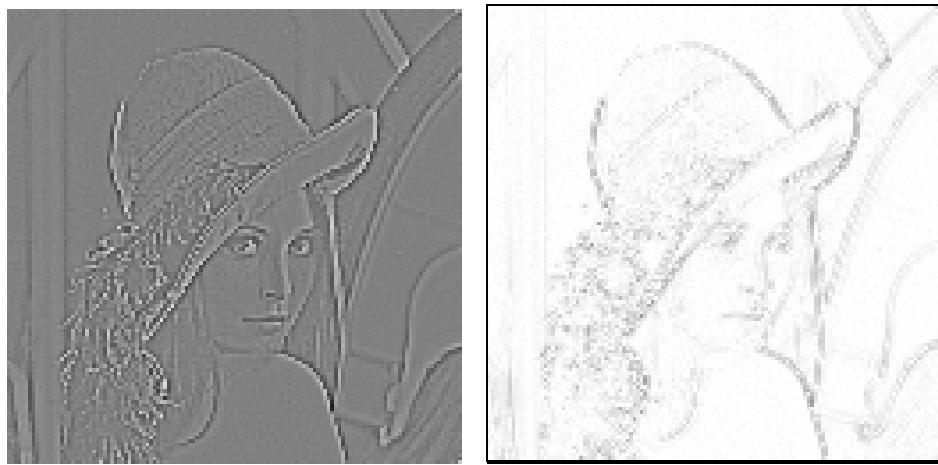
Rys. 3.42. Obraz z zakłóceniami (po lewej stronie) i efekt jego filtracji z pomocą filtra konwolucyjnego.



Rys. 3.43. Efekt filtracji konwolucyjnej dla niezakłóconego obrazu naturalnego.

Efekt działania filtra uśredniającego na bardziej realistycznym obrazie („Lena”) pokazuje rysunek 3.43. Tu także rzuca się w oczy efekt „erozji” obrazu powstający w wyniku zastosowania konwolucyjnej techniki filtracji.

Skutki dolnoprzepustowej filtracji konwolucyjnej stają się bardziej zauważalne, gdy pokaże się obraz będący różnicą między obrazem źródłowym i obrazem przefiltrowanym (rys. 3.44).



Rys. 3.44. Skutki filtracji konwolucyjnej - po lewej stronie prosty obraz różnicowy (przeskalowany), po prawej - obraz modułu różnicicy.

W celu zmniejszenia negatywnych skutków filtracji konwolucyjnej stosuje się czasem filtry uśredniające wartości pikseli wewnątrz rozważanego obszaru w sposób „ważony” - tak, aby pierwotna wartość piksela  $L(m,n)$  w większym stopniu wpływała na wartość piksela po przetworzeniu  $L'(m,n)$ . Stosuje się w tym celu macierze konwolucji zawierające większe wzmocnienie dla punktu  $L(m,n)$  w stosunku do jego sąsiadów. Mogą to być maski:

1	1	1
1	2	1
1	1	1

1	2	1
2	4	2
1	2	1

Efekty działania takich masek mogą istotnie być mniej „dewastujące” dla obrazu, co można zaobserwować na rysunku 3.45.

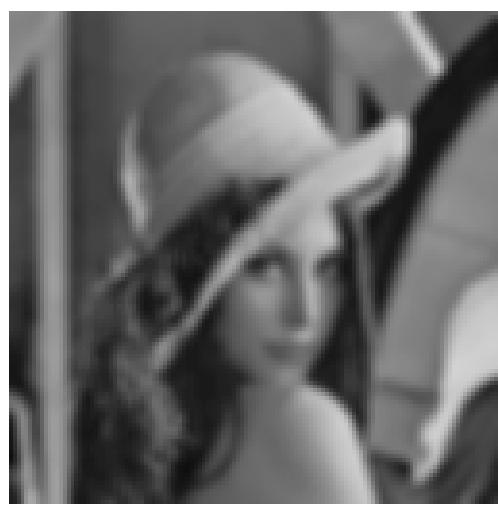


Rys. 3.45. Obraz „Lena” poddany filtracji z uwydatnionym punktem centralnym w macierzy konwolucji.

Czasem z kolei przeciwnie - stosowana jest konwolucja z maską pozabawioną całkowicie centralnego elementu:

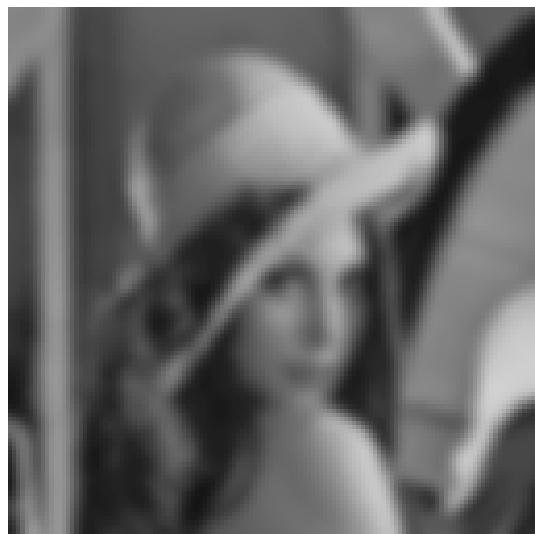
1	1	1
1	0	1
1	1	1

Efekt takiej filtracji obejrzeć można na rysunku 3.46.



Rys. 3.46. Obraz „Lena” poddany filtracji z pominiętym punktem centralnym w macierzy konwolucji.

Opisywane wyżej efekty dotyczyły skutków kontekstowej filtracji obrazów przy założeniu, że operujemy maską o rozmiarach  $3 \times 3$  piksele. Warto chociaż raz zobaczyć, jaki skutek daje powiększenie tej maski. Na rysunku 3.47 pokazano wynik dolnoprzepustowej filtracji obrazu przy użyciu maski o rozmiarach  $5 \times 5$  pikseli.



Rys. 3.47. Obraz „Lena” poddany filtracji za pomocą macierzy konwolucji o rozmiarach  $5 \times 5$ .

Łatwo zauważyc, że powiększenie zakresu kontekstu powoduje znacznie bardziej „radykalne” działanie filtra, co na ogół objawia się znacznie silniejszą „erozją” obrazu. Równocześnie większy rozmiar maski konwolucji prowadzi do znacznego zwiększenia pracochłonności obliczeń związanych z procesem przetwarzania obrazu. Na przykład konwolucja typowego obrazu o rozmiarach  $512 \times 512$  pikseli z maską  $5 \times 5$  wymaga 6.553.600 mnożeń i dodawań dla uzyskania obrazu wynikowego, zaś zwiększenie maski do rozmiarów  $7 \times 7$  pikseli powoduje zapotrzebowanie na wykonanie 16.384.000 mnożeń i dodawań w celu uzyskania jednego obrazu. Bez wątpienia zdarzają się okoliczności, kiedy trzeba usunąć wyjątkowo dokuczliwe zakłócenia z obrazu - i wtedy stosuje się większe maski przy filtracji dolnoprzepustowej. Na ogół jednak się ich unika, a przytoczona wyżej dyskusja stanowi wystarczające uzasadnienie dla takiego postępowania.

#### 3.4.2.5 Filtry górnoprzepustowe - gradienty

Podobnie jak przy przetwarzaniu innych sygnałów - w przetwarzaniu obrazów mają swój udział i swoją rolę do odegrania także filtry górnoprzepustowe. Filtry tego typu służą do wydobywania z obrazu składników odpowiedzialnych

za szybkie zmiany jasności - a więc konturów, krawędzi, drobnych elementów faktury, itp. Popularnie mówi się, że filtry górnoprzepustowe dokonują wyostrzania sygnału. Ponieważ pojęcie wyostrzania obrazu może być w praktyce trudne do jednoznacznego określenia, przeważnie przyjmuje się, że operacja ta polega na uwypukleniu krawędzi obiektów na obrazie.

**Krawędź** jest to linia (w najprostszym przypadku prosta) oddzielająca obszary o różnej jasności  $L_1, L_2$ . Prosty model matematyczny krawędzi ma postać skoku jednostkowego w zerze:

$$i(z) = \begin{cases} 1; z > 0 \\ \frac{1}{2}; z = 0 \\ 0; z < 0 \end{cases}$$

Zachodzi więc zależność:

$$i(z) = \int_{-\infty}^z \delta(t) dt; \text{ gdzie } \delta(t) - \text{delta Diraca}$$

Dla potrzeb analizy można przyjąć, że krawędź leży wzdłuż prostej o równaniu:

$$y = ax + b$$

Ponadto:  $a = \operatorname{tg}\alpha = \frac{\sin\alpha}{\cos\alpha}$ ;  $b = \frac{\beta}{\cos\alpha}$ ;  $\alpha, \beta$  - stałe.

$$y = \frac{\sin\alpha}{\cos\alpha}x + \frac{\beta}{\cos\alpha}$$

$$y \cos\alpha = x \sin\alpha + \beta$$

Postać kanoniczna:

$$x \sin\alpha - y \cos\alpha + \beta = 0$$

Jasność obrazu w poszczególnych punktach opisać można funkcją:

$$f(x, y) = L_1 + (L_2 - L_1)i(x \sin\alpha - y \cos\alpha + \beta)$$

Pierwsze pochodne cząstkowe mają postać:

$$\begin{aligned} \frac{\partial f}{\partial x} &= \sin\alpha(L_2 - L_1)\delta(x \sin\alpha - y \cos\alpha + \beta) \\ \frac{\partial f}{\partial y} &= -\cos\alpha(L_2 - L_1)\delta(x \sin\alpha - y \cos\alpha + \beta) \end{aligned}$$

Wektor  $\left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right)$  nazywany jest gradientem intensywności. Do wykrywania krawędzi używany jest kwadrat jego długości:

$$\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2 = ((L_2 - L_1)\delta(x \sin \alpha - y \cos \alpha + \beta))^2$$

Obliczając drugie pochodne  $f$ :

$$\begin{aligned}\frac{\partial^2 f}{\partial x^2} &= \sin^2 \alpha (L_2 - L_1) \delta'(x \sin \alpha - y \cos \alpha + \beta) \\ \frac{\partial^2 f}{\partial x \partial y} &= -\sin \alpha \cos \alpha (L_2 - L_1) \delta'(x \sin \alpha - y \cos \alpha + \beta) \\ \frac{\partial^2 f}{\partial y^2} &= \cos^2 \alpha (L_2 - L_1) \delta'(x \sin \alpha - y \cos \alpha + \beta)\end{aligned}$$

otrzymujemy Laplasjan funkcji intensywności:

$$\nabla^2 f(x, y) = \frac{\partial^2 f(x, y)}{\partial x^2} + \frac{\partial^2 f(x, y)}{\partial y^2} = (L_2 - L_1) \delta'(x \sin \alpha - y \cos \alpha + \beta)$$

Laplasjan jest symetryczny względem obrotu oraz zachowuje znak krawędzi (mówi o tym, czy intensywność obrazu za krawędzią jest większa, czy mniejsza od intensywności obrazu przed krawędzią). Co ważne, jest to operator liniowy, a więc prosty w implementacji (np. jako konwolucja). Dlatego znajduje szerokie zastosowanie w przetwarzaniu obrazów.

Jeśli w poprzednim rozdziale mogliśmy mówić, że filtry dolnoprzepustowe w pewnym sensie całkują, czy też uśredniają obraz - tak obecnie musimy stwierdzić, że filtry górnoprzepustowe realizują zawsze jakąś formę różniczkowania sygnału. Obejrzyjmy to najpierw na najprostszym możliwym przykładzie tak-zwanego **gradientu Robertsa**. Macierz współczynników  $w(i,j)$  dla tego gradientu może być przedstawiona w postaci następującej tabeli:

0	0	0
-1	0	0
0	1	0

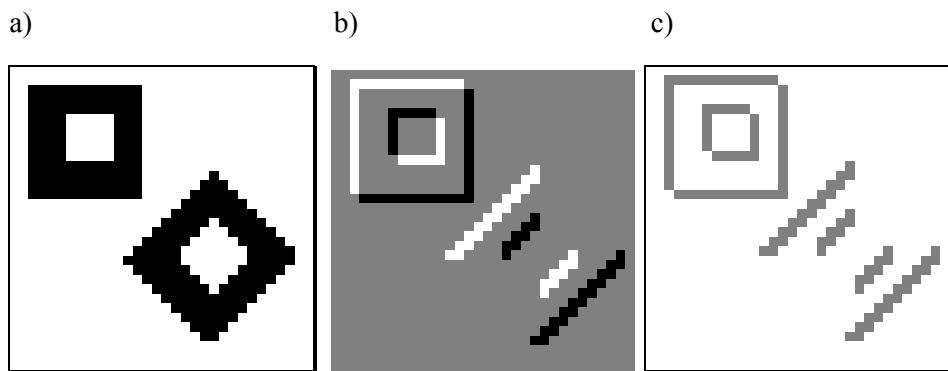
W zasadzie, jak widać z tabeli, gradient Robertsa daje się opisać jednoznacznie i wyczerpująco za pomocą macierzy współczynników o rozmiarach  $2 \times 2$

-1	0
0	1

jednak taki sposób zdefiniowania potrzebnego operatora konwolucji pozostawia pewną dowolność w zakresie sposobu umiejscowienia centralnego piksela, dla którego proces przetwarzania jest prowadzony - stąd w tej książce macierze dla

gradientu Robertsa będą także traktowane jako macierze o rozmiarach  $3 \times 3$  piksele.

Skutki różniczkowania obrazu gradientem Robertsa można prześledzić na rysunku 3.48. Dzięki temu, że na rysunku pokazano działanie gradientu na obrazie sztucznym, specjalnie do tego celu spreparowanym, ujawnione zostało, że istota filtracji górnoprzepustowej polega na podkreślaniu pewnych krawędzi.

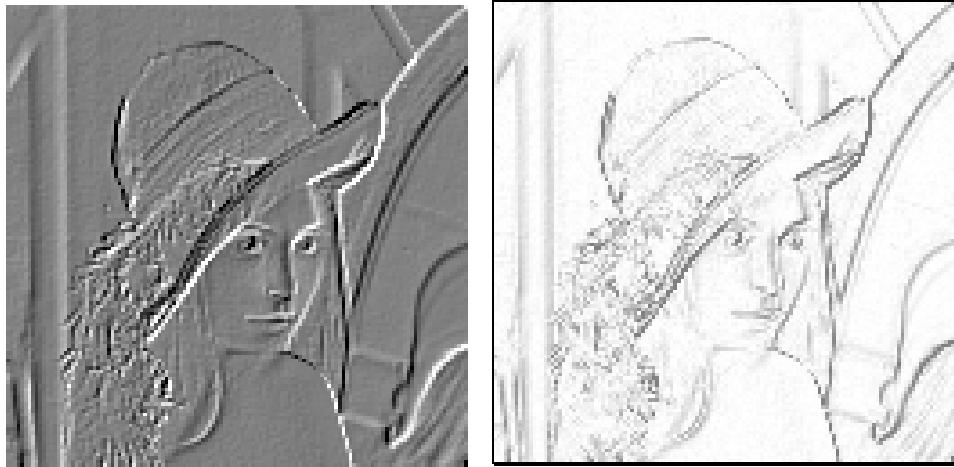


Rys. 3.48. Filtracja obrazu gradientem Robertsa. a) obraz poddawany filtracji, b) obraz po wykonaniu filtracji, c) obraz po wzięciu modułu wartości pikseli po wykonaniu filtracji.

Warto zwrócić uwagę na pewien szczegół rysunku 3.48, który będzie się potem powtarzał na dalszych rysunkach, w tym i w dalszych podrozdziałach. Otóż w macierzach konwolucji filtrów górnoprzepustowych występują zarówno dodatnie, jak i ujemne współczynniki, co oznacza, że po wykonaniu splotu w obrazie wynikowym będą występowały zarówno piksele o wartościach dodatnich, jak i piksele o wartościach ujemnych (warto zauważyć, że zjawisko takie nie występowało nigdy podczas filtracji dolnoprzepustowych). W związku z tym w celu prawidłowego zinterpretowania wyniku konwolucji w postaci wynikowego obrazu trzeba albo dokonać skalowania, albo brać pod uwagę jedynie wartość bezwzględną odpowiednich pikseli. Przy skalowaniu obrazu wadą jest fakt, że dotyczasowe tło obrazu otrzymuje pewne „zabarwienie” (zwykle jest to szarość odpowiadająca wartości pikseli wynoszącej 128), natomiast piksele dodatnie są intensywnie ciemne, zaś piksele ujemne - pojawiają się jako jaśniejsze od tła.

Alternatywny sposób prezentacji, zakładający tworzenie obrazu wynikowego w postaci wartości bezwzględnych (modułów) wartości odpowiednich pikseli też ma swoje zalety, ale zwykle trochę „rozmywa” efekty, a ponadto ukrywa informacje o tym, czy poszczególne zmiany obliczone zostały jako dodatnie (wzrost jasności) czy jako ujemne (zmniejszenie jasności w odpowiednim kierunku). Przesłedzenie działania gradientu Robertsa na obrazie rzeczywistym (pokazane na

rysunku 3.49) uwidacznia, że efekt końcowy może być subiektywnie odbierany jako „wystrzenie” obrazu, chociaż towarzyszą temu wyraźne zniekształcenia.



Rys. 3.49. Filtracja obrazu „Lena” gradientem Robertsa (pokazano obraz po wykonaniu filtracji; przeskalowany - po lewej oraz w postaci modułu - po prawej).

Gradient Robertsa ma wyraźnie kierunkowy charakter, co można prześledzić zarówno na rysunku 3.48, jak i na rysunku 3.49. Jest to oczywiste następstwo faktu, że różniczkowanie funkcji dwuwymiarowej musi zawsze odbywać się wzdłuż pewnego kierunku - w przypadku gradientu Robertsa jest to kierunek położony pod kątem  $45^\circ$ . Jeszcze lepiej można to prześledzić używając gradientu Robertsa dla komplementarnej maski o postaci:

0	-1
1	0

a dokładniej (dla uniknięcia nieporozumień odnośnie ulokowania centralnego punktu) maski:

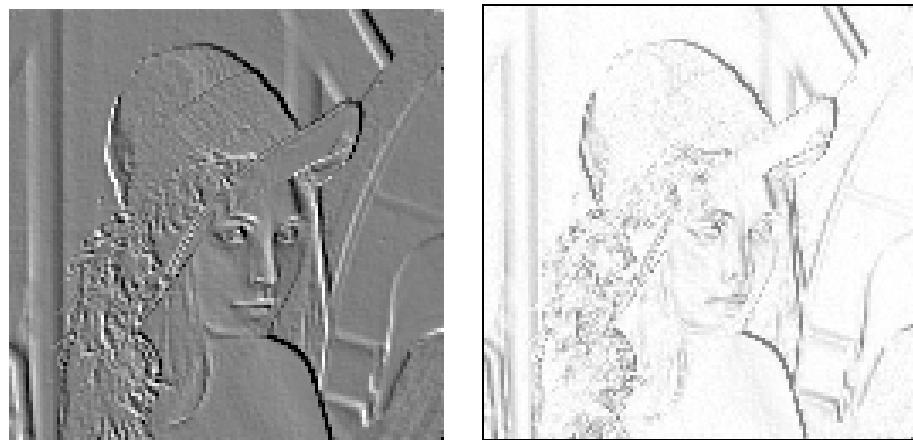
0	0	0
0	0	-1
0	1	0

Skutki działania tego gradientu, pokazane na rysunku 3.50 nie pozostawiają żadnych wątpliwości: gradient podkreśla zawsze linie o pewnej ustalonej orientacji, zależnej od postaci używanej maski.

Gradienty Robertsa (oparte na masce o rozmiarach  $2 \times 2$ ) można w dość naturalny sposób przenieść na maski o rozmiarach  $3 \times 3$ . Powstające w ten sposób tzw.

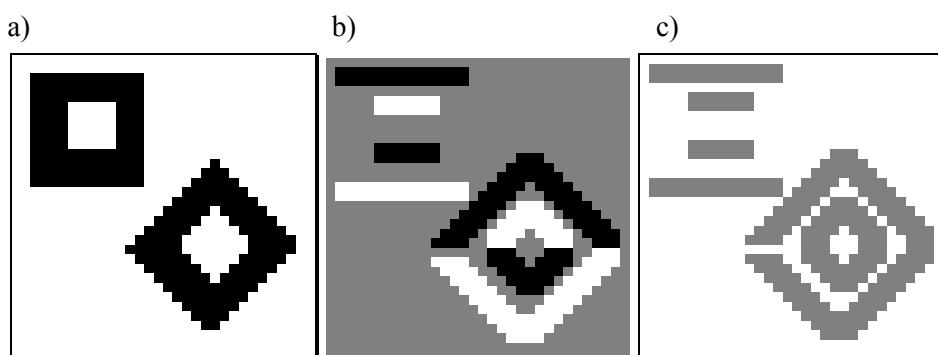
maski Prewitta pozwalają różniczkować obraz w różnych kierunkach. Rozważmy na początek poziomą maskę Prewitta o następującej strukturze:

-1	-1	-1
0	0	0
1	1	1



Rys. 3.50. Filtracja obrazu „Lena” gradientem Robertsa o przeciwej (w stosunku do rys. 3.49) orientacji.

Maska ta po dokonaniu operacji konwolucji produkuje dla sztucznego obrazu wynik pokazany na rysunku 3.51.



Rys. 3.51. Sztuczny obraz filtrowany górnoprzepustowo z użyciem poziomej maski Prewitta.

Nie ulega wątpliwości, że tym razem wzmacnieniu podlegają linie o orientacji zbliżonej do horyzontalnej. Potwierdza to także wygląd obrazu uzyskanego po spleceniu maski poziomej Prewitta z obrazem „Lena” (pokazany na rysunku 3.52).



Rys. 3.52. Obraz „Lena” filtrowany górnoprzepustowo z użyciem poziomej maski Prewitta.

Przez prostą transpozycję maski Prewitta można uzyskać filtr akcentujący elementy o orientacji horyzontalnej. Efekt działania tego filtra obejrzeć można na rysunku 3.53.



Rys. 3.53. Obraz „Lena” filtrowany górnoprzepustowo z użyciem pionowej maski Prewitta.

Przy filtracjach gradientowych można także wzmacniać wpływ bezpośrednio najbliższego otoczenia piksela, dla którego wyznaczana jest wartość piksela na obrazie wynikowym. Służą do tego tak zwane maski Sobela. Pozioma maska Sobela ma postać:

-1	-2	-1
0	0	0
1	2	1

Skutek działania tej maski pokazuje rysunek 3.54.



Rys. 3.54. Obraz „Lena” filtrowany górnoprzepustowo z użyciem poziomej maski Sobela.

Maskę Sobela można też transponować (obrócić) uzyskując maskę pionową:

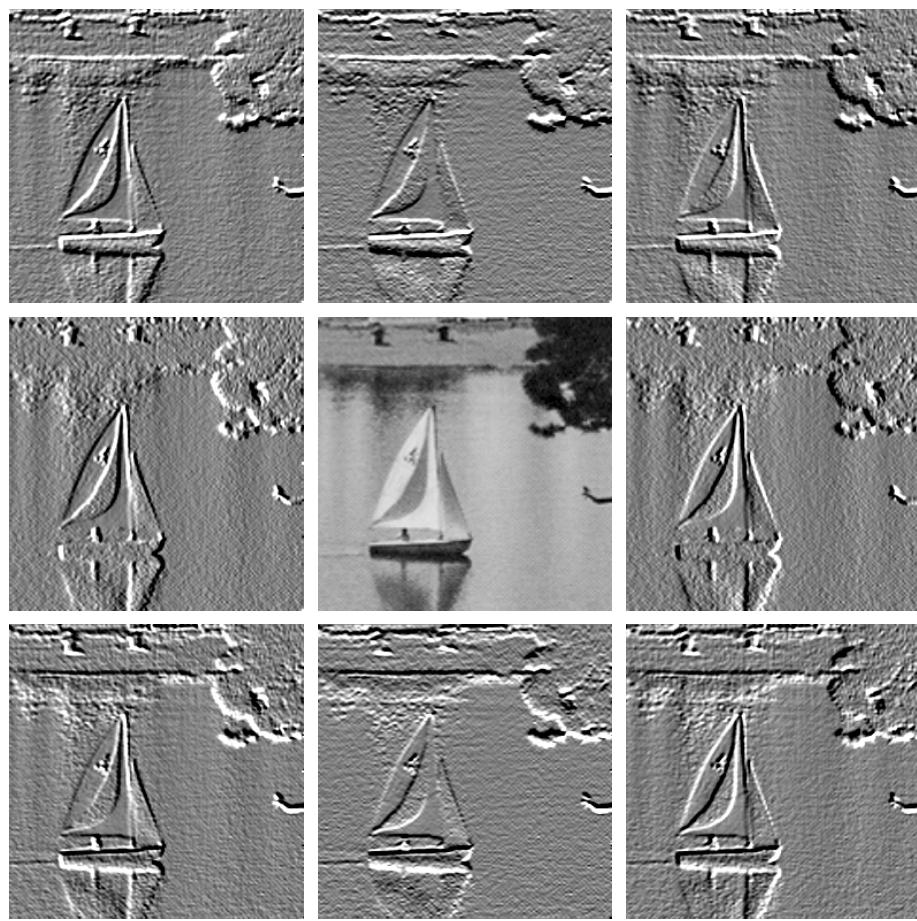
-1	0	1
-2	0	2
-1	0	1

której efekt działania pokazuje rysunek 3.55.

Mało tego - maski Sobela mogą być swobodnie obracane nie tylko o  $90^\circ$ , co pozwala na wyznaczenie gradientów w różnych kierunkach. Dla pełnego pokazania tego efektu użyjemy jednak innego obrazu testowego, gdyż obraz „Lena” znany jest ze swych krągłości (!!!), a tutaj potrzebne są proste linie o różnych kierunkach. Dlatego chcąc pokazać skutek działania gradientów opartych na maskach Sobela obracanych kolejno o  $45^\circ$  użyjemy obrazka przedstawiającego żaglowkę na jeziorze. Wynik pokazuje rysunek 3.56.



Rys. 3.55. Obraz „Lena” filtrowany górnoprzepustowo z użyciem poziomej maski Sobela.



Rys. 3.56. Obraz żaglówki (w centrum) filtrowany górnoprzepustowo z użyciem masek Sobela o różnych orientacjach.

Do filtracji obrazka żaglówki zastosowano na rysunku 3.56 maski Sobela, które pokazano niżej na rysunku 3.57 w tym samym ułożeniu, jakie wynikło z ich użycia na rysunku 3.56. Łatwo zauważać porównując rysunki 3.56 i 3.57, że maska Sobela o określonej orientacji wydobywa z obrazu linie i struktury o tej właśnie orientacji.

<table border="1"><tr><td>2</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>-1</td></tr><tr><td>0</td><td>-1</td><td>-2</td></tr></table>	2	1	0	1	0	-1	0	-1	-2	<table border="1"><tr><td>1</td><td>2</td><td>1</td></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>-1</td><td>-2</td><td>-1</td></tr></table>	1	2	1	0	0	0	-1	-2	-1	<table border="1"><tr><td>0</td><td>1</td><td>2</td></tr><tr><td>-1</td><td>0</td><td>1</td></tr><tr><td>-2</td><td>-1</td><td>0</td></tr></table>	0	1	2	-1	0	1	-2	-1	0
2	1	0																											
1	0	-1																											
0	-1	-2																											
1	2	1																											
0	0	0																											
-1	-2	-1																											
0	1	2																											
-1	0	1																											
-2	-1	0																											
<table border="1"><tr><td>1</td><td>0</td><td>-1</td></tr><tr><td>2</td><td>0</td><td>-2</td></tr><tr><td>1</td><td>0</td><td>-1</td></tr></table>	1	0	-1	2	0	-2	1	0	-1		<table border="1"><tr><td>-1</td><td>0</td><td>1</td></tr><tr><td>-2</td><td>0</td><td>2</td></tr><tr><td>-1</td><td>0</td><td>1</td></tr></table>	-1	0	1	-2	0	2	-1	0	1									
1	0	-1																											
2	0	-2																											
1	0	-1																											
-1	0	1																											
-2	0	2																											
-1	0	1																											
<table border="1"><tr><td>0</td><td>-1</td><td>-2</td></tr><tr><td>1</td><td>0</td><td>-1</td></tr><tr><td>2</td><td>1</td><td>0</td></tr></table>	0	-1	-2	1	0	-1	2	1	0	<table border="1"><tr><td>-1</td><td>-2</td><td>-1</td></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>1</td><td>2</td><td>1</td></tr></table>	-1	-2	-1	0	0	0	1	2	1	<table border="1"><tr><td>-2</td><td>-1</td><td>0</td></tr><tr><td>-1</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>2</td></tr></table>	-2	-1	0	-1	0	1	0	1	2
0	-1	-2																											
1	0	-1																											
2	1	0																											
-1	-2	-1																											
0	0	0																											
1	2	1																											
-2	-1	0																											
-1	0	1																											
0	1	2																											

Rys. 3.57. Maski Sobela użyte do górnoprzepustowej filtracji obrazu na rysunku 3.56.

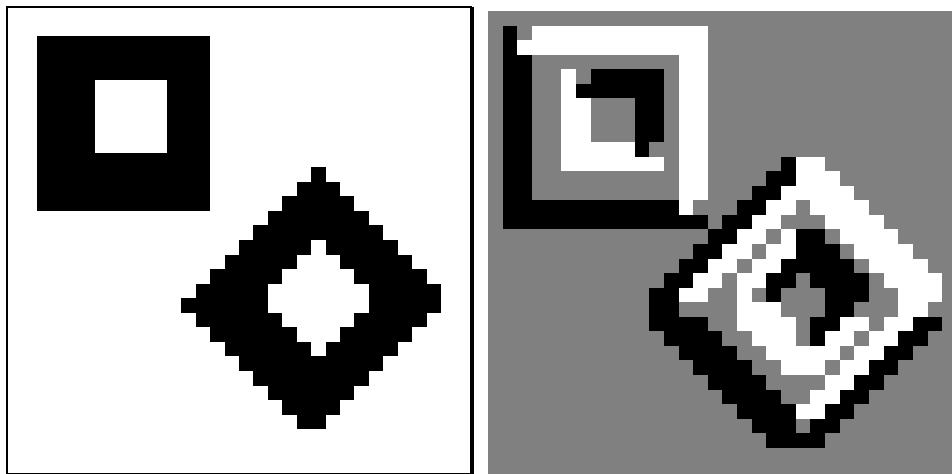
#### 3.4.2.6 Filtry górnoprzepustowe wykrywające narożniki

Obok masek służących do wyznaczania gradientów kierunkowych (Roberts, Prewitta, Sobela i innych, nie opisywanych w tej książce, ale także spotykanych w praktyce - na przykład w pakietach programów do przetwarzania obrazów) do górnoprzepustowej filtracji obrazu stosowane bywają także inne maski, na przykład maska:

1	1	1
-1	-2	1
-1	-1	1

która zwykle kojarzy się z operacją wykrywania narożników (w pokazanym wyżej przypadku chodzi o prawy górny narożnik). Wynik konwolucji tej maski ze

znanym już z wcześniejszych przykładów sztucznym obrazem testowym potwierdza taką interpretację omawianego filtra (rys. 3.58).



Rys. 3.58. Splot obrazu testowego pokazanego po lewej stronie z maską wykrywającą prawy górny narożnik.

Obraz „Lena” potraktowany omawianą maską daje mniej oczywisty obraz (rysunek 3.59), niemniej można przyjąć, że istotnie wykrywane są tu głównie narożniki (jeśli „Lena” ma jakieś narożniki).

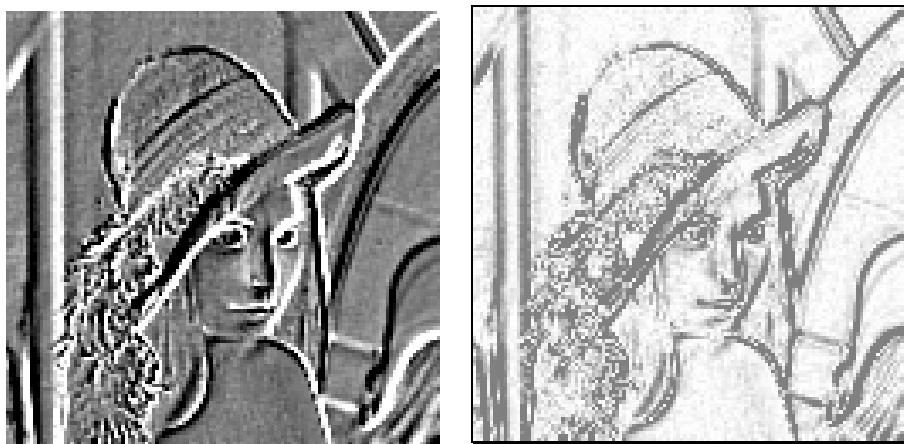


Rys. 3.59. Obraz „Lena” po wykonaniu konwolucji z maską wykrywającą narożniki. Po lewej stronie obraz wynikowy skalowany, po prawej - obraz po wykonaniu inwersji i uwzględnieniu tylko bezwzględnych wartości pikseli.

Po obróceniu maski w taki sposób, by wykrywała lewy górny narożnik, przyjmuje ona postać:

1	1	1
1	-2	-1
1	-1	-1

Przetworzone obrazy „Lena” wyglądają jak na rysunku 3.60.



Rys. 3.60. Obraz „Lena” po wykonaniu konwolucji z obróconą maską wykrywającą narożniki. Układ jak na rysunku 3.59.

Oczywiście możliwych form, w jakich mogą występować maski wykrywające narożniki jest znacznie więcej. Niżej pokazano niektóre dalsze spośród nich.

-1	1	1
-1	-2	1
-1	1	1

1	1	1
1	-2	1
-1	-1	-1

-1	-1	1
-1	-2	1
1	1	1

-1	-1	-1
1	-2	1
1	1	1

1	-1	-1
1	-2	-1
1	1	1

1	1	-1
1	-2	-1
1	1	-1

1	1	1
1	-2	-1
1	-1	-1

Narożniki można też wykrywać i uwidaczniać z pomocą innych macierzy konwolucji, na przykład z pomocą macierzy:

3	3	3
3	0	-5
3	-5	-5

3	-5	-5
3	0	-5
3	3	3

i dalszych podobnych, powstających w wyniku łatwych do wyobrażenia obrotów.

#### 3.4.2.7 Filtry górnoprzepustowe wykrywające krawędzie - laplasjany

Wspólną cechą wszystkich wyżej wymienionych operatorów, wykorzystywanych do wyznaczania gradientów obrazu lub do wykrywania narożników (jako specjalnych form górnoprzepustowej filtracji obrazu), był fakt, że odpowiednie macierze konwolucji wykazywały określoną asymetrię. Następstwem tej asymetrii były kierunkowe własności realizowanych z użyciem tych macierzy transformacji

obrazu - podkreślały one lub tłumiały pewne cechy obrazu - zależnie od tego, pod jakim kątem przebiegały określone krawędzie obiektów na obrazie w stosunku do kierunku asymetrii macierzy konwolucji.

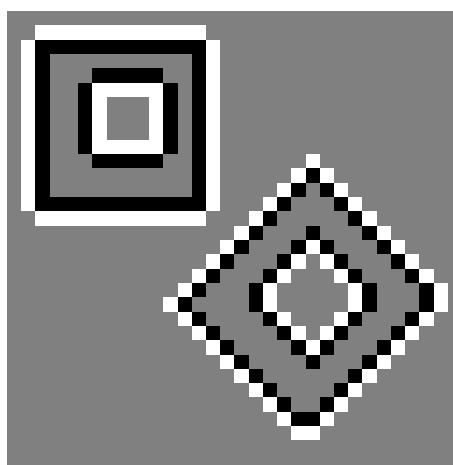
Czasem jednak zadania stawiane przed algorytmem przetwarzającym obrazy mają wyraźnie bezkierunkowy charakter - wtedy do filtracji górnoprzepustowej trzeba użyć innego typu masek. Typowym zadaniem, przy realizacji którego trzeba uwolnić się od kierunkowego działania maski jest zadanie wykrywania i podkreślania na obrazie wszelkich krawędzi i konturów obiektów, niezależnie od tego, pod jakim kątem one przebiegają. Do tego celu można używać różnych metod, w większości nielinowych (i w związku z tym dyskutowanych w następnym podrozdziale), ale w prostych zadaniach dobre efekty uzyskać można stosując tak zwane **laplasjany**. Z matematycznego punktu widzenia laplasjan jest kombinacją drugich pochodnych cząstkowych wejściowej funkcji  $L(m,n)$ :

$$L'(m,n) = \frac{\partial^2 L(m,n)}{\partial m^2} + \frac{\partial^2 L(m,n)}{\partial n^2}$$

jednak z praktycznego punktu widzenia w przetwarzaniu obrazów laplasjanem nazywa się wynik konwolucji obrazu z maską:

0	-1	0
-1	4	-1
0	-1	0

Wynik ten dla sztucznego obrazu testowego można obejrzeć na rysunku 3.61, a dla obrazu rzeczywistego (oczywiście „Lena”) - na rysunku 3.62.



Rys. 3.61. Przeskalowany wynik splotu sztucznego obrazu testowego z maską laplasjanu.

Istotnie łatwo się upewnić, że operacja oparta na wykorzystaniu maski laplasjanu uwypukla i podkreśla wszelkie linie i krawędzie na obrazie.

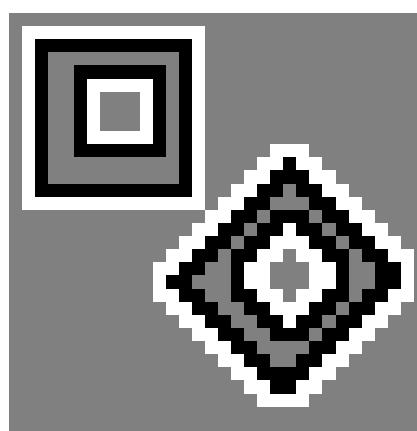


Rys. 3.62. Obraz „Lena” po zastosowaniu laplasjanu. Po lewej obraz skalowany, po prawej - moduł wartości pikseli.

Laplasjan może być też zdefiniowany za pomocą nieco bardziej rozbudowanej maski

-1	-1	-1
-1	8	-1
-1	-1	-1

która daje nieco odmienne efekty, co pokazano na rysunku 3.63 dla obrazu sztucznego i na rysunku 3.64 dla obrazu „Lena”.



Rys. 3.63. Przeskalowany wynik splotu sztucznego obrazu testowego z maską rozbudowanego laplasjanu.



Rys. 3.64. Obraz „Lena” po zastosowaniu rozbudowanego laplasjanu.

Rozbieżności między obiema formami definicji laplasjanu można dodatkowo prześledzić analizując obraz pokazany na rysunku 3.65, będący różnicą między obrazem otrzymanym za pomocą zwykłego laplasjanu i laplasjanu rozbudowanego.



Rys. 3.65. Obraz różnic po zastosowaniu zwykłego i rozbudowanego laplasjanu.

Inne sposoby definicji laplasjanu pokazują podane niżej maski:

1	-2	1
-2	4	-2
1	-2	1

-1	-1	-1
-1	9	-1
-1	-1	-1

0	-1	0
-1	5	-1
0	-1	0

1	-2	1
-2	5	-2
1	-2	1

Oczywiście, laplasjan wyznaczać można przy wykorzystaniu macierzy konwolucji większych niż 3x3, na przykład 5x5:

0,25	0,5	0,5	0,5	0,25
0,5	-0,25	-1	-0,25	0,5
0,5	-1	-2	-1	0,5
0,5	-0,25	-1	-0,25	0,5
0,25	0,5	0,5	0,5	0,25

### 3.4.3 Filtry nieliniowe

Przy wszystkich swoich zaletach filtry liniowe nie we wszystkich przypadkach prawidłowo filtruają obraz, w związku z tym zajmiemy się obecnie kilkoma przykładami filtrów dokonujących filtracji obrazu w sposób nieliniowy.

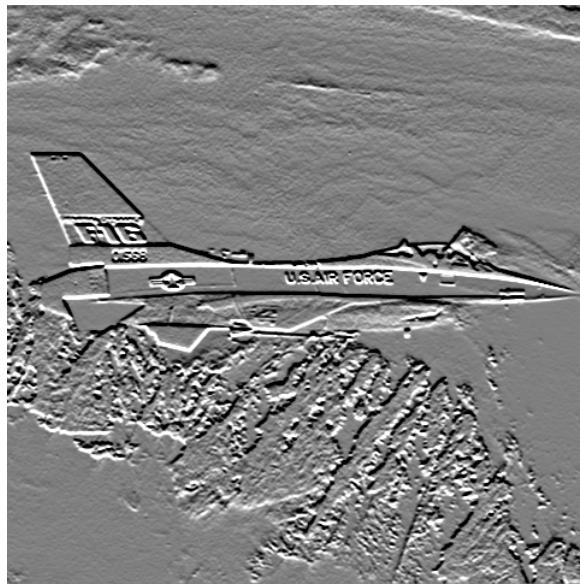
#### 3.4.3.1 Filtry kombinowane wykrywające krawędzie

Opisane wyżej laplasjany pozwalają wykrywać krawędzie i kontury obiektów na obrazach z dość dobrą dokładnością w większości praktycznych zastosowań, jednak wtedy, gdy chcemy użyć potem wyznaczonych krawędzi do jakichś bardziej zaawansowanych technik przetwarzania obrazów - potrzebne jest narzędzie lokalizujące kontury dokładniej i w sposób bardziej pewny niż laplasjan. W opisanych okolicznościach uciekamy się najczęściej do filtrów nieliniowych znanych pod nazwą filtrów kombinowanych. Idea tych filtrów polega na kolejnym zastosowaniu dwóch gradientów w prostopadłych do siebie kierunkach, a następnie na dokonaniu nieliniowej kombinacji wyników tych gradientów. Dzięki nieliniowej kombinacji rezultatów liniowych transformacji obrazu tworzy się w ten sposób obraz wynikowy o wyjątkowo dobrze podkreślonych konturach - niezależnie od kierunku ich przebiegu.

Rozważmy to na konkretnym przykładzie. Powiedzmy, że chcemy wydobyć kontury z obrazu o nazwie *Jet* pokazanego na rysunku 3.66. Może się wydać dziwne, że nie używamy tu - jak zawsze - obrazu „Lena”, jednak powód takiego wyboru zostanie podany dopiero na końcu tego podrozdziału.



Rys. 3.66. Obraz, z którego wydobywane będą kontury.

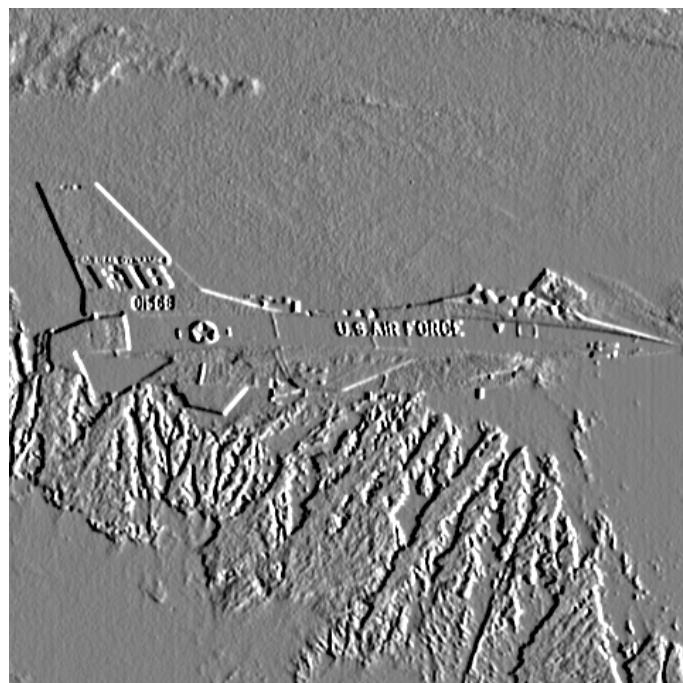


Rys. 3.67. Obraz  $L_I(m,n)$  z wydobytymi konturami poziomymi.

Zgodnie z podaną metodyką najpierw wyznaczymy dwa prostopadłe do siebie gradienty tego obrazu. Można w tym celu użyć dowolnej spośród masek opisywanych w podrozdziale 3.4.2.5, jednak tradycja w dziedzinie przetwarzania obrazów nakazuje tu użycie masek Sobela.

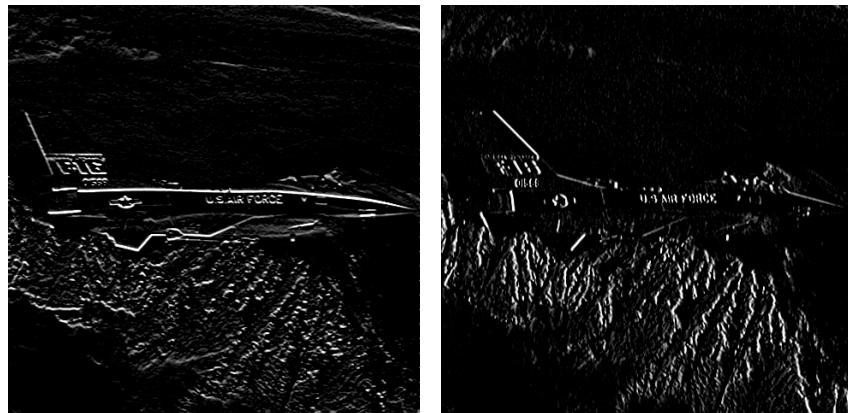
Po przefiltrowaniu obrazu Jet za pomocą maski Sobela wykrywającej krawędzie poziome otrzymuje się obraz pomocniczy  $L_1(m,n)$  pokazany na rysunku 3.67.

Analogicznie - po przefiltrowaniu obrazu Jet za pomocą maski Sobela wykrywającej krawędzie pionowe otrzymuje się obraz pomocniczy  $L_2(m,n)$  pokazany na rysunku 3.68.



Rys. 3.68. Obraz  $L_2(m,n)$  z wydobytymi konturami pionowymi.

Teraz trzeba te obrazy połączyć razem. Ponieważ w wyniku zastosowania filtrów liniowych powstały na obrazach  $L_1(m,n)$  oraz  $L_2(m,n)$  obszary o zarówno dodatnich, jak i ujemnych wartościach pikseli - trzeba zastosować wzór, który wyeliminuje ewentualne efekty wzajemnego kompensowania się dodatnich i ujemnych wartości odpowiednich pikseli. O tym, że taki zabieg jest konieczny można się przekonać oglądając rysunek 3.69, na którym pokazano obrazy  $L_1(m,n)$  oraz  $L_2(m,n)$  bez skalowania - czyli ze zgubionymi ujemnymi pikselami. Widać wyraźnie, że w takim przypadku znaczna część wykrytych krawędzi po prostu ginie.

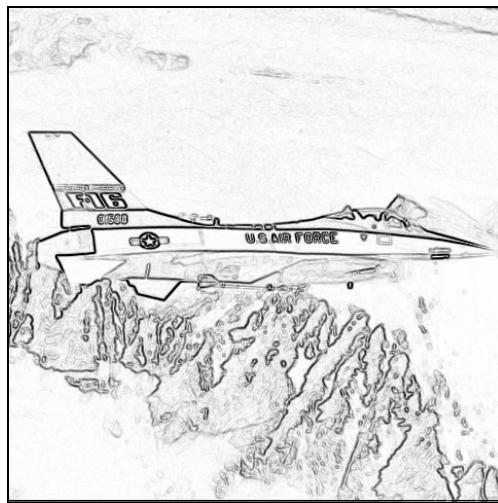


Rys. 3.69. Obrazy  $L_1(m,n)$  oraz  $L_2(m,n)$  pokazane bez skalowania.

Do połączenia (kombinowania) obrazów  $L_1(m,n)$  oraz  $L_2(m,n)$  użyć można formuły Euklidesowej:

$$L'(m,n) = \sqrt{(L_1(m,n))^2 + (L_2(m,n))^2}$$

Efekt jej działania pokazano na rysunku 3.70. Istotnie, jakość odtworzenia krawędzi na obrazie może być uznana za dobrą.

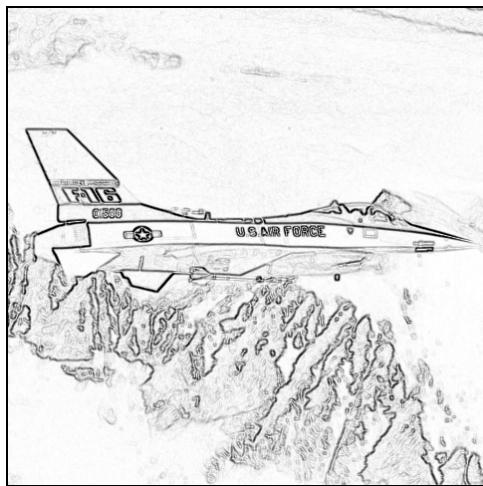


Rys. 3.70. Obraz konturowy powstały w wyniku Euklidesowej kombinacji obrazów  $L_1(m,n)$  i  $L_2(m,n)$ .

Biorąc pod uwagę złożoność obliczeniową Euklidesowej kombinacji obrazów  $L_1(m,n)$  i  $L_2(m,n)$  stosuje się czasem uproszczoną formułę modułową:

$$L'(m,n) = |L_1(m,n)| + |L_2(m,n)|$$

pozwalającą na uzyskiwanie praktycznie równie dobrych wyników (por. rysunek 3.71) przy zdecydowanie mniejszym koszcie obliczeniowym.



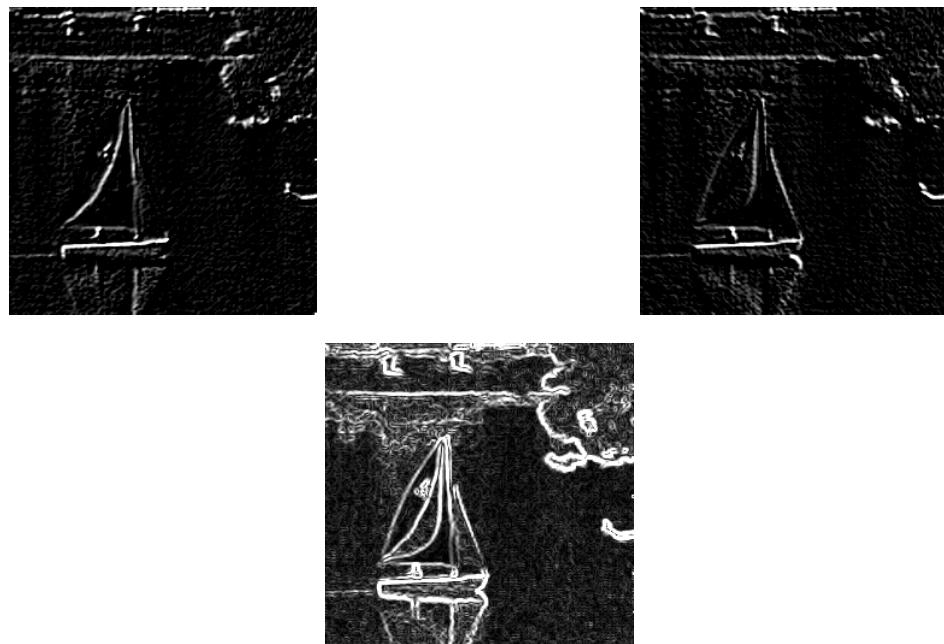
Rys. 3.71. Obraz konturowy powstały w wyniku modułowej kombinacji obrazów  $L_1(m,n)$  i  $L_2(m,n)$ .

Opisana wyżej procedura kombinacyjnej filtracji obrazu poprzez stosowania dwóch prostopadłych masek Sobela i kombinacji Euklidesowej lub modułowej - bywa w literaturze określana mianem „gradientu Sobela”. Na analogicznej zasadzie, zmieniając tylko maski, otrzymuje się niekiedy „gradient Prewitta”, a listę domyka (nie omawiany tu wcale) „gradient Kirscha”. Wszystkie te „gradienty” są oczywiście nazywane błędnie, jako że gradient jest liniowym operatorem o określonych właściwościach, niewiele mających wspólnego z opisanymi tu nieliniowymi przekształceniami kombinacyjnymi. Jednak tradycja w dziedzinie przetwarzania obrazów jest bardzo silna - i trzeba się z nią liczyć, stąd przytoczone wyżej uwagi terminologiczne.

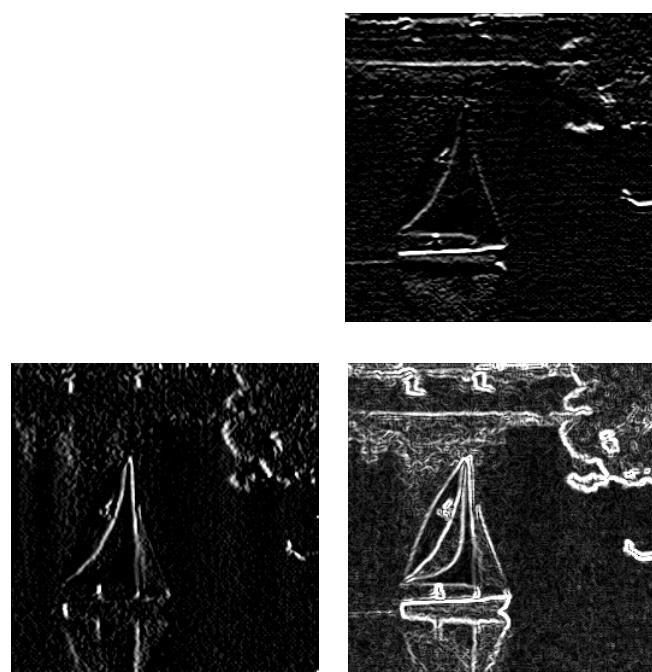
Przy tworzeniu filtrów kombinowanych najistotniejsze jest użycie obrazów  $L_1(m,n)$  i  $L_2(m,n)$  pochodzących z różniczkowania obrazu  $L(m,n)$  w dwóch różnych kierunkach, przy czym nie muszą to być wcale kierunki - poziomy i pionowy. Na rysunku 3.72 pokazano proces rekonstrukcji pełnego zestawu konturów obrazu żaglowki (por. rys. 3.56) na podstawie dwóch obrazów pochodzących z gradientów „skośnych”.

Optycznie kontury te nie różnią się od tych, jakie dla tego samego obrazu można uzyskać dla typowej metodyki wyznaczania „gradientu Sobela” - por. rys. 3.73.

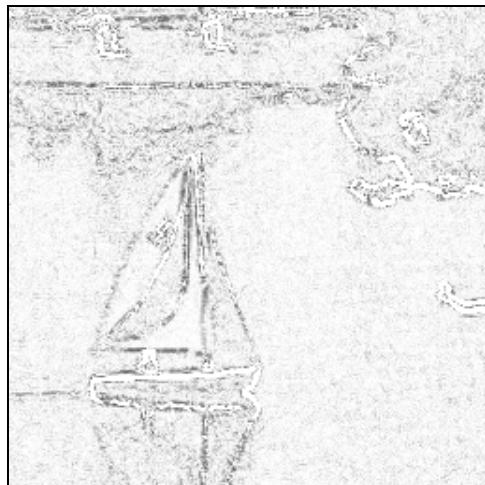
Jednak dla kompletu informacji trzeba dodać, że przy numerycznej realizacji obydwu rozważanych technik błędy zaokrągleń nieco inaczej się układają w obu rozważanych przypadkach, w związku z czym po odjęciu obrazu konturów z rysunku 3.71 od obrazu konturów z rysunku 3.72 uzyskuje się niezerowy obraz ujawniający błędy - pokazany na rysunku 3.73.



Rys. 3.72. Odtworzenie konturów z gradientów skośnych.



Rys. 3.73. Odtworzenie konturów typową metodą „gradientu Sobela”.



Rys. 3.73. Błedy odtworzenia konturów ze skosów i typową metodą gradientu Sobela”.

Na koniec tego podrozdziału pozostała już tylko jedna sprawa: wyjaśnienie, czemu to wszystko nie było pokazywane na standardowym obrazie „Lena”. Otóż obraz „Lena” po wydobyciu konturów gradientem Sobela traci cały swój naturalny urok (patrz rys. 3.74). Czy można było przez cały rozdział tak masakrować taką ładną dziewczynę?!



Rys. 3.74. Obraz „Lena” po wydobyciu konturów.

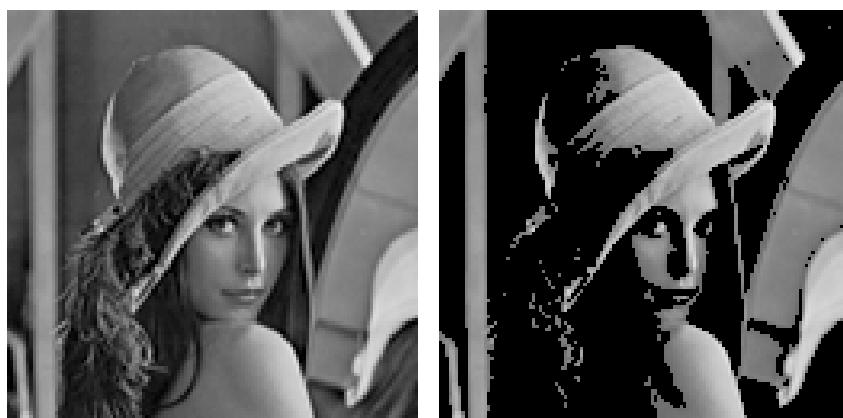
### 3.4.3.2 Operacje logiczne

Operacje logiczne najczęściej przeprowadzane są na obrazach binarnych (których każdy element może przyjąć tylko jeden spośród dwóch stanów logicznych *prawda* (1) i *falsz* (0). Dla większości operacji logicznych danymi wejściowymi są dwa obrazy, a rezultatem jest jeden obraz. Na poszczególnych punktach obrazu wykonywane są najczęściej następujące operacje logiczne:

- **NOT** - zaprzeczenie (negatyw obrazu - rys. 3.75),
- **AND** - iloczyn logiczny („wykrawanie” obrazu za pomocą „maski” - rys. 3.76),
- **OR** - suma logiczna („fotomontaż” - rys. 3.77),
- **SUB** - różnica logiczna (rys. 3.78a i 3.79a),
- **XOR** - suma rozłączna (rys. 3.78b i 3.79b),
- **NXOR** - równoważność logiczna (rys. 3.78c i 3.79c).



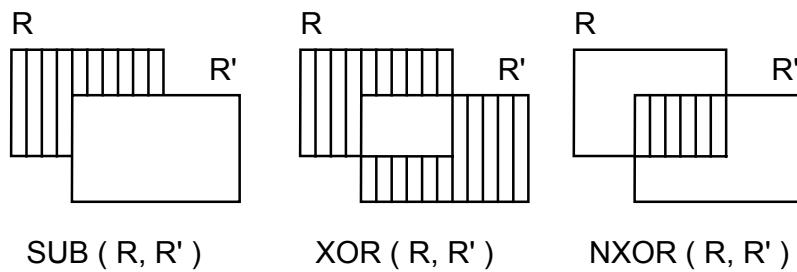
Rys. 3.75. Obraz binarny i jego negacja.



Rys. 3.76. Obraz „Lena” przed i po zastosowaniu maski.



Rys. 3.77. Obraz „Lena” po zastosowaniu logicznego sumowania z obrazem trzech kólek.



Rys. 3.78. Efekt działania operacji SUB, XOR i NXOR.



Rys. 3.79. Skutki działania operacji SUB, XOR i NXOR.

Dla obrazów o większej ilości stanów poszczególnego elementu (kolorów) tradycyjna logika Boole'a jest niewystarczająca. Należy wtedy stosować logikę wielowartościową lub tzw. rozmytą<sup>21</sup>. Dla przykładu dla obrazów monochromatycznych o poziomach jasności pojedynczego elementu od 0 do 255 można zdefiniować operacje logiczne następująco:

<sup>21</sup> ang. *Fuzzy Logic*

**a) negacja**

$$\bar{x} = \begin{cases} 0; & x \leq 127 \\ 1; & x > 127 \end{cases}$$

**b) iloczyn logiczny**

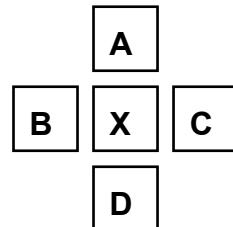
$$x \otimes y = \begin{cases} 0; & x * y \leq 16256 \\ 1; & x * y > 16256 \end{cases}$$

**c) suma logiczna**

$$x \oplus y = \begin{cases} 0; & x + y \leq 255 \\ 1; & x + y > 255 \end{cases}$$

Na podobnej zasadzie można zdefiniować też wszelkie inne operacje logiczne dla obrazów o wielowartościowej skali stopni szarości.

Dyskutowane wyżej filtry logiczne miały charakter bezkontekstowy. Można jednak równie dobrze zdefiniować kontekstowe operatory logiczne, działające nieco podobnie jak operatory filtracji dolnoprzepustowej. Zdefiniujmy - jak na rysunku 3.80 - oznaczenia identyfikujące piksele A, B, C, D otaczające piksel X podlegający w danym momencie procesowi przetwarzania.



Rys. 3.80. Otoczenie przetwarzanego piksela w operacjach filtracji logicznych

Wówczas wartości piksela  $X'$  po przetworzeniu logicznym mogą być uzyskane za pomocą jednej z niżej podanych formuł logicznych

$$X' = \begin{cases} A \text{ jeśli } A=D \\ X \text{ w innym przypadku} \end{cases}$$

$$X' = \begin{cases} B \text{ jeśli } B=C \\ X \text{ w innym przypadku} \end{cases}$$

$$X' = \begin{cases} A \text{ jeśli } A=B=C=D \\ X \text{ w innym przypadku} \end{cases}$$

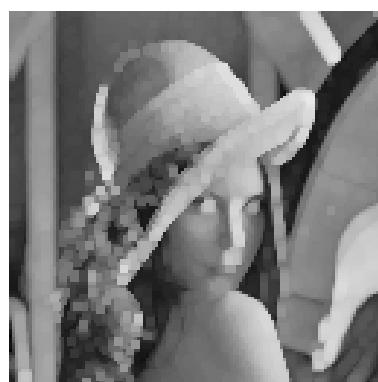
Formuły te mają stosunkowo prostą interpretację - jeśli otoczenie rozważanego piksela ma pewną ustaloną wartość - wówczas piksel centralny powinien mieć tę

samą wartość, co jego otoczenie - niezależnie od tego jaką jasność miał uprzednio. Jeśli przypomnijmy sobie, że dotyczy to sytuacji, w której wszystkie piksele mają wyłącznie jasność **0** albo **1**, to opisana zasada stanie się dla nas oczywista, oczywista i w pełni aprobowalna - służy to po prostu eliminacji izolowanych białych punktów na czarnym tle lub pojedynczych czarnych kropek na tle białym.

Podobnie jak w przypadku jednoargumentowych operatorów logicznych, opisane wyżej zasady można łatwo uogólnić na przypadek obrazów o pełnej skali stopni szarości - wystarczy po prostu zamiast idealnych równości typu  $B = D$ , itp. do magać się tylko tego, by odpowiednia różnica była dostatecznie mała. Na przykład można zapisać warunek:

$$X' = \begin{cases} A & \text{jeśli } \text{Abs}(A - D) < \epsilon \\ X & \text{w innym przypadku} \end{cases}$$

Wynik działania tej funkcji na obrazie „Lena” pokazuje rysunek 3.81.



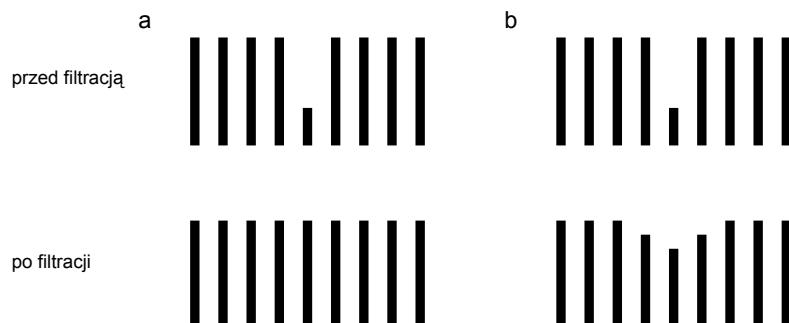
Rys. 3.81. Obraz „Lena” po kontekstowej filtracji logicznej.

### 3.4.3.3 Filtry medianowe

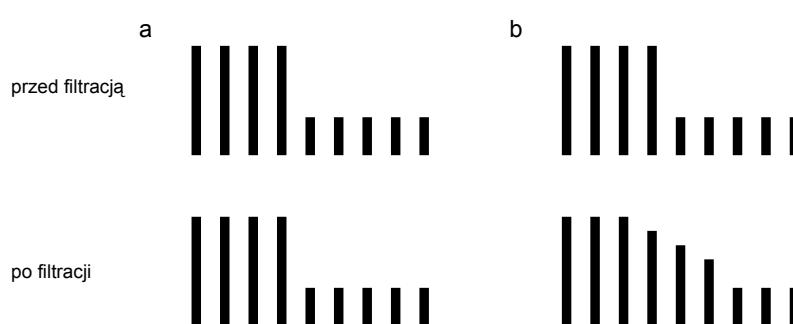
Większość omawianych wyżej filtrów miała jedną wspólną niemiłą cechę: usuwając zakłócenia niszczyły one także drobne szczegóły i krawędzie przetwarzanych obrazów. Lepsze efekty dają także i w tym zakresie filtry nieliniowe, wybierające dla przetwarzanego punktu na obrazie wynikowym jedną z wartości z jego otoczenia na obrazie źródłowym. Wybór ten dokonywany jest oczywiście według pewnej reguły, która w ogólnym przypadku może być dowolna. Najczęściej spotykanyem przykładem filtru działającego na tej zasadzie jest filtr wykorzystujący medianę. Przypomnijmy: mediana jest wartością środkową w uporządkowanym rosnąco ciągu wartości jasności pikseli z całego rozważanego otoczenia przetwarzanego piksela. Dla przykładowego rozkładu wartości jasności punktów w otoczeniu pewnego bardzo jasnego punktu:

$$\begin{bmatrix} 1 & 13 & 19 \\ 12 & 198 & 17 \\ 17 & 16 & 13 \end{bmatrix}$$

mediana spośród wartości jasności pikseli zawartych w oknie  $\{1, 12, 13, 13, 16, 17, 17, 19, 198\}$  przyjmuje wartość 16 - i taką właśnie wartość będzie miał odpowiedni piksel na obrazie wynikowym. Chwila zastanowienia pozwala upewnić się, że jest to wybór sensowny. Filtr medianowy jest filtrem mocnym, gdyż ekstremalne wartości, znacznie odbiegające od średniej (w rozważanym przypadku jest to pierwotna wartość przetwarzanego punktu wynosząca 198 i - także chyba zakłócona - wartość 1) nie mają wpływu na wartość, jaką filtr przekazuje na swoim wyjściu. Filtr medianowy bardzo skutecznie zwalcza wszelkie lokalne szумy, nie powodując ich „rozmażywania” na większym obszarze, co jest niestety przypadłością wszystkich filtrów konwolucyjnych. Na rysunku 3.82 pokazano jak to się dzieje, posługując się dla zwiększenia poglądowości jednowymiarowym modelem filtrowanego sygnału (wysokości zaznaczonych na rysunku słupków symbolizują wartości stopnia jasności przetwarzanych pikseli).



Rys. 3.82. Usuwanie zakłóceń filtrem medianowym (a) i filtrem uśredniającym (b).

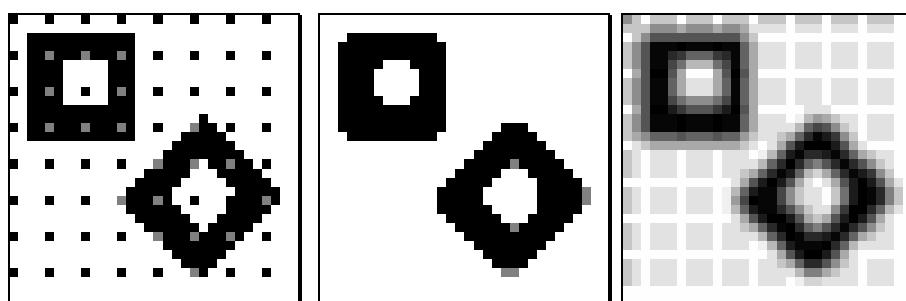


Rys. 3.83. Wpływ filtru medianowego (a) i filtru uśredniającego (b) na brzegi obiektu.

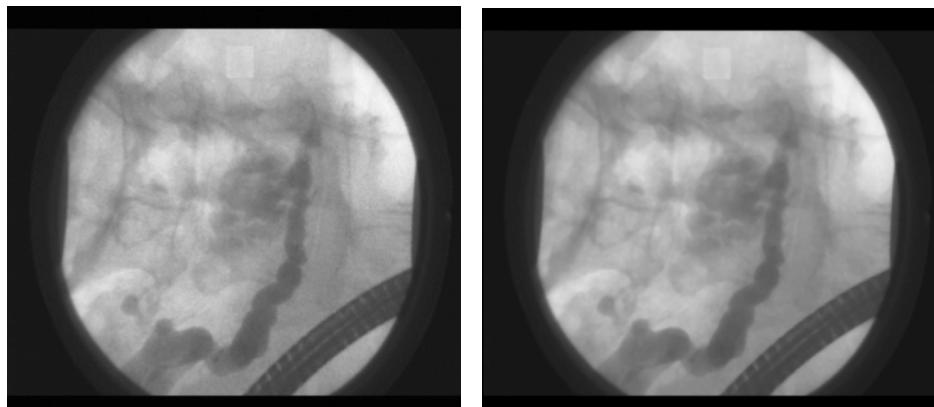
Filtracja medianowa nie wprowadza do obrazu nowych wartości, obraz po wykonaniu filtracji nie wymaga więc żadnego dodatkowego skalowania, co także jest

pewną zaletą. Najważniejszy atut filtracji medianowej polega jednak na tym, że **na ogół** nie powoduje ona pogorszenia ostrości krawędzi obecnych na filtrowanym obrazie poszczególnych obiektów. Ilustruje to (znowu na modelu jednowymiarowym) rysunek 3.83. Widać, że uśrednianie (charakterystyczne dla filtracji konwolucyjnych) produkuje sztuczne pośrednie poziomy jasności pomiędzy całkowitą czernią i całkowitą bielą - mediana natomiast tego nie robi.

Jak wynika z omówionych właściwości - filtr medianowy ma sporo zalet. Usuwa dość skutecznie zakłócenia, nie niszczy obiektów - filtry niemal idealnie. Wydaje się taką opinię potwierdzać rysunek 3.84, na którym pokazano wynik filtracji medianowej i konwolucyjnej tego samego sztucznego, sztucznie zaszmionego obrazu.



Rys. 3.84. Sztuczny obraz (po lewej) po filtracji medianowej (pośrodku) i konwolucyjnej (po prawej).

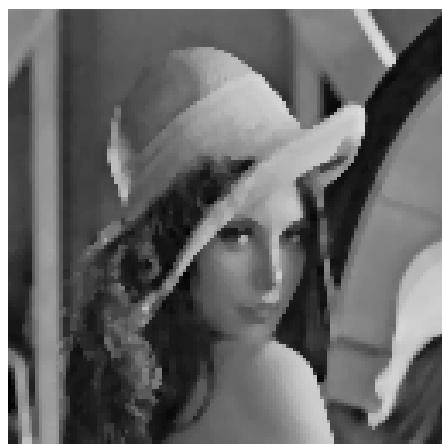


Rys. 3.85. Obraz medyczny - z lewej oryginalny, z prawej polepszony metodą filtracji medianowej.

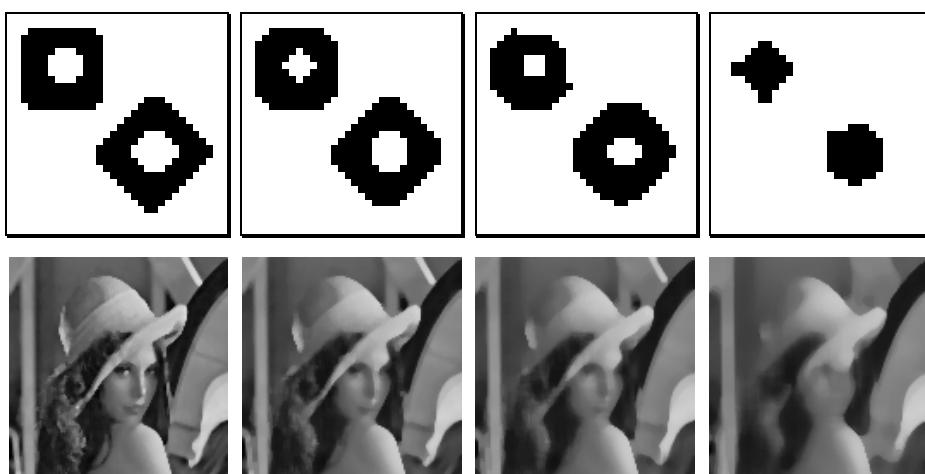
Zalety filtracji medianowej, niewątpliwie jednej z najlepszych technik filtracji obrazu, można też prześledzić na przykładzie rzeczywistych obrazów medycznych (związanych ze wspominaną już wcześniej techniką endoskopowej wstecznej cholangio-pakreatografii czyli ERCP). Na rysunku 3.85 pokazano oryg-

ginalny obraz rentgenowski, będący bezpośrednim wynikiem badania pacjenta oraz ten sam obraz po wykonaniu na nim filtracji medianowej. Technika druku może nie ujawnić tego z całą dobitnością - jednak przy bezpośrednim oglądaniu obrazów na ekranie przetwarzającego komputera różnica jakości jest uderzająca - oczywiście na korzyść obrazu po filtracji.

Z przytoczonych uwag można by wyciągnąć wniosek, że technika medianowa jest tą właśnie idealną filtracją, o której zawsze marzyliśmy. Nie wpadajmy jednak w przesadny entuzjazm. Mediana też trochę psuje obraz, co można przy odrobinie uwagi zauważyc nawet na rysunku 3.84, gdzie ujawniła się typowa dla mediany skłonność do „obgryzania narożników” (swoją drogą - warto się chwilę zastanowić na tym, **dla czego** mediana w taki właśnie sposób psuje obrazy?). Jeszcze lepiej zaobserwujemy to na przykładzie obrazu „Lena” (rys. 3.86).



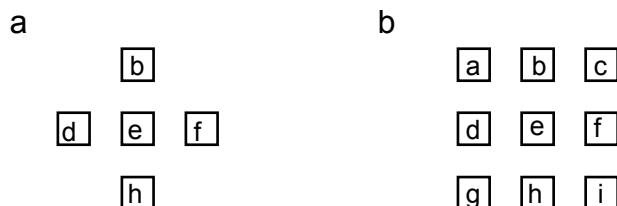
Rys. 3.86. „Lena” po filtracji medianowej.



Rys. 3.87. Obrazy po filtracji medianowej w oknach o coraz większym rozmiarze.

Niszczycielskie właściwości mediany można prześledzić dokładniej, gdy zastosuje się ten rodzaj filtracji z powiększającym się oknem. Na rysunku 3.87 pokazano skutki filtracji medianowej dla okien o rozmiarach kolejno  $3 \times 3$ ,  $5 \times 5$ ,  $7 \times 7$  i  $9 \times 9$  pikseli.

Erosja obrazu, jaka jednak nieuchronnie ma miejsce podczas filtracji medianowej nie jest jedyną wadą tej techniki. Kolejnym jej mankamentem jest długi czas obliczeń konieczny do tego, by cały obraz poddać filtracji zgodnie z tym algorytmem. Dla zmniejszenia obciążenia procesora przetwarzającego obraz podczas filtracji medianowej stosuje się różne zabiegi - najprostszy polega na tym, by zmniejszyć ilość punktów tworzących okno, dla którego wyznaczać trzeba medianę. Dlatego przy filtracji medianowej nagminnie stosuje się okna o ograniczonej liczbie elementów (pięciopunktowe), pokazane na rysunku 3.88.



Rys. 3.88. Otoczenie w medianie pięciopunktowej (a) i dziewięciopunktowej (b).

Inną techniką zwiększenia szybkości znajdowania mediany jest stosowanie specjalnych wzorów, pozwalających na obliczenie wartości mediany bez konieczności uciążliwego sortowania elementów okna. Niżej podano taki wzór dla przypadku okna pięciopunktowego.

$$\begin{aligned} \text{MED } (b, d, e, f, h) = & \text{MAX} [\text{MIN} (b, d, e), \text{MIN} (b, d, f), \text{MIN} (b, d, h), \\ & \text{MIN} (b, e, f), \text{MIN} (b, e, h), \text{MIN} (b, f, h), \\ & \text{MIN} (d, e, f), \text{MIN} (d, e, h), \text{MIN} (d, f, h), \\ & \text{MIN} (e, d, h)] \end{aligned}$$

Do filtrów nieliniowych zalicza się także inne metody analizujące stopnie szarości wybranego otoczenia punktu, jak ciągi liczb. Wybierając spośród takiego ciągu wartość największą uzyskujemy **filtre maksymalny**. Wybierając wartość najmniejszą - uzyskujemy **filtre minimalny**. Zasadę działania tych filtrów (w porównaniu z szerzej omówioną medianą) ilustruje podany niżej przykład.

3	12	21		$\downarrow$ mediana
4	12	43	$\Rightarrow 1, 3, 4, 8, 12, 12, 21, 43, 100$	
1	8	100	$\uparrow$ filtr minimalny	$\uparrow$ filtr maksymalny

### 3.4.3.4 Filtry adaptacyjne

Rodzina nieliniowych filtrów jest bardzo bogata. Należy do nich wiele funkcji znacznie różniących się właściwościami. Bardziej skomplikowaną grupą pośród filtrów nieliniowych są tzw. *filtry adaptacyjne*. Filtry te zmieniają charakterystykę działania w zależności od cech analizowanego obszaru. Przydatność filtrów adaptacyjnych wynika z faktu, że większość prostych filtrów działa jednostronnie. Na przykład prosty filtr uśredniający oprócz wyrównania intensywności jednolitych płaszczyzn rozmazuje krawędzie, a prosty filtr górnoprzepustowy wzmacniający krawędzie wydobywa niepotrzebne szczegóły z jednolitych obszarów. Adaptacyjny filtr uśredniający działa dwuetapowo:

1. W pierwszym etapie wyznaczany jest parametr klasyfikujący dany punkt do krawędzi. Jako kryterium można przyjąć wariancję stopni szarości w jego otoczeniu. Jeżeli taka wariancja przyjmuje duże wartości, punkt z dużym prawdopodobieństwem należy do krawędzi obszarów.
2. W drugim etapie dokonuje się filtracji filtrem uśredniającym, ale tylko tych punktów, które nie zostały zakwalifikowane do krawędzi. Punkty należące do krawędzi pozostają bez zmian. Unika się w ten sposób ich rozmycia.

Efekt działania takiej techniki filtracji można obejrzeć na rysunku 3.88.

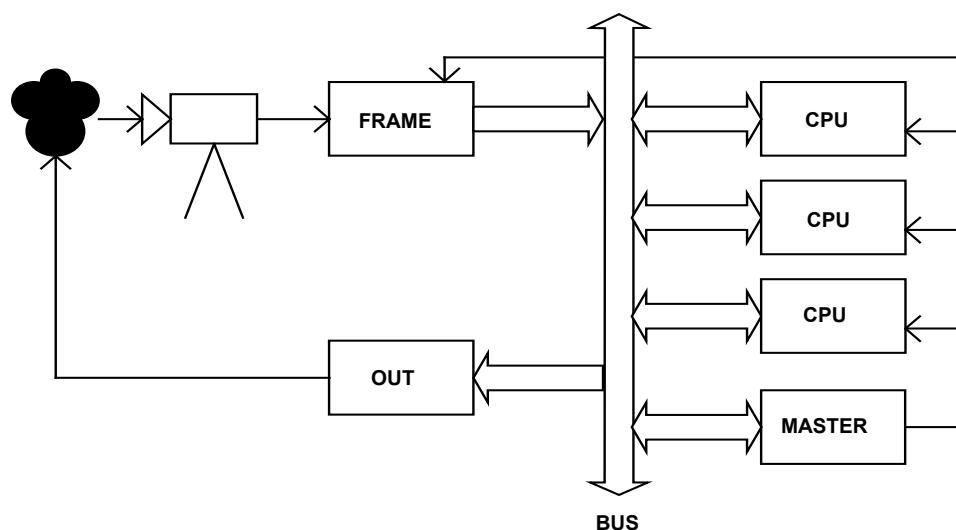


Rys. 3.88. Efekt działania filtra adaptacyjnego.

## 3.5 Realizacja funkcji przetwarzania obrazów przez specjalizowane procesory

### 3.5.1 Struktura specjalizowanego systemu do przetwarzania obrazów

Czasochłonność operacji obliczeniowych związanych z różnymi formami komputerowego przetwarzania obrazów skłania do konstruowania specjalizowanych systemów wieloprocesorowych służących do bardzo szybkiej obróbki wstępnej danych wizyjnych (rys. 3.89).



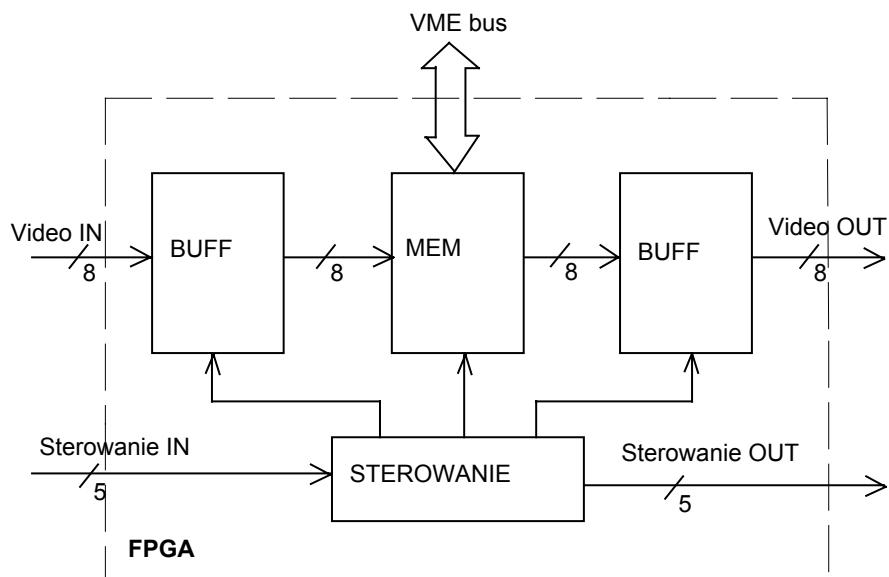
Rys. 3.89. Architektura wieloprocesorowego systemu do analizy obrazów.

W systemach tego typu kluczową rolę odgrywają specjalizowane procesory (oznaczone na rysunku 3.89 jako CPU), które służą do szybkiego, w pełni sprzętowego przetwarzania obrazu. Niżej omówimy kilka przykładów takich konstrukcji.

### 3.5.2 Procesory operacji punktowych

Zgodnie z kolejnością omawiania różnych operacji wykonywanych na obrazach zacząć należy od najprostszych procesorów, wykonujących operacje wynikające z punktowych przekształceń obrazów. Służy do tego zazwyczaj (jeśli decydujemy się na realizację tych operacji techniką sprzętową) procesor *look-up-table* realizujący funkcję przekodowania według tablicy. Jak wiadomo z rozdziału 3.3. jest to funkcja bardzo przydatna na różnych etapach wstępnego przetwarzania. Praktyczna implementacja procesora *look-up-table* została przedstawiona na rysunku 3.90.

Służąca do przekodowania według zasady *look-up-table* pamięć oznaczona na rysunku jako MEM może być typu ROM lub RAM. Zapamiętanie tablicy przekodowań w pamięci typu ROM znacznie przyspiesza szybkość przetwarzania i ogranicza znacznie ilość użytych zasobów. Wówczas zawartość tablicy przekodowań ustawiana jest jednorazowo podczas programowania konfiguracji układu FPGA i ewentualna jej zmiana wymaga reprogramowania całego układu (czas około 2,5 sekundy). Zastosowanie natomiast implementacji tablicy przekodowań w pamięci typu RAM pozwala na dynamiczne (w trybie on-line) zmiany tablicy z poziomu magistrali VME bez konieczności reprogramowania całego układu. Wpis tablicy do RAM może nastąpić w trakcie przerwy w przepływie danych przez potok (np. w czasie trwania impulsów synchronizacji pionowej). Jest to rozwiązanie nieco wolniejsze i wnosi opóźnienie odpowiadające czasowi trwania dwu pikseli. W takich przypadkach logika procesora przetwarza wchodzące do modułu procesora kolejne bajty jednocześnie z tymi, które nie zdążyły go jeszcze opuścić.



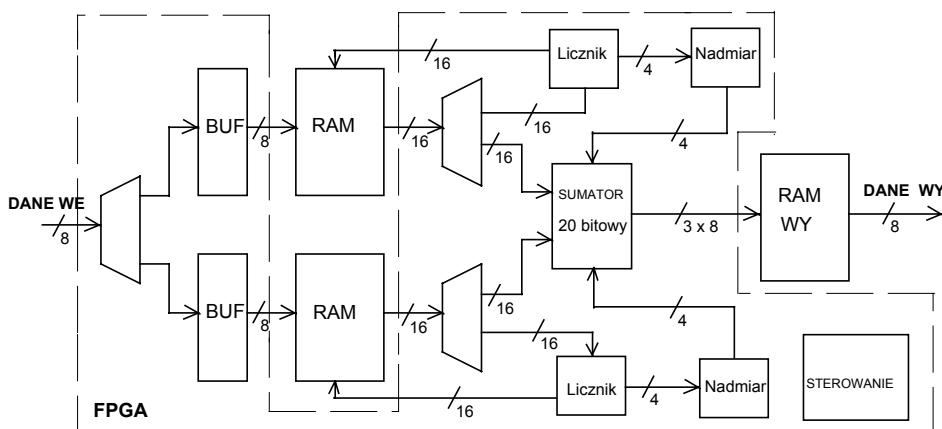
Rys. 3.90. Schemat procesora "look-up-table".

Na rysunku 3.90. zaznaczono także układ sterowania, generujący szereg sygnałów dla procesora *look-up-table*, ale także opóźniającego impulsy sterujące potoku. Szczególnym przypadkiem przekodowania *look-up-table* jest binaryzacja. Ograniczenie liczby bitów informacyjnych na wyjściu tej operacji powoduje pewne uproszczenia prowadzące do zwiększenia szybkości pracy procesora oraz możliwości zastosowania mniejszego układu FPGA. Rozwiązanie takie (w postaci odzielnego od LUT procesora binaryzacji) jest kuszące ze względów ekonomicznych, niesie jednak pewne ograniczenia elastyczności całego systemu.

### 3.5.3 Procesor obliczający histogram

Kolejnym (pod względem złożoności struktury) elementem realizującym sprzętowe przetwarzanie obrazu jest procesor obliczający histogram. Liczenie histogramu nie jest funkcją przekształcającą obraz źródłowy (pozostaje on nie zmieniony), pełni jednak ważną rolę w procesie związanym z wieloma formami przetwarzania obrazu (na przykład z binaryzacją). Omawiany tu procesor (rysunek 3.91) wylicza histogram, jako zestaw parametrów charakteryzujących obraz, przydatnych dla potrzeb dalszego przetwarzania.

Zasada działania tego procesora jest prosta. W miarę napływania kolejnych wartości poszczególnych pikseli, następuje ich zliczanie w bardzo szybkich pamięciach statycznych. Wartość odpowiadająca danemu poziomowi szarości jest adresem komórki pamięci zliczającej ilość pikseli o takiej właśnie wartości - należy odczytać tę komórkę, zwiększyć jej wartość o jeden i wpisać ponownie do pamięci. Pamięć musi zostać oczywiście wyzerowana przed rozpoczęciem zliczania.



Rys. 3.91. Schemat blokowy procesora liczącego histogram.

### 3.5.4 Procesory konwolucji

#### 3.5.4.1 Ogólna struktura

Mianem "procesor konwolucji" określono tu procesor realizujący liniowe funkcje kontekstowe. Założono, że procesor konwolucji ma pracować w potoku obróbczym na strumieniu danych video z kamery wizyjnej. Procesor ten oblicza i wstawia nową wartość punktu obrazowego w miejsce wartości dotychczasowej. Nowa wartość punktu jest znormalizowaną sumą dotychczasowej wartości punktu i jego otoczenia, przemnożonych przez zestaw współczynników wagowych (nazywanych w podrozdziale 3.4.2 maskami), zadawanych programowo. Konwolucja jest przekształceniem łącznym, to znaczy:

$$(f*g)*h = f*(g*h)$$

W praktyce pozwala to, przy odpowiednim doborze współczynników, na podział dużych macierzy filtrów na mniejsze części i filtrację w kilku przebiegach. Poniżej pokazano, na przykładzie wyrażenia macierzy konwolucji o rozmiarach  $5 \times 5$  za pomocą dwóch macierzy o rozmiarach  $3 \times 3$ , na czym taka łączność polega. Dzięki konsekwentnemu stosowaniu podanych zależności można z pomocą układów sprzętowych realizujących konwolucję uzyskiwać - zależnie od potrzeb - filtry o różnej wielkości obszaru, w którym dokonuje się operacji kontekstowych.

1	0	0	0	0
0	1	0	0	0
0	0	1	0	0
0	0	0	1	0
0	0	0	0	1

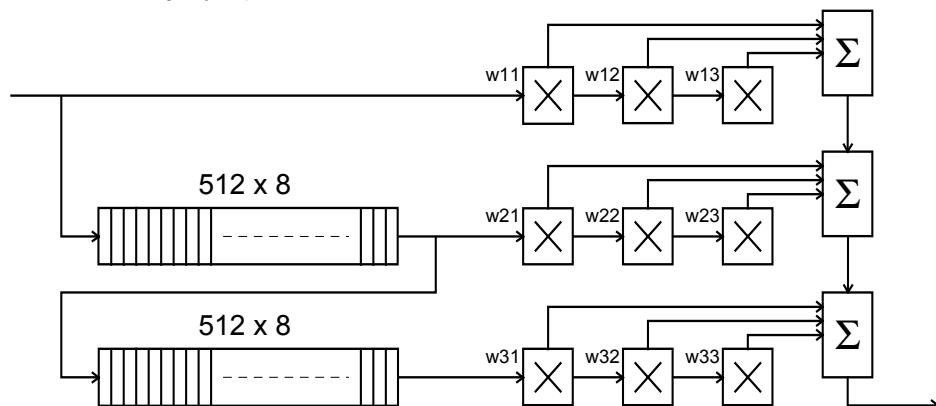
↔

1	0	0
0	1	0
0	0	1

,

0.5	0	0
0	0.33	0
0	0	0.5

Do realizacji funkcji konwolucji w czasie rzeczywistym procesor potokowy musi być wyposażony w linie opóźniające, zapewniające ciągły przepływ potoku danych przy jednoczesnym dostępie do pełnego otoczenia analizowanego punktu (rys. 3.92). Wnosi to nie dające się eliminować opóźnienie równe czasowi trwania dwóch linii. Dokładność operacji kontekstowych zależy od wielkości analizowanego otoczenia punktu (np.  $3 \times 3$ ,  $5 \times 5$ ,  $7 \times 7$ ). Wiąże się to z wyposażeniem procesora w odpowiednią liczbę linii opóźniających (odpowiednio: 2, 4, 6) i układów arytmetycznych (odpowiednio: 9, 25, 49 układów mnożących i 3, 5, 7 układów sumujących).



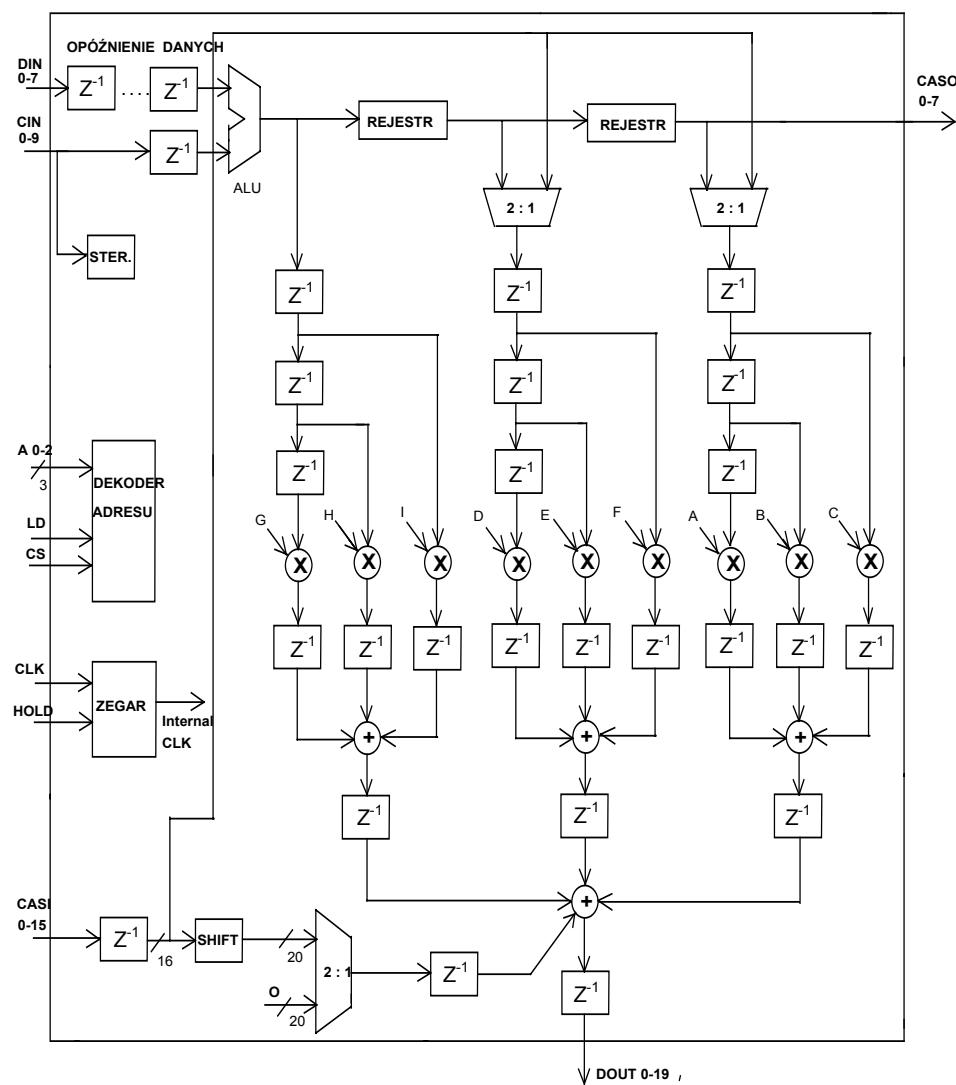
Rys. 3.92. Struktura procesora do realizacji funkcji kontekstowych.

W ostatnich latach, z uwagi na szybki rozwój dziedziny cyfrowej obróbki sygnałów, gwałtownie rośnie liczba komercyjnie dostępnych rozwiązań scalonych

procesorów konwolucji. W chwili obecnej istnieje już co najmniej kilka typów specjalizowanych procesorów sygnałowych, przeznaczonych wyłącznie do wykonywania tej operacji. Poniżej krótko scharakteryzowane zostaną trzy wybrane procesory konwolucji, pochodzące od różnych producentów.

### 3.5.4.2 Procesor HSP-48908

Procesor konwolucji HSP-48908 firmy Harris stanowi stosunkowo ubogie rozwiązanie, jeśli chodzi o rozmiar aktywnego okna obróbki obrazu (rys. 3.93).

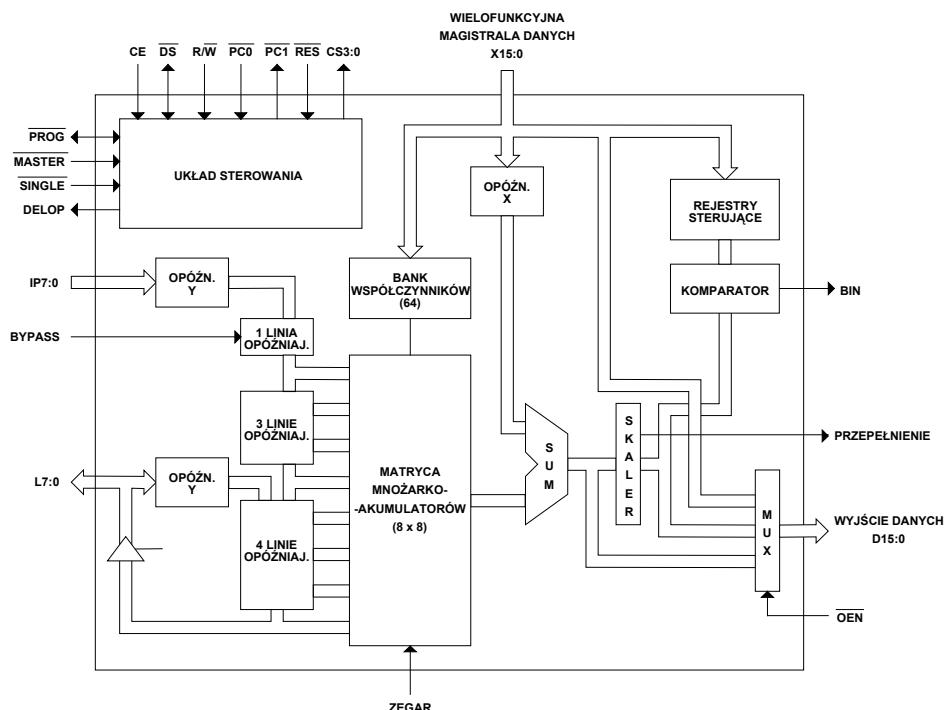


Rys. 3.93. Struktura blokowa procesora HSP48908.

Przy zastosowaniu pojedynczego układu można zaimplementować okno o rozmiarze  $3 \times 3$  piksele. Istnieje możliwość kaskadowego łączenia układów, ale już dla okna o rozmiarze  $5 \times 5$  pikseli należałoby zastosować aż 4 takie jednostki. W przypadku tego procesora odpowiedni sposób kaskadowania umożliwia również obróbkę obrazów o większej niż 1024 piksele długości linii (wynikającej z maksymalnej długości wewnętrznych cyfrowych linii opóźniających), co jednak rzadko ma zastosowanie w praktyce. Natomiast szczególnie ciekawą cechą tego procesora jest możliwość poddania strumienia danych wejściowych jednej z 19-tu różnych, programowalnych operacji arytmetyczno-logicznych oraz dodatkowo normalizacji danych (przesunięciu danych maksymalnie do 3 bitów w lewo lub prawo). Drugim argumentem operacji arytmetyczno-logicznych są stałe dla danego ciągu próbek zestawu wartości współczynników wag zaprogramowane wcześniej w rejestrze ALU przy pomocy zewnętrznego procesora nadzorowanego.

### 3.5.4.3 Procesor PDSP-16488

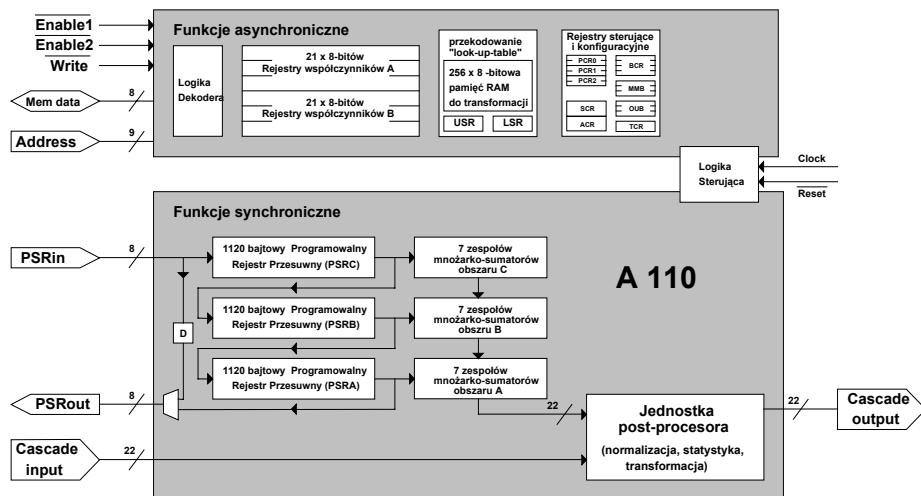
Całkowicie odmienne rozwiązanie architektury zastosowano w procesorze PDSP-16488 firmy Plessey. Architektura wewnętrzna procesora PDSP-16488 jest przedstawiona na rys. 3.94.



Rys. 3.94. Architektura wewnętrzna procesora PDSP16488.

Jest to jednostka bardzo dobrze nadająca się do implementacji okien o dużych rozmiarach (aż do  $8 \times 8$  w pojedynczym układzie) i do obróbki danych wejściowych o dużej dokładności (do 16 bitów). Architektura tego układu jest zaprojektowana w sposób bardzo elastyczny, umożliwiający dostosowanie w szerokich granicach do aktualnych potrzeb użytkownika. Jednostka ta posiada jednak szereg ograniczeń. Ze względu na fakt, iż największą część powierzchni struktury tego typu układów zajmują cyfrowe linie opóźniające, nie jest możliwe równoczesne wykorzystanie pełnej długości linii opóźniających (1024 piksele) i pełnego rozmiaru okna ( $8 \times 8$ ). Z kolei szczególny sposób wykonania bloku mnożarek, polegający na wielokrotnym wykorzystaniu każdej mnożarki podczas jednego okresu zegara danych, przy zwiększeniu dokładności do 16 bitów, implikuje zmniejszenie długości wewnętrznych linii opóźniających i rozmiaru okna. Powoduje to także zmniejszenie maksymalnej prędkości przepływu strumienia danych przez układ. Podobnie jak układ HSP48908, tak i ten posiada możliwość kaskadowego łączenia kilku jednostek. Układ ten posiada dodatkowo normalizator danych (układ przesunięć) oraz prosty, dwustanowy komparator danych wyjściowych.

### 3.5.4.4 Procesor IMS-A110



Rys. 3.95. Architektura wewnętrzna procesora IMS A110.

Procesor konwolucji IMS-A110 firmy INMOS stanowi jedno z najwcześniejszych rozwiązań problemu implementacji konwolucji w strukturze hardware'owej. Architektura wewnętrzna jest przejrzysta, a bardzo cennym jest umieszczenie w strumieniu danych wyjściowych bogato wyposażonego tzw. *postprocesora*, którego szerokie możliwości ułatwiają w znacznym stopniu zastosowanie tej jednostki w wielu różnorodnych aplikacjach. Zasadniczy człon procesora składa

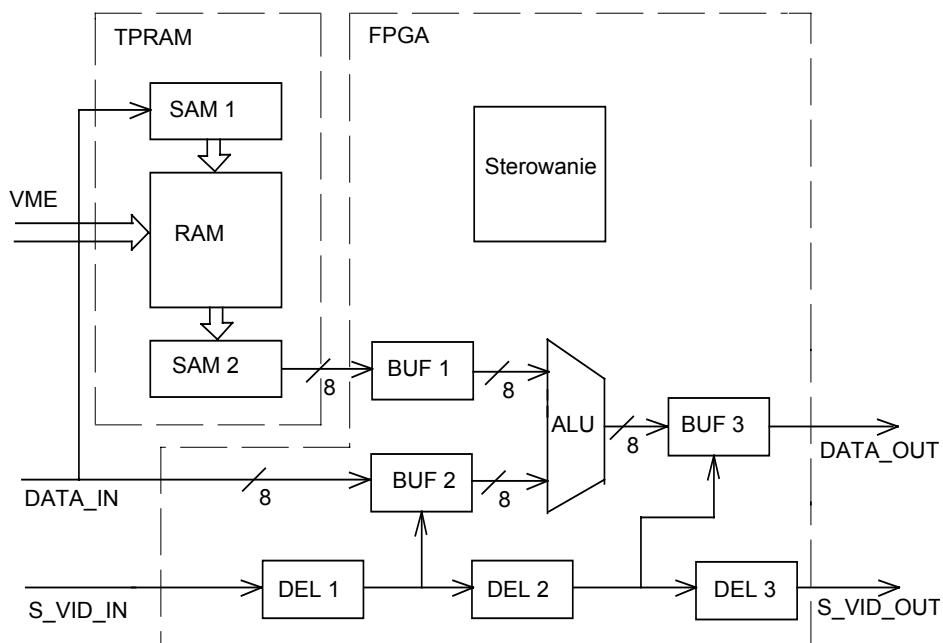
się z trzech linii opóźniających o długości 1120 bajtów oraz trzech siedmio-stopniowych mnożarko-sumatorów. Jest on wyposażony (tak jak poprzednio omówione układy) w możliwość kaskadowego łączenia z innymi tego typu jednostkami w celu zwiększenia rozmiaru aktywnego okna lub zwiększenia dokładności danych i/lub współczynników konwolucji (rys. 3.95).

Na szczególną uwagę zasługuje wspomniany wcześniej układ postprocesora. Składa się on z czterech bloków: bloku wejściowego (rejestr przesuwny, sumator kaskadowy), bloku monitora statystycznego (detektor wartości minimalnej i maksymalnej oraz komparator z rejestrami przepełnienia/niedoboru), bloku formowania danych (układ transformacji i układ normalizacji danych) oraz bloku wyjściowego (sumatora i multipleksera wyjściowego).

Dodatkowo procesor A110 wyposażony jest w ośmiorozbitowy interfejs mikroprocesora, służący do konfiguracji układu przez zapisanie odpowiednich danych sterujących do rejestrów wewnętrznych. Przestrzeń pamięciowa procesora obejmuje 512 lokacji adresowych.

### 3.5.5 Moduł procesora logicznego

Procesor logiczny realizuje funkcje logiczne w oparciu o wbudowaną jednostkę arytmetyczno-logiczną (ALU) - rys. 3.96. Oczywiście jednostka arytmetyczno-logiczna może realizować dowolne proste funkcje logiczne na dwu obrazach, przy czym piksele są reprezentowane przez 8-bitów.



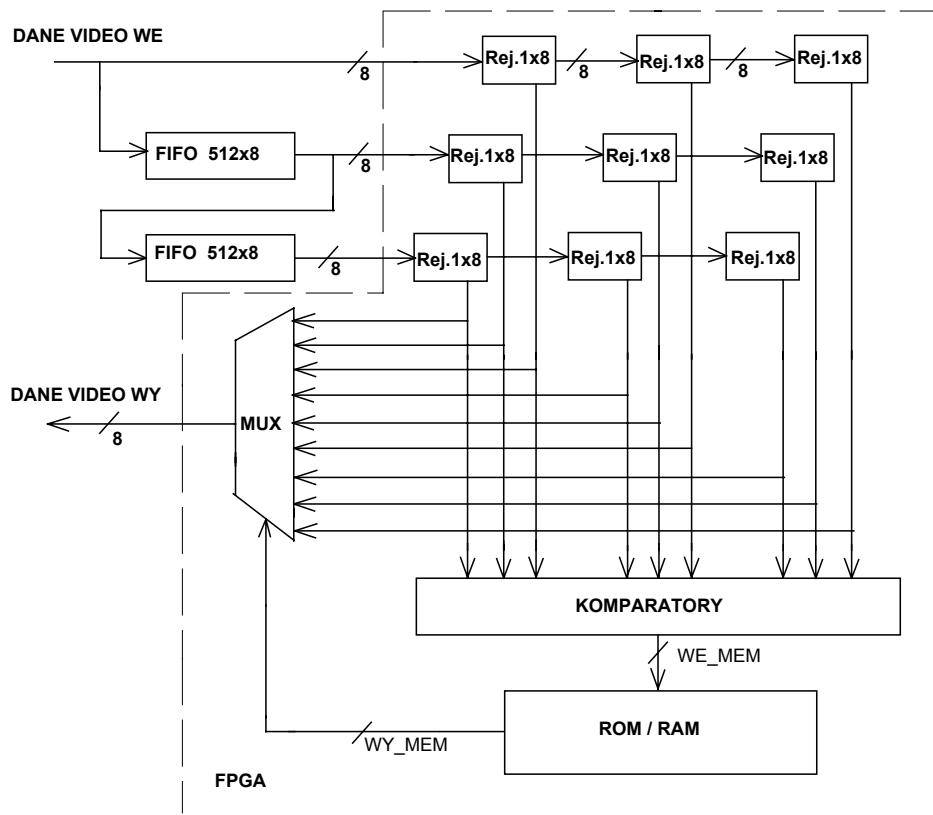
Rys. 3.96. Blokowa budowa procesora logicznego.

### 3.5.6 Procesor filtracji medianowej

Ważną operacją dla wstępного przetwarzania danych wizyjnych jest filtracja medianowa przedstawiona w rozdziale 3.4.3.

Zadaniem filtracji medianowej jest obliczenie wartości środkowej obrabianego punktu i jego otoczenia. Moduł procesora filtracji medianowej (procesora medianowego) musi zawierać:

- linie opóźniające, zapewniające jednoczesny dostęp do koniecznego otoczenia punktu,
- zestaw komparatorów, porównujących wartości poszczególnych punktów,
- multiplexer wyjściowy,
- interface magistrali potokowej.



Rys. 3.97. Blokowa budowa procesora medianowego.

W praktyce oznacza to wyposażenie modułu procesora medianowego w dwie linie opóźniające o długości 512 (jedna linia obrazu) i szerokości 8 bitów (256 po-

ziomów szarości). Dla zapewnienia jednoczesnego dostępu do całego otoczenia punktu dla mediany dziewięciopunktowej moduł procesora musi zawierać 9 rejestrów 8 bitowych, których wyjścia są doprowadzone do zestawu komparatorów oraz do multiplexera (odpowiednio dla mediany pięciopunktowej będzie to 5 rejestrów 8 bitowych) - rys. 3.97.

Z punktu widzenia zastosowania procesora medianowego jako elementu obróbki potokowej w czasie rzeczywistym ważna jest szybkość działania opracowanego hardware'u. Szybkość ta mierzona jest czasem, jaki upływa od momentu ustalenia się ważnych danych na wejściu do momentu pojawięcia się ważnych danych na wyjściu. Limit tak wyrażonego opóźnienia podyktowany jest czasem trwania jednego punktu obrazowego (piksela) i jest bardzo krótki. Wymaga to użycia równoległej struktury przetwarzania danych. Przykładowo - dla mediany 9-punktowej jednocześnie porównywanych jest w bloku komparatorów 45 par liczb 8-bitowych (dla mediany 5-punktowej 10 komparatorów porównuje pary: 1-2, 1-3, 1-4, 1-5, 2-3, 2-4, 2-5, 3-4, 3-5, 4-5).

## **4 Transformacja Fouriera**

### **4.1 Wprowadzenie**

W dziedzinie przetwarzania sygnałów jednowymiarowych transformacja Fouriera jest niezaprzeczalnie narzędziem fundamentalnym. Dla sygnałów cyfrowych stosowana jest jej zmodyfikowana wersja zwana Dyskretną Transformacją Fouriera (w skrócie: DTF). Transformacja Fouriera wiąże się bardzo ściśle z pojęciem częstotliwości. Powszechnie stosowanie transformacji Fouriera jako skutecznego narzędzia w różnych dziedzinach - w szczególności w akustyce - powoduje, że często przyjmuje się, iż przydatność tej transformacji jest uniwersalna. O ile częstotliwość dźwięku jest dla naszego zmysłu słuchu informacją naturalną i bardzo użyteczną, o tyle zmysł wzroku opiera się raczej bezpośrednio na kształcie obiektu na obrazie. Dlatego też w zdecydowanej większości problemów z zakresu przetwarzania obrazów interpretacja częstotliwościowa jest mało przydatna. Można naturalnie przyjąć, że krawędź (niosąca zwykle w obrazie istotną informację) odpowiada obecności wysokich częstotliwości, a obszary o stałym kolorze zawierają głównie częstotliwości niskie. Taka interpretacja jest poprawna - niewiele jednak z niej wynika dla rozwiązymania konkretnych problemów. Także i inne elementy częstotliwościowej interpretacji obrazu niewiele wnoszą merytorycznie do technik przetwarzania obrazów. W rozdziale tym zostanie pokazane na przykład, jaki związek z reprezentacją częstotliwościową mają znane ze swej skuteczności filtry splotowe. Chociaż można wykazać istnienie tego związku, to można też pokazać, że nie jest on szczególnie przydatny. Podobne przykłady można mnożyć - trudno jest określić użyteczny związek pomiędzy informacją wizualną zawartą w obrazie i poszczególnymi częstotliwościowymi składowymi jego opisu w dziedzinie Fouriera. W zdecydowanej większości zagadnień metody abstrahujące całkowicie od interpretacji częstotliwościowej dają zdecydowanie lepsze rezultaty - skuteczność filtrów splotowych można przecież uzasadnić bez odwoływanego się do transformacji Fouriera.

Czytelnik może zatem zadać pytanie: dlaczego w takim razie zadano sobie tyle trudu i umieszczono w książce o przetwarzaniu obrazów obszerny rozdział o transformacji Fouriera? Odpowiedź jest prosta - ponieważ ktoś może znaleźć, aczkolwiek nie jest to łatwe, nowe, pozyteczne zastosowanie tej transformacji do przetwarzania obrazów. Do tego jednak, by było to naprawdę użyteczne zastosowanie, należy najpierw dokładnie zrozumieć, na czym polegają istotne cechy transformacji Fouriera w odniesieniu do obrazów. Wyszczególnienie w tym rozdziale wybranych cech DTF w zastosowaniu do obrazów ma ustrzec Czytelnika

przed przeoczeniem którejś z nich przy projektowaniu metody przetwarzania obrazów opartej na interpretacji częstotliwościowej.

Jak podkreślono, w przetwarzaniu obrazów uwypuklają się inne aspekty DTF (tym razem w wersji dwuwymiarowej) niż w przypadku sygnałów jednowymiarowych, pomimo tego iż z matematycznego punktu widzenia obraz cyfrowy może być traktowany jako rozszerzenie pojęcia cyfrowego sygnału jednowymiarowego do dwóch wymiarów. Teoretyczne uzasadnienie wzorów DTF oraz dowody właściwości transformacji zainteresowany Czytelnik może znaleźć w licznych podręcznikach z zakresu cyfrowego przetwarzania sygnałów. Z drugiej strony w wielu z nich, nawet jeśli prezentuje się rozszerzenie DTF do dwóch wymiarów, to nie podkreśla się dostatecznie wyraźnie specyfiki zastosowań tej transformacji do obrazów cyfrowych. Z tych powodów skoncentrujemy się na cechach, które mają szczególne znaczenie z punktu widzenia zastosowań DTF właśnie do przetwarzania obrazów. Wydaje się zatem celowe, aby Czytelnik, nawet z dobrą znajomością zastosowań transformacji Fouriera dla jednowymiarowych sygnałów cyfrowych, nie pomijał tego rozdziału. Ponieważ jednocześnie zakładamy, iż Czytelnik nie musi mieć żadnego przygotowania z zakresu cyfrowego przetwarzania sygnałów, więc na początek dokonamy wprowadzenia - w oparciu o sygnały jednowymiarowe - podstawowych, niezbędnych w dalszym wyjaśnianiu pojęć.

## 4.2 Transformacja Fouriera dla cyfrowych sygnałów jednowymiarowych

Cyfrowy sygnał jednowymiarowy można przedstawić jako uporządkowany i ponumerowany ciąg liczb. Ich wartości, z powodu ograniczonej precyzji procesorów, należą do pewnego dyskretnego zbioru, będącego podzbiorem liczb wymiernych. Dlatego też nazywa się ten sygnał cyfrowym, a nie np. sygnałem z czasem dyskretnym, co miałoby miejsce, gdyby był to ciąg liczb pochodzących z próbkowania sygnału ciągłego, przyjmujących wartości odpowiadające próbkomu sygnałowi z dowolną dokładnością. Można naturalnie zakładać, że numery elementów ciągu mają bezpośredni związek z równo oddalonymi punktami na osi czasu, jednak nie jest to konieczne. Transformacja będzie miała sens również, gdy przeprowadzimy ją dla dowolnego ciągu liczb, które są opatrzone kolejnymi indeksami.

Przykładami cyfrowych sygnałów jednowymiarowych, użytecznych w dalszych rozważaniach, mogą być:

ośmioelementowy ciąg pewnych liczb:

$$L = \{l_n : n = 0, 1, 2, \dots, 7\} \quad (4.1)$$

ciąg liczb całkowitych o parzystej liczbie elementów:

$$L = \{-3,5,1,-2,2,-3,-1,3\} \quad (4.2)$$

ciąg liczb całkowitych zawierających nieparzystą liczbę elementów:

$$L = \{-3,5,1,-2,2,-3,-1\} \quad (4.3)$$

W dalszych rozważaniach będziemy zakładać, że długość ciągu wynosi ogólnie  $N$ . Wówczas indeksy elementów ciągu będą zmieniać się w zakresie od 0 do  $N-1$ . Transformacją Fouriera<sup>22</sup> tego ciągu nazywa się następujące odwzorowanie:

$$f_k = \beta_L \cdot \sum_{n=0}^{N-1} l_n \cdot \exp\left(\frac{-j \cdot 2 \cdot \pi \cdot n \cdot k}{N}\right) \quad \text{dla } k = 0,1,\dots,N-1 \quad (4.4)$$

gdzie  $j = \sqrt{-1}$  to jednostka urojona, a  $\beta_L$  to współczynnik, którego rola zostanie wyjaśniona kilka linijek niżej.

Jako wynik transformacji Fouriera otrzymujemy inny ciąg o tej samej długości, o wartościach zespolonych:

$$F = \{f_k : k = 0,1,\dots,N-1\} \quad (4.5)$$

Dla przykładowego ciągu (4.2) będzie to ciąg:

$$\begin{aligned} F = \{ & 2; 4,1924 - j \cdot 4,1213; -1 - j \cdot 1; -14,1924 - j \cdot 0,1213; \\ & -4; -14,1924 + j \cdot 0,1213; -1 + j \cdot 1; 4,1924 + j \cdot 4,1213 \} \end{aligned} \quad (4.6)$$

Ciąg (4.6) przedstawiono z dokładnością do czwartego miejsca po przecinku. W praktyce dokładność otrzymanych zespolonych współczynników transformaty według wzoru (4.4) będzie zależała od precyzji reprezentacji liczb w wykorzystanym do obliczeń procesorze. Transformacja Fouriera jest w ogólności przystosowana do przekształcania ciągów o wartościach zespolonych - ciąg rzeczywisty stanowi tu tylko szczególny przypadek. Jednak w praktyce zazwyczaj oryginalny ciąg cyfrowy  $L$  jest właśnie rzeczywisty, to znaczy każdy jego element ma zerową część urojoną.

Transformacja Fouriera jest odwracalna (z dokładnością do precyzji obliczeń) i jej odwrotna postać jest następująca:

<sup>22</sup> Dokładniej jest to Dyskretna Transformacja Fouriera (DTF) - dla odróżnienia od:

- a) Transformacji Fouriera (gdzie zarówno dane pierwotne, jak i po transformacji są określone dla ciągłych wartości zmiennej niezależnej),
  - b) Transformacji Fouriera z Czasem Dyskretnym (gdzie jako transformatę ciągu pierwotnego otrzymuje się w dziedzinie transformacji wartości określone na ciągim zbiorze wartości). Jednak ponieważ nie będzie tutaj rozważana ani postać a) ani b), więc dla uproszczenia używana będzie nazwa: Transformacja Fouriera, rozumiana zawsze jako DTF.
- Terminologicznie rozróżnia się niekiedy transformację - operację wykonywaną na ciągu - od transformaty - wyniku tej operacji. Zatem wzór (4.4) określa transformację, natomiast ciąg  $F$  (4.6) jest transformatą ciągu  $L$  (4.2).

$$l_n = \beta_F \cdot \sum_{k=0}^{N-1} f_k \cdot \exp\left(\frac{j \cdot 2 \cdot \pi \cdot k \cdot n}{N}\right) \quad \text{dla } n = 0, 1, \dots, N-1 \quad (4.7)$$

W tym miejscu warto zwrócić uwagę na dwie cechy:

1. Wzory (4.4) i (4.7) różnią się jedynie znakiem w potędze eksponenty, co oznacza, że:
  - a) własności transformacji (4.4) można łatwo odnieść do transformacji odwrotnej (4.7),
  - b) do wyliczenia wartości opisanych obydwojoma wzorami można stosować niemal identyczne algorytmy.
2. We wzorach (4.4) i (4.7) przed znakiem sumy znajdują się współczynniki:  $\beta_L$  dla transformacji w przód i  $\beta_F$  dla transformacji odwrotnej. Transformacje (4.4) i (4.7) będą wzajemnie odwrotne zawsze, gdy iloczyn obu współczynników będzie równy  $1/N$ :

$$\beta_L \cdot \beta_F = \frac{1}{N} \quad (4.8)$$

Najczęściej stosuje się następujące pary współczynników ( $\beta_L, \beta_F$ ):

$$\left(\frac{1}{N}, 1\right), \quad \left(1, \frac{1}{N}\right) \quad \text{oraz} \quad \left(\frac{1}{\sqrt{N}}, \frac{1}{\sqrt{N}}\right) \quad (4.9)$$

W przykładzie (4.6) oraz w podanych dalej przykładach wykorzystano drugą parę. To, którą parę współczynników  $\beta_L$  i  $\beta_F$  się wybierze, nie wpłynie na poprawność wzoru, może jednak ułatwić interpretację wyników transformacji. Parokrotnie wróćmy jeszcze do tego zagadnienia.

Rzeczywisty ciąg  $L$  można przedstawić jako ważoną sumę pewnych specjalnych ciągów  $L_k$ , które można nazwać ciągami składowymi lub bazowymi:

$$L = \sum_{k=0}^{\frac{N}{2}} a_k \cdot L_k \quad \text{dla } N \text{ parzystego} \quad (4.10)$$

$$L = \sum_{k=0}^{\frac{N-1}{2}} a_k \cdot L_k \quad \text{dla } N \text{ nieparzystego} \quad (4.11)$$

Ciągi te muszą być tej samej długości  $N$ :

$$L_k = \{l_{n,k} : n = 0, 1, \dots, N-1\} \quad (4.12)$$

Biorąc pod uwagę (4.12) wzory (4.10) i (4.11) można rozpisać dla poszczególnych elementów ciągów:

$$l_n = \sum_{k=0}^{\frac{N}{2}} a_k \cdot l_{n,k} \quad \text{dla } N \text{ parzystego} \quad (4.13)$$

$$l_n = \sum_{k=0}^{\frac{N-1}{2}} a_k \cdot l_{n,k} \quad \text{dla } N \text{ nieparzystego} \quad (4.14)$$

Indeks  $k$  określa numer ciągu, a wiąże się to jednocześnie z informacją, jak ten ciąg wygląda. Ciąg  $L_0$  składa się z samych jedynek. Ciąg  $L_k$  zawiera wartości będące wartościami funkcji kosinus o amplitudzie 1, okresie  $T_k$  i przesunięciu w fazie  $\varphi_k$ , tak dobranych, by najlepiej pasowały do ciągu  $L$ :

$$l_{n,k} = \cos\left(\frac{2 \cdot \pi \cdot n \cdot \Delta t}{T_k} + \varphi_k\right) \quad (4.15)$$

We wzorze (4.15)  $\Delta t$  oznacza jednakowy odstęp pomiędzy próbками kosinusoidalnego sygnału ciągłego. Okresy  $T_k$  dla funkcji kosinus mieszą się całkowitą ilość razy w przedziale  $T = N \cdot \Delta t$ , przy czym:

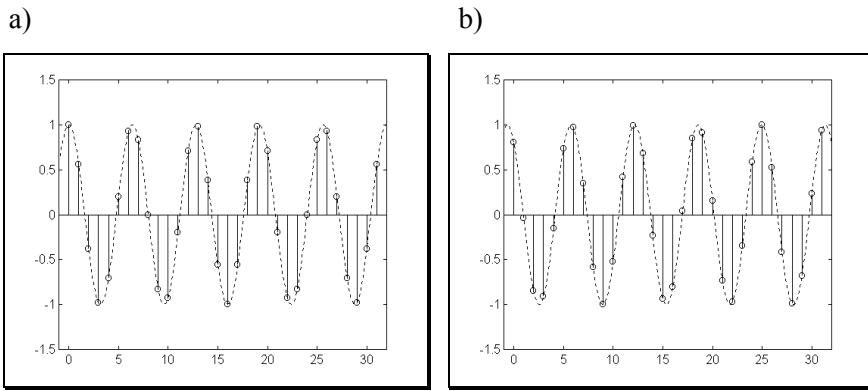
$$T_k = \frac{T}{k} \quad \text{dla } k = 1, \dots, \frac{N}{2}; \quad \text{dla } N \text{ parzystego} \quad (4.16)$$

$$T_k = \frac{T}{k} \quad \text{dla } k = 1, \dots, \frac{N-1}{2}; \quad \text{dla } N \text{ nieparzystego} \quad (4.17)$$

Dla  $k=0$  przyjmuje się funkcję stałą o wartości 1, czyli funkcję kosinus (4.15) o nieskończonym okresie  $T_k$  (zerowej częstotliwości). Naturalne jest więc, że dla  $k=0$  nie ma sensu brać pod uwagę wartości  $\varphi_k$ . Natomiast dla innych  $k$  wybór wartości  $\varphi_k$  ma bardzo duży wpływ na wygląd ciągu  $L_k$ , co ilustruje rysunek 4.1.

Na szczęście nie musimy się martwić, jak dla danego  $L$  dopasowywać  $a_k$  i  $\varphi_k$  - transformacja Fouriera to wyliczy. Ciąg  $F$  przedstawia bowiem informację, jaki jest udział w ciągu  $L$  poszczególnych ciągów  $L_k$  (współczynniki  $a_k$ ) oraz jak dla każdego podciągu przesunięta była jego funkcja kosinus (argumenty  $\varphi_k$ ). Zespolony  $k$ -ty element ciągu  $F$  można wyrazić jako część rzeczywistą i urojoną lub, równoważnie, jako amplitudę i fazę:

$$\begin{aligned} f_k &= A_k \cdot \exp(j \cdot \varphi_k) = \operatorname{re}(f_k) + j \cdot \operatorname{im}(f_k) = \\ &= A_k \cdot \cos(\varphi_k) + j \cdot A_k \cdot \sin(\varphi_k) \end{aligned} \quad (4.18)$$



Rys. 4.1. Przykład sposobu otrzymywania ciągu  $L_k$  z funkcji kosinus, dla  $N=32$  i  $k=5$ :

$$\text{a) faza } \varphi_k = 0, \text{ b) faza } \varphi_k = \frac{1}{5}\pi.$$

Amplituda  $A_k$  jest powiązana bezpośrednio z wartością odpowiedniego współczynnika  $a_k$  we wzorach od (4.10) do (4.14) - patrz zależności (4.19) i (4.20) poniżej.

Dla  $N$  parzystego:

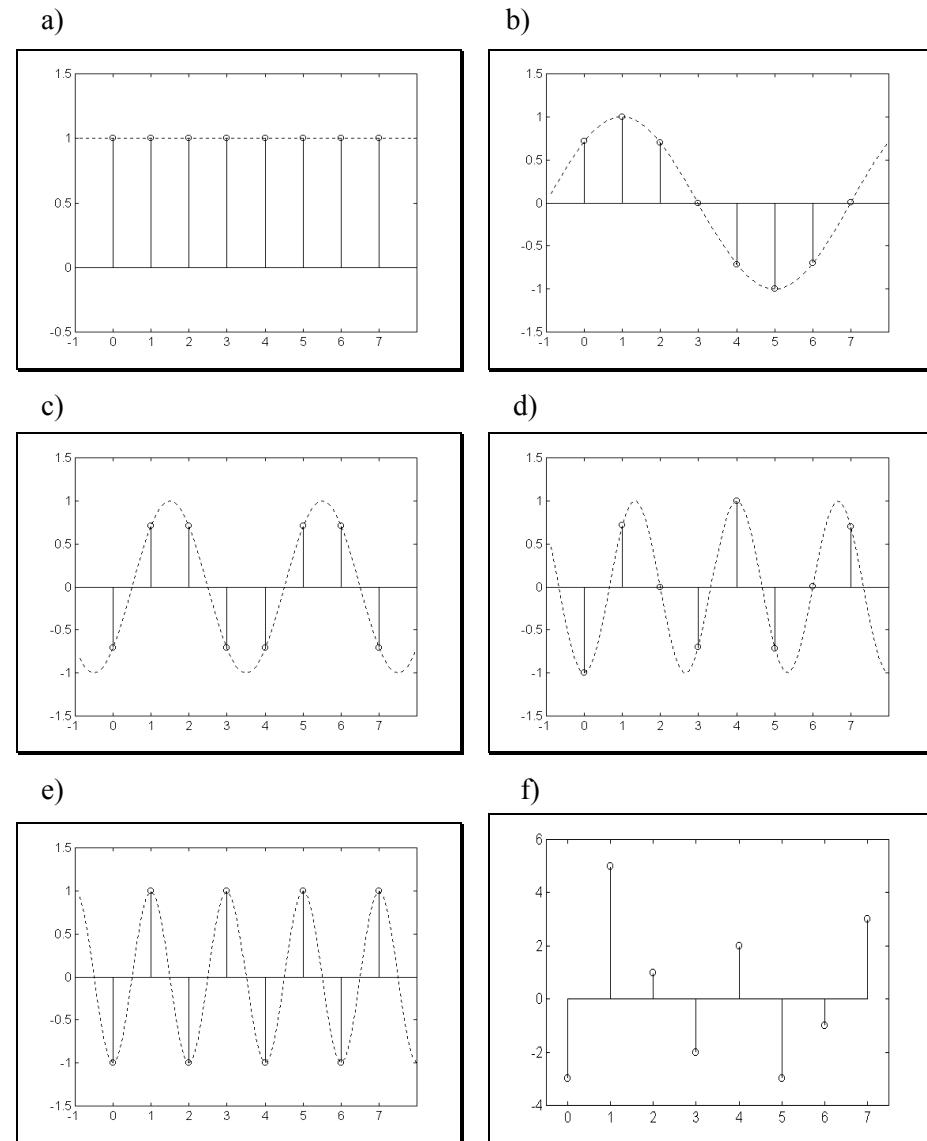
$$A_k = \begin{cases} \frac{1}{2} \cdot a_k \cdot \frac{1}{\beta_F} & \text{dla } 0 < k < \frac{N}{2} \\ a_k \cdot \frac{1}{\beta_F} & \text{dla } k = 0 \text{ i } k = \frac{N}{2} \end{cases} \quad (4.19)$$

Dla  $N$  nieparzystego:

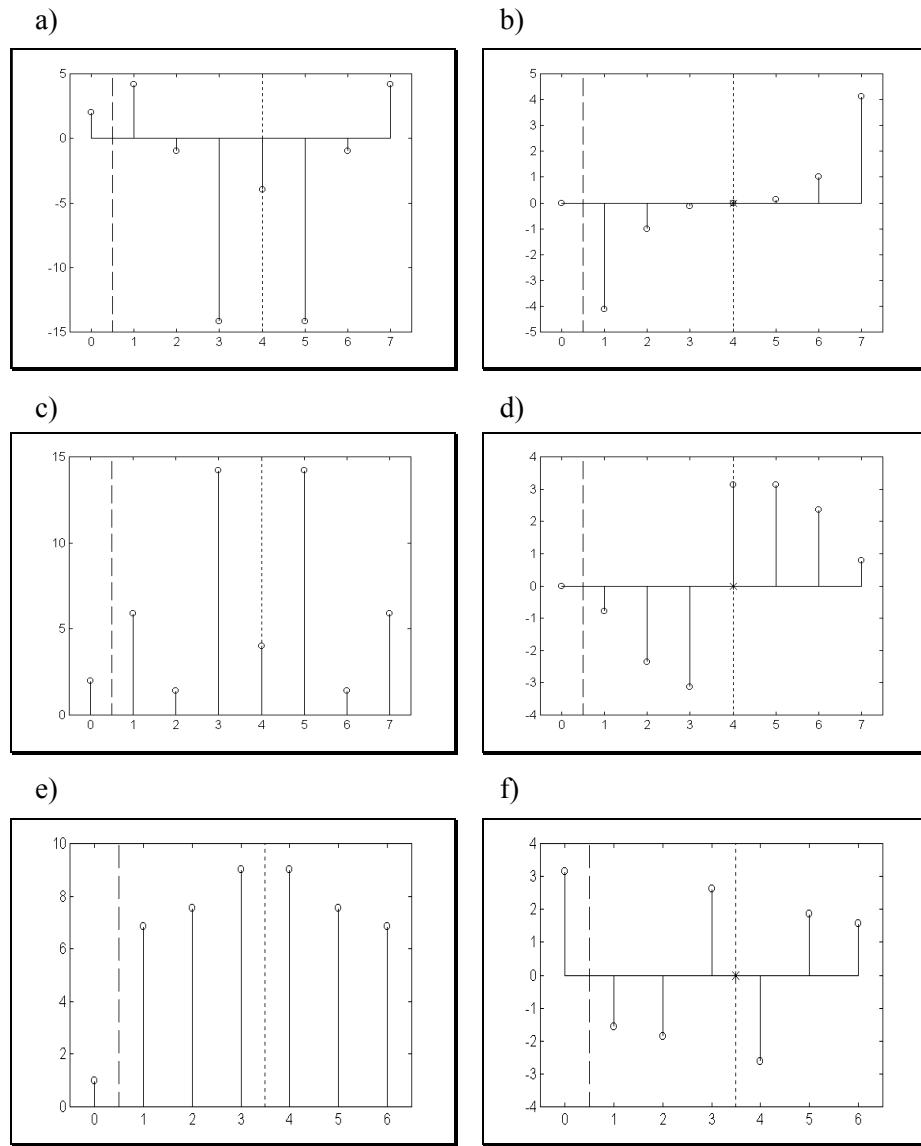
$$A_k = \begin{cases} \frac{1}{2} \cdot a_k \cdot \frac{1}{\beta_F} & \text{dla } 0 < k \leq \frac{N-1}{2} \\ a_k \cdot \frac{1}{\beta_F} & \text{dla } k = 0 \end{cases} \quad (4.20)$$

Natomiast faza  $\varphi_k$  jest tą samą fazą, jaką występuje we wzorze (4.15). Zatem transformacja Fouriera wyznacza rozłożenie ciągu  $L$  na sumę podciągów (4.10) lub (4.11). Warto zwrócić uwagę, że niemal wszystkie współczynniki  $a_k$  pozostają w tej samej proporcji do odpowiednich amplitud  $A_k$ , zależnej od zastosowanego współczynnika  $\beta_F$ . Wyjątek stanowi pierwszy współczynnik (o in-

deksie 0) powiązany z ciągiem bazowym zawierającym same jedynki oraz, w przypadku  $N$  parzystego, współczynnik ostatni (o indeksie  $N/2$ ), powiązany z ciągiem bazowym zawierającym naprzemiennie wartości +1 i -1. Ten ostatni ciąg również nie jest nigdy wiązany z jakimkolwiek przesunięciem fazowym (naturalnie jedynie dla rzeczywistego ciągu  $L$ ).



Rys. 4.2. Reprezentacja ciągu (4.2) za pomocą ciągów  $L_k$ : a) ciąg  $L_0$ , b) ciąg  $L_1$ , c) ciąg  $L_2$ , d) ciąg  $L_3$ , e) ciąg  $L_4$ , f) ciąg oryginalny  $L$  opisany zależnością (4.10) przy użyciu ciągów  $L_k$  i współczynników  $a_k$ .



Rys. 4.3. Symetrie w dziedzinie transformacji Fouriera, gdy ciąg pierwotny nie zawiera ani jednego elementu o części urojonej różnej od zera. Linia przerywana po lewej stronie każdego wykresu przypomina, że element zerowy nie ma udziału w żadnej symetrii.

Linia przerywana w okolicy centralnej przedstawia oś symetrii, gwiazdka na linii centralnej oznacza punkt symetrii (dla części urojonej oraz fazy). Dla  $N$  parzystego w symetriach nie należy również uwzględniać elementu leżącego na centralnej osi symetrii. Ten element musi być zawsze rzeczywisty, więc jego faza będzie zawsze całkowitą - dodatnią lub ujemną - wielokrotnością kąta  $\pi$ . a)  $N$  parzyste - część rzeczywista, b)  $N$  parzyste – część urojona, c)  $N$  parzyste - amplituda, d)  $N$  parzyste - faza, e)  $N$  nieparzyste - amplituda, f)  $N$  nieparzyste - faza.

Interpretacja współczynnika  $A_0$  zależy od przyjętej pary współczynników  $\beta$  (4.9). Dla danego ciągu  $L$  wartość  $a_0$  jest określona i zawsze równa wartości średniej arytmetycznej wszystkich elementów ciągu  $L$ , podczas gdy  $A_0$  stanowi dla

$\beta_L = \frac{1}{N}$  średnią wartość wszystkich elementów ciągu  $L$ , natomiast dla  $\beta_L = 1$   $A_0$  jest sumą wszystkich elementów ciągu  $L$ .

Ponieważ informacja podana w postaci ciągu  $N$  liczb rzeczywistych  $l_n$  jest przekształcana w równoważną, w dziedzinie Fouriera, w postaci  $2N$  liczb -  $N$  amplitud i  $N$  faz - naturalne jest, że musi się w takim wypadku pojawić jakaś nadmiarowość. Faktycznie, gdy ciąg pierwotny jest rzeczywisty, to w dziedzinie Fouriera ciąg zespolony posiada symetrie przedstawione na rys. 4.3. Po pominięciu elementu  $k=0$ , który jest w takim wypadku zawsze rzeczywisty, pozostałe amplitudy (a także części rzeczywiste) elementów ciągu  $F$  są symetryczne względem osi przebiegającej przez punkt środkowy, natomiast fazy (i części urojone) są symetryczne względem punktu przecięcia tej osi z poziomą osią, czyli osią indeksów. Należy zwrócić uwagę na różnicę w położeniu osi dla ciągu o  $N$  parzystym i nieparzystym. Ponadto dla  $N$  parzystego środkowy element, leżący na osi symetrii, jest także zawsze rzeczywisty - wynika to z braku informacji o fazie, o czym już wcześniej wspomniano. Symetrie powyższe nie powstają, gdy ciąg  $L$  zawiera choć jedną liczbę o części urojonej i przynajmniej jedną liczbę o części rzeczywistej różnej od zera - może to być ta sama, jedna liczba zespolona.

Jak widać, rzeczywisty ciąg  $L$  może być dokładnie opisany, gdy znamy połowę<sup>23</sup> elementów ciągu w dziedzinie Fouriera. Jednak przed zastosowaniem wzoru na transformację odwrotną (4.7) należy odtworzyć pełny ciąg  $F$ , wykorzystując podane symetrie.

Powstawanie symetrii w dziedzinie  $F$  dla pierwotnego ciągu  $L$  składającego się wyłącznie z elementów o wartościach rzeczywistych można wyjaśnić bez wnikania w matematyczną strukturę transformacji Fouriera tym, że transformacja przeznaczona w ogólności do ciągów zespolonych potrafi także dla ciągu pierwotnego zespolonego znaleźć reprezentację tego ciągu w postaci ciągów bazowych - jak we wzorach od (4.10) do (4.14). W takim przypadku ciągi bazowe muszą być jednak także zespolone, czyli muszą pochodzić nie od rzeczy-

---

<sup>23</sup> Dokładnie dla  $N$  parzystego musimy znać  $\frac{N}{2} + 1$  współczynników w dziedzinie Fouriera, a dla  $N$  nieparzystego potrzebne jest  $\frac{N+1}{2}$  współczynników.

wistej funkcji kosinus, ale od funkcji zespolonej. Tak się składa, że dla większości ciągów bazowych zespolonych można je pogrupować w pary, które będą miały identyczne części rzeczywiste. Zatem rzeczywisty ciąg  $L$  o długości  $N$  jest tak naprawdę według transformacji Fouriera opisywany przez  $N$  ciągów bazowych. Jednak po pominięciu ciągu bazowego  $L_0$  (oraz - dla  $N$  parzystego - także ciągu bazowego  $\frac{L_N}{2}$ ), który nie ma swojej pary, pozostałe ciągi bazowe można połączyć parami, opierając się na identyczności ich części rzeczywistych. Powyższy opis wyjaśnia również, dlaczego większość amplitud transformaty Fouriera danego ciągu rzeczywistego  $L$  odpowiada połowie amplitud odpowiadających im współczynników w rozwinięciu (4.10)-(4.14).

Zagadnienie symetrii jest szczególnie istotne przy przeprowadzaniu różnych przekształceń ciągu w dziedzinie  $F$ . Jeżeli nie dopilnujemy, by powyższe symetrie były zachowane, to po transformacji odwrotnej otrzymamy ciąg zespolony, z reguły nieprzydatny w praktycznych zastosowaniach.

Często w celu wygodniejszej interpretacji w dziedzinie  $F$  przeprowadza się pewne przemieszczenie elementów ciągu. Wszystkie elementy leżące na prawo od osi symetrii przenosi się w niezmienionym porządku na lewo od elementu  $k=0$  i następuje dotychczasowe indeksy indeksami ujemnymi odpowiadającymi ich nowemu położeniu. W ten sposób dany element ciągu  $F$  w miarę wzrostu  $|k|$  odpowiada coraz większej częstotliwości funkcji kosinus, z której powstał dany składowy ciągu  $L_k$ . Po takim przemieszczeniu dla  $L$  rzeczywistego symetrie można określić jako parzystość amplitudy i nieparzystość fazy. Opisaną operację będziemy nazywać przesunięciem w dziedzinie  $F$ . Należy jednak pamiętać, że przed zastosowaniem wzoru (4.7) na transformację odwrotną konieczne jest powrócenie do właściwej kolejności elementów ciągu  $F$  - tą operację nazwiemy odwrotnym przesunięciem w dziedzinie  $F$ . Warto zauważyć, że dla  $N$  parzystego można do obu operacji wykorzystać tę samą procedurę przemieszczania elementów ciągu, natomiast dla  $N$  nieparzystego dałoby to wynik błędny.

Przesunięcie w dziedzinie  $F$  można zinterpretować także inaczej. Skorzystamy z faktu, że ciąg  $F$  o długości  $N$  otrzymany ze wzoru (4.4) może być uznany za jeden okres o wiele dłuższego, a nawet nieskończonego ciągu o indeksach całkowitych od minus do plus nieskończoności. Taki nieskończony ciąg  $F$  otrzymalibyśmy, gdyby we wzorze (4.4) indeksy  $k$  zmieniały się w pełnym, nieskończonym zakresie. Jeżeli tak, to dlaczego nie wybrać jednego okresu reprezentującego cały nieskończony okresowy ciąg inaczej: tak, aby jego środek wypadał w okolicy indeksu  $k=0$ ? Dla  $N$  parzystego wzory (4.4) i (4.7) można by więc przepisać do postaci podanej niżej.

Transformacja w przód:

$$f_k = \beta_L \cdot \sum_{n=0}^{N-1} l_n \cdot \exp\left(\frac{-j \cdot 2 \cdot \pi \cdot n \cdot k}{N}\right) \quad (4.21)$$

dla  $k = -\left(\frac{N}{2}\right), -\left(\frac{N}{2}-1\right), \dots, -1, 0, 1, \dots, \left(\frac{N}{2}-2\right), \left(\frac{N}{2}-1\right)$

Transformacja odwrotna:

$$l_n = \beta_F \cdot \sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1} f_k \cdot \exp\left(\frac{j \cdot 2 \cdot \pi \cdot k \cdot n}{N}\right) \quad \text{dla } n = 0, 1, \dots, N-1 \quad (4.22)$$

Dla  $N$  nieparzystego we wzorach (10) i (11) indeks  $k$  musiałby się zmieniać następująco:

$$k = -\left(\frac{N-1}{2}\right), -\left(\frac{N-1}{2}-1\right), \dots, -1, 0, 1, \dots, \left(\frac{N-1}{2}-1\right), \left(\frac{N-1}{2}\right) \quad (4.23)$$

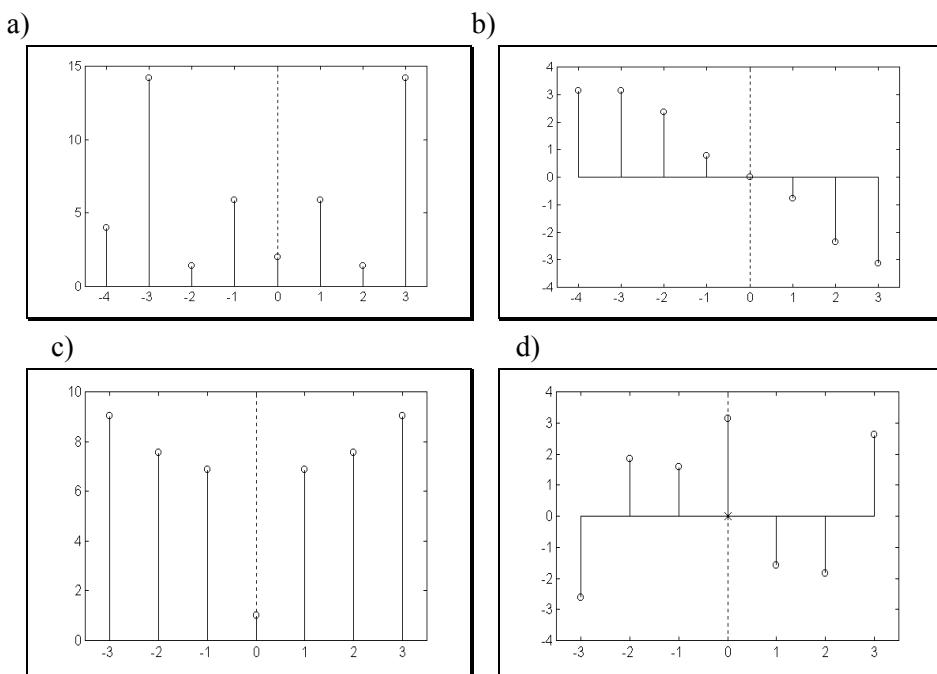
Możliwość zmiany zakresów zmienności indeksów we wzorach (4.21) i (4.22) jest naturalną konsekwencją okresowości funkcji  $\exp$  takiej samej, jak funkcji trygonometrycznych - patrz wzór (4.18) - względem wartości  $2 \cdot \pi$ . Tę okresowość dla dowolnego całkowitego  $m$  zapisać można na przykład tak:

$$\exp\left(\frac{j \cdot 2 \cdot \pi \cdot k \cdot n}{N}\right) = \exp\left(\frac{j \cdot 2 \cdot \pi \cdot (k \pm m \cdot N) \cdot n}{N}\right) \quad (4.24)$$

Z zależności (4.24) wynika bezpośrednio możliwość okresowości ciągu  $L$  względem indeksu  $k$  z podstawowym okresem  $N$ . Jak łatwo się domyślić, z tego samego powodu można by w analogiczny sposób przesunąć także zakres zmienności indeksu  $n$  w dziedzinie Fouriera, ponieważ funkcja  $\exp$  (4.24) posiada identyczną okresowość względem tego indeksu. W większości jednak dostępnych pakietów oprogramowania wyliczane są transformaty w oparciu o wzory (4.4) i (4.7), które nie zmieniają się zależnie od parzystości czy nieparzystości  $N$ . Zatem w dalszym ciągu będziemy pisali o transformacji Fouriera zdefiniowanej zależnością (4.4) i w razie potrzeby stosowali przesunięcie w dziedzinie  $F$ .

Na zakończenie warto wspomnieć, że wyznaczanie transformaty w oparciu o parę wzorów (4.4) i (4.7) jest dość wymagające obliczeniowo i zarówno w literaturze, jak i w bibliotekach różnych języków programowania (np. C++), czy pakietów typu MatLab, można znaleźć algorytmy tak zwanej szybkiej transformacji Fouriera (*skrót ang. FFT*) oraz jej wersji odwrotnej (*skrót ang. IFFT*). Ze względu na powszechnie stosowanie tych algorytmów oraz angielskich skrótów FFT oraz IFFT w środowiskach inżynierskich i naukowych, często traktuje się te skróty jako synonimy w stosunku do obu części dyskretnej transformacji

Fouriera - w przód (4.4) i odwrotnej (4.7). Warto też wspomnieć, że przewaga tych algorytmów nad dosłownym stosowaniem wzorów (4.4) i (4.7) jest szczególnie wyraźna dla  $N$  będącego potęgą liczby 2, na przykład: 128, 256, czy 1024. Nie należy jednak dziwić się, gdy standardowa procedura FFT lub IFFT okaże się niewiele szybsza od bezpośredniego użycia wzoru (4.4) lub (4.7) w przypadku  $N$  innego niż potęga 2.



Rys. 4.4. Przykłady symetrii z rysunku 4.3 po zastosowaniu przesunięcia w dziedzinie  $F$ :

- a)  $N$  parzyste - amplituda, b)  $N$  parzyste - faza, c)  $N$  nieparzyste - amplituda, d)  $N$  nieparzyste - faza.

### 4.3 Transformacja Fouriera dla obrazów cyfrowych

Z matematycznego punktu widzenia zastosowanie dyskretnej transformacji Fouriera do obrazów cyfrowych jest stosunkowo prostym poszerzeniem odpowiednich zależności o jeden wymiar. Przyjmijmy, że obraz cyfrowy to uporządkowany i ponumerowany dwuwymiarowo zbiór liczb, inaczej mówiąc ciąg dwuwymiarowy o wartościach rzeczywistych:

$$L = \{L(m, n) \in \Re: m = 0, 1, \dots, M-1; n = 0, 1, \dots, N-1\} \quad (4.25)$$

Ze względu na podwójne indeksy oraz interpretację obrazu  $L$  jako funkcji określonej na dyskretnie dwuwymiarowej dziedzinie zastosowano zapis indeksów w nawiasach.

Zwykle wartości ciągu  $L$  są nie tylko rzeczywiste, ale nawet naturalne z przedziału na przykład 0-255 - dla pikseli opisanych ośmiobitowo. Z punktu widzenia transformacji Fouriera wystarczy na razie przyjąć, że są to liczby rzeczywiste. Dość często obrazy cyfrowe są kwadratowe. Wtedy  $M=N$  i tak będzie we wszystkich przedstawionych dalej przykładach. W ogólnych zależnościach zachowane zostanie jednak rozróżnienie rozmiaru w poziomie i w pionie. Zatem dla obrazu cyfrowego o rozmiarach  $M \cdot N$  pikseli jego transformatę Fouriera określa się następująco:

$$F(i, k) = \beta_L \cdot \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} L(m, n) \cdot \exp\left(-\frac{j \cdot 2 \cdot \pi \cdot m \cdot i}{M}\right) \cdot \exp\left(-\frac{j \cdot 2 \cdot \pi \cdot n \cdot k}{N}\right) \quad (4.26)$$

dla  $i = 0, 1, \dots, M-1; \quad k = 0, 1, \dots, N-1$

Operacja dwuwymiarowej odwrotnej transformacji Fouriera zdefiniowana jest wzorem:

$$L(m, n) = \beta_F \cdot \sum_{i=0}^{M-1} \sum_{k=0}^{N-1} F(i, k) \cdot \exp\left(\frac{j \cdot 2 \cdot \pi \cdot m \cdot i}{M}\right) \cdot \exp\left(\frac{j \cdot 2 \cdot \pi \cdot n \cdot k}{N}\right) \quad (4.27)$$

dla  $m = 0, 1, \dots, M-1; \quad n = 0, 1, \dots, N-1$

Aby operacje (4.26) i (4.27) były wzajemnymi odwrotnościami, musi zachodzić zależność:

$$\beta_L \cdot \beta_F = \frac{1}{M \cdot N} \quad (4.28)$$

W przedstawionych w tym rozdziale przykładach przyjęto:

$$\beta_L = 1; \quad \beta_F = \frac{1}{M \cdot N} \quad (4.29)$$

Oznacza to, że w dziedzinie transformacji Fouriera element o indeksach  $(0,0)$  jest sumą wszystkich wartości dwuwymiarowego ciągu  $L$ , czyli obrazu:

$$F(0,0) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} L(m, n) \quad (4.30)$$

Duwymiarowy ciąg  $F(i, k)$  przyjmujący wartości ze zbioru liczb zespolonych posiada dokładnie tyle samo elementów co dwuwymiarowy ciąg  $L(m, n)$  będący reprezentacją obrazu cyfrowego. Indeksy obu ciągów zmieniają się w tych samych zakresach. Obraz cyfrowy interpretuje się jako dwuwymiarowy ciąg, dlatego też dwuwymiarowy ciąg w dziedzinie transformaty Fouriera też można zinterpretować jako specyficzny obraz cyfrowy o dokładnie tych samych rozmiarach w pikselach, jedynie prezentacja tego obrazu wymaga pewnych

dodatkowych zabiegów. Można na przykład obejrzeć obraz części rzeczywistej, części urojonej, amplitudy lub fazy. Za każdym razem odpowiednio dostosowując zakres wartości do poziomów szarości z zakresu 0-255 (w zapisie 8-bitowym, co naturalnie można w razie potrzeby zmienić). W celu podkreślenia pewnych cech można także takie obrazy zaprezentować w postaci kolorowej po przypisaniu odpowiednim przedziałom liczb rzeczywistych wybranych kolorów. Z możliwości interpretacji ciągu w dziedzinie transformacji Fouriera jako obrazu skorzystamy nieco później, wtedy też podane zostaną przykłady. Na razie uwaga powyższa upoważnia nas do operowania polem obrazu dla dziedziny transformacji Fouriera w sposób analogiczny jak dla oryginalnego obrazu cyfrowego, łącznie z określeniem poszczególnych elementów tego pola - wraz z odpowiadającymi im wartościami zespolonymi - jako pikseli. Żeby nie było nieporozumień można się umówić, że obrazy w dziedzinie transformacji Fouriera będą nazywane krótko  $F$ -obrazami a ich piksele  $F$ -pikselami. Dzięki powyższej umowie można przyjąć, że mamy do czynienia cały czas z obrazem cyfrowym o tych samych rozmiarach wyrażonych w pikselach, jedynie w dwóch wzajemnie wymiennych postaciach - dwuwymiarowego ciągu  $L(m,n)$  oraz dwuwymiarowego ciągu  $F(i,k)$ .

Podobnie jak dla ciągów jednowymiarowych łatwo można znaleźć gotowe procedury wyznaczające transformaty proste i odwrotne. Różnić się one mogą współczynnikami  $\beta$  i zawsze przed wykorzystaniem wybranej procedury należy się upewnić, jaką parę tych współczynników w danej procedurze zastosowano. Nie będziemy tutaj przytaczać szczegółowego opisu algorytmu dwuwymiarowej transformacji Fouriera. Dla intuicyjnego zrozumienia tego, co wylicza wzór (4.26) i powiązania z rozważaniami dla transformacji jednowymiarowej, warto jednak choć powierzchownie zastanowić się nad tym wzorem. Dwuwymiarowa transformacja Fouriera jest operacją separowaną ze względu na wymiar. Wynika to z faktu, że wzory (4.26) i (4.27) można przepisać do postaci następującej:

$$F(i,k) = \beta_L \cdot \sum_{m=0}^{M-1} \left[ \sum_{n=0}^{N-1} L(m,n) \cdot \exp\left(\frac{-j \cdot 2 \cdot \pi \cdot n \cdot k}{N}\right) \right] \cdot \exp\left(\frac{-j \cdot 2 \cdot \pi \cdot m \cdot i}{M}\right) \quad (4.31)$$

dla  $i = 0, 1, \dots, M-1; \quad k = 0, 1, \dots, N-1$

$$L(m,n) = \beta_F \cdot \sum_{i=0}^{M-1} \left[ \sum_{k=0}^{N-1} F(i,k) \cdot \exp\left(\frac{j \cdot 2 \cdot \pi \cdot n \cdot k}{N}\right) \right] \cdot \exp\left(\frac{j \cdot 2 \cdot \pi \cdot m \cdot i}{M}\right) \quad (4.32)$$

dla  $m = 0, 1, \dots, M-1; \quad n = 0, 1, \dots, N-1$

Nawiasy kwadratowe we wzorach (4.31) i (4.32) wskazują, że najpierw można wyznaczyć sumy wewnętrzne - czyli jednowymiarowe transformaty Fouriera, na przykład dla wszystkich kolumn obrazu - a następnie tak otrzymany obraz po-

średni przetransformować ponownie, ale tym razem wiersz po wierszu. Transformację dwuwymiarową można naturalnie przeprowadzić w dowolnej z dwóch kolejności - najpierw wierszami i potem kolumnami albo najpierw kolumnami a następnie wierszami. W obu przypadkach wynik będzie identyczny. Widać stąd, że dysponując algorytmami jednowymiarowymi FFT oraz IFFT można łatwo stworzyć oparte na tych algorytmach ich dwuwymiarowe odpowiedniki. Tak właśnie działa wiele standardowych procedur dwuwymiarowej FFT oraz IFFT.

Przyjmijmy, że najpierw transformujemy obraz kolumnami. Każda kolumna jest traktowana niezależnie jako jednowymiarowy ciąg. Jak interpretować wynik transformacji jednowymiarowej, opisano w poprzednim podrozdziale. Tutaj jednak otrzymane wyniki układają się obok siebie w miejsce oryginalnych kolumn. W takim ułożeniu współczynniki po transformacji kolumn dotyczące ciągów pochodzących z kosinusoid o tym samym okresie leżą w tych samych wierszach. W pierwszym wierszu leżą - zależnie od użytego współczynnika  $\beta$  - wartości średnie kolumn lub ich sumy.

W drugim etapie transformujemy wiersz po wierszu. Tym razem jednowymiarowy ciąg pierwotny będzie niemal zawsze zespolony, więc fakt, że transformacja Fouriera może w ogólności transformować ciągi zespolone, także się do czegoś przydaje. Ostatecznie, po obu etapach, obraz może zostać opisany jako suma dwuwymiarowych ciągów bazowych z odpowiednimi współczynnikami:

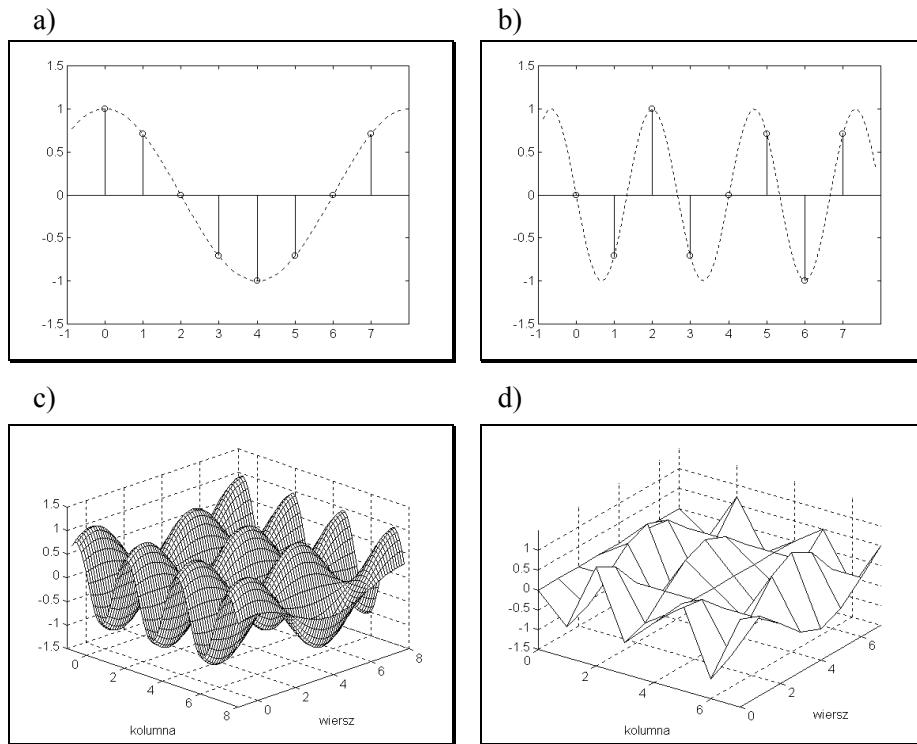
$$L(m, n) = \sum_{i=0}^{\frac{M}{2}} \sum_{k=0}^{\frac{N}{2}} a_{i,k} \cdot L_{i,k}(m, n) \quad (4.33)$$

We wzorze (4.33) założono, że zarówno  $M$  jak  $N$  są parzyste. Jakie będą granice sumowania dla pozostałych przypadków, można wywnioskować z zależności (4.10)-(4.14).

Każdy z ciągów składowych  $L_{i,k}$  otrzymany jest z iloczynu ciągów analogicznych do opisanych w rozdziale 4.1. Korzystając więc z oznaczeń wprowadzonych w rozdziale 4.1 (patrz wzory (4.10)-(4.14)) każdy element ciągu bazowego  $L_{i,k}$  można zapisać następująco za pomocą elementów ciągów jednowymiarowych zdefiniowanych według zależności (4.15):

$$L_{i,k}(m, n) = l_{m,i} \cdot l_{n,k} \quad (4.34)$$

Przykład otrzymywania bazowego ciągu dwuwymiarowego według zależności (4.34) pokazano na rysunku 4.5.



Rys. 4.5. Dwuwymiarowy ciąg składowy powstały z dwóch jednowymiarowych ciągów - wierszowego i kolumnowego:

a) wolnozmienny ciąg wierszowy,  $N=8$ ,  $k=1$ ,  $\varphi_k = 0$  , b) szybkozmienny ciąg

kolumnowy  $M=8$ ,  $i=3$ ,  $\varphi_i = \frac{\pi}{2}$  , c) powierzchnia, której próbkowanie daje dwuwymiarowy ciąg z punktu d), d) wynikowy ciąg dwuwymiarowy - wartości ciągu w punktach przecięcia siatki wykresu.

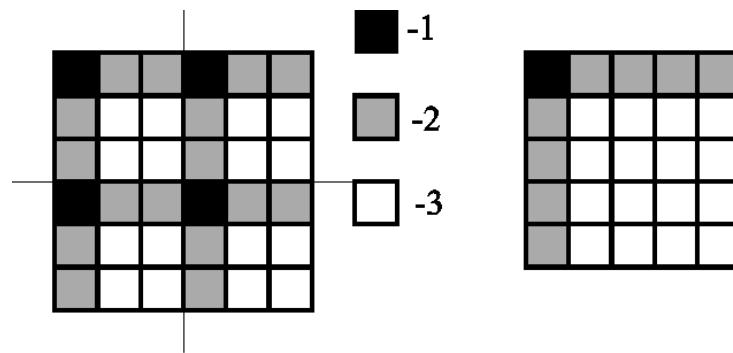
Patrząc na kompletny zestaw bazowych ciągów dwuwymiarowych  $L_{i,k}$  zauważamy, że powstały one przez skojarzenie ciągów jednowymiarowych na zasadzie „każdy z każdym”. Jest zatem wśród nich ciąg dwuwymiarowy powstały z kolumnowego jednowymiarowego ciągu wolnozmiennego (powstałego z kosinusoidą o długim okresie) i wierszowego ciągu szybkozmiennego (z kosinusoidą o krótkim okresie). Inny ciąg składowy powstał ze skojarzenia dwóch ciągów - kolumnowego i wierszowego - szybkozmiennych, a jeszcze inny z dwóch jednowymiarowych ciągów wolnozmiennych. Położenie współczynnika po pełnej dwuwymiarowej transformacji Fouriera na dwuwymiarowym polu indeksów jednoznacznie informuje, z jaką kombinacją ciągów jednowymiarowych należy go łączyć. Przykładowo: w punkcie  $(0,0)$  znajduje się wartość średnia (lub suma) - zwana często składową stałą, w punkcie  $(0,N-1)$  współczynnik określający udział w obrazie ciągu powstałego ze stałego ciągu wierszowego i najszybciej

zmiennego ciągu kolumnowego, a w punkcie  $\left(\frac{M-1}{2}, \frac{N-1}{2}\right)$  umieszczony jest

współczynnik dotyczący połączenia ciągów kolumnowego i wierszowego o średnio-szybkiej zmienności. Może się zdarzyć, że gdy  $M$  jest na przykład dwa razy większe od  $N$ , to inna będzie ta pośrednia szybkość zmian dla ciągu wierszowego a inna dla kolumnowego. W każdym razie zawsze maksymalna szybkość zmian w danym kierunku - kolumnowym lub wierszowym - oznacza kosinusoidę generującą odpowiedni ciąg jednowymiarowy o ekstremach (+1 oraz -1) w miejscu położenia sąsiednich pikseli.

Porównując wzór (4.33) dla przypadku dwuwymiarowego i (4.10) dla przypadku jednowymiarowego oraz odpowiednie wyjaśnienia - np. zależności (4.15), (4.19) i (4.20), można zauważać podobieństwa i różnice. Również w przypadku dwuwymiarowym współczynniki transformaty Fouriera - elementy ciągu dwuwymiarowego  $F$  - mają bezpośredni związek ze współczynnikami  $a_{i,k}$  oraz przesunięciami fazowymi kosinusoid generujących ciągi jednowymiarowe, z których następnie otrzymywane są bazowe ciągi dwuwymiarowe. Współczynniki transformaty dwuwymiarowej są zespolone i podobnie jak w (4.18) można je interpretować w postaci części rzeczywistej i urojonej lub w postaci amplitudy i fazy:

$$\begin{aligned} F(i, k) &= A(i, k) \cdot \exp(j \cdot \varphi(i, k)) = \\ &= \operatorname{Re}(F(i, k)) + j \cdot \operatorname{Im}(F(i, k)) = \\ &= A(i, k) \cdot \cos(\varphi(i, k)) + j \cdot A(i, k) \cdot \sin(\varphi(i, k)) \end{aligned} \quad (4.35)$$

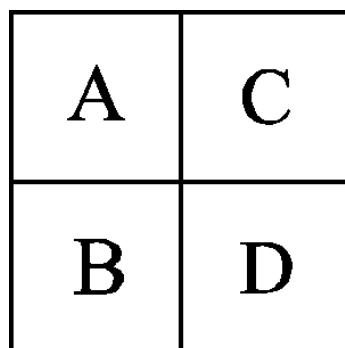


Rys. 4.6. Lokalizacja  $F$ -pikseli poszczególnych typów dla wzoru (4.36):  
z lewej dla obrazu o rozmiarach parzystych (6x6),  
z prawej dla obrazu o rozmiarach nieparzystych (5x5).

Tym razem jednak zależność pomiędzy zespolonymi współczynnikami transformaty -  $F(i, k)$  - oraz współczynnikami  $a_{i,k}$  we wzorze (4.33) jest następująca (na rysunku 4.6 pokazano rozmieszczenie  $F$ -pikseli poszczególnych typów):

$$A(i, k) = \begin{cases} a_{i,k} \cdot \frac{1}{\beta_L} & \text{dla } F - \text{pikseli typu 1} \\ \frac{1}{2} \cdot a_{i,k} \cdot \frac{1}{\beta_L} & \text{dla } F - \text{pikseli typu 2} \\ \frac{1}{4} \cdot a_{i,k} \cdot \frac{1}{\beta_L} & \text{dla } F - \text{pikseli typu 3} \end{cases} \quad (4.36)$$

Podobnie jak w przypadku jednowymiarowym sama znajomość współczynników w równaniu (4.33) nie wystarcza, by jednoznacznie określić ciąg  $L$ . Potrzebna jest jeszcze informacja, jak były przesunięte w fazie obie funkcje kosinus, które posłużyły do wygenerowania ciągów jednowymiarowych o elementach  $l_{m,i}$  oraz  $l_{n,k}$  - patrz wzór (4.34). Faza jednego  $F$ -piksla nie może jednoznacznie określać dwóch faz kosinusów generujących ciąg bazowy. Zatem zależność nie jest tak bezpośrednia jak dla przypadku jednowymiarowego. Oznaczmy - patrz wzór (4.15) - przez  $\varphi_i$  fazę kosinusa generującego ciąg jednowymiarowy o elementach  $l_{i,m}$ , przez  $\varphi_k$  fazę kosinusa generującego ciąg jednowymiarowy o elementach  $l_{n,k}$ , a przez  $\varphi(i, k)$  fazę  $F$ -piksla o współrzędnych  $i, k$  na polu obrazowym – porównaj wzór (4.35).  $F$ -piksele typu 1 - patrz rysunek 4.6 - powinny być zawsze rzeczywiste.  $F$ -piksele typu 2 wiążą się z dwuwymiarowymi ciągami bazowymi, które powstały z ciągu stałego (bez fazy) i jednej kosinusoidy (które można przesuwać fazowo). Zarówno dla typu 1 jak i 2 łatwo jest określić związek pomiędzy odpowiednimi fazami. Natomiast dla wyjaśnienia zależności fazowych  $F$ -pikseli typu 3 wygodnie będzie przyjąć uproszczony opis  $F$ -obrazu, pomijający istnienie  $F$ -pikseli typu 1 i typu 2 - rysunek 4.7.



Rys. 4.7. Pole  $F$ -obrazu z zaznaczonymi obszarami -  $A, B, C$  i  $D$  - używanymi do wyjaśnienia symetrii amplitudy, fazy. Założono duży rozmiar obrazu w pikselach, zatem pojedyncze  $F$ -piksele nie są widoczne. W rezultacie nie jest istotne, czy obraz ma rozmiar parzysty, czy nieparzysty, ponieważ  $F$ -piksele typu 1 i typu 2 stanowią pomijalnie mały fragment  $F$ -obrazu.

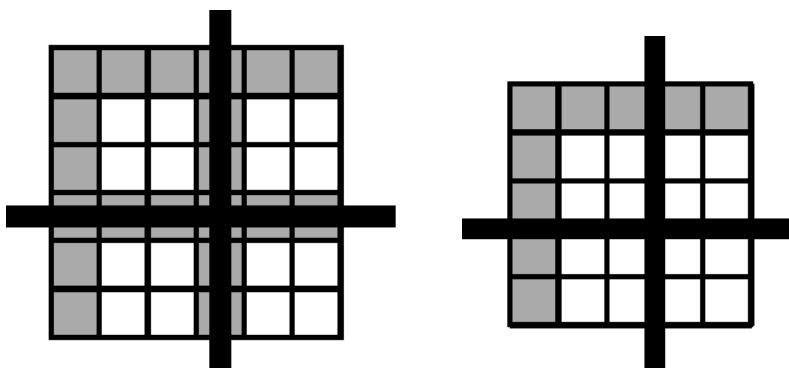
Jeżeli dany  $F$ -piksel typu 3 należy do obszaru  $A$ , to jego faza jest powiązana z fazami kosinusów generujących - zależności (4.15), (4.34) i (4.35) - następująco:

$$\phi(i, k) = \phi_i + \phi_k \quad (4.37)$$

Dla  $F$ -pikseli typu 3 należących do pozostałych obszarów:  $B$ ,  $C$  i  $D$ , wygodnie będzie najpierw wyjaśnić symetrie zachodzące po dwuwymiarowej transformacji Fouriera obrazu cyfrowego, który przecież składa się z pikseli o wartościach bez części urojonej.

#### 4.3.1 Symetrie w dwuwymiarowej transformacji Fouriera obrazu o rzeczywistych wartościach pikseli

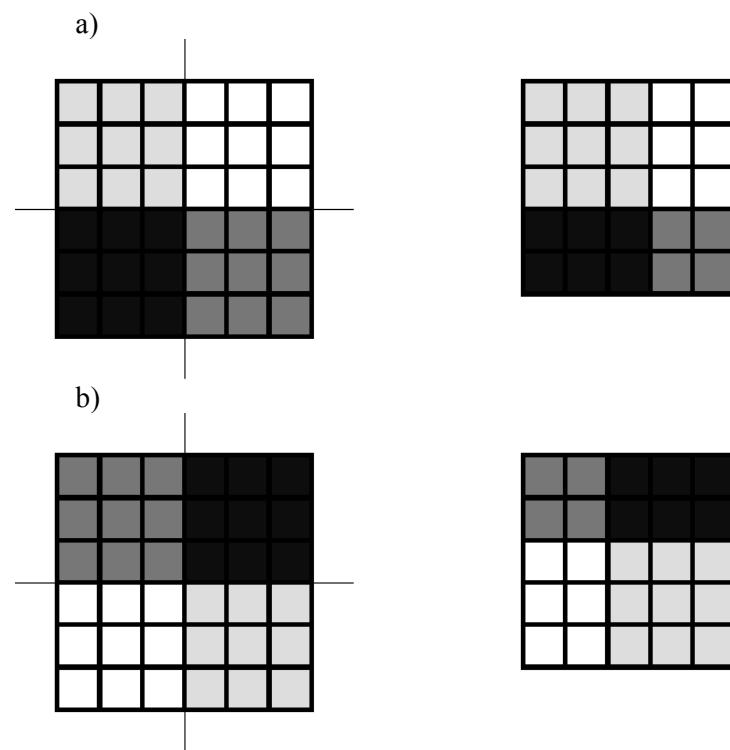
W przypadku transformacji Fouriera ciągów dwuwymiarowych zagadnienie symetrii jest nieco bardziej złożone niż dla ciągów jednowymiarowych. Specyficzne symetrie w dziedzinie transformaty Fouriera powstają, gdy oryginalny ciąg składa się z elementów o wartościach wyłącznie rzeczywistych. W przypadku obrazów cyfrowych sytuacja taka ma miejsce - wartości pikseli nie posiadają części urojonej. Warto zwrócić uwagę, że gdyby w dziedzinie transformaty  $F$ -obraz był wyłącznie rzeczywisty albo wyłącznie urojony, to także obraz oryginalny wykazywałby opisywane symetrie. Zagadnienie symetrii zostanie opisane w sposób nieco uproszczony, bez wnikania w szczególne analityczne. Rozważmy dwa obrazy cyfrowe, jeden o rozmiarach parzystych - na przykład 6x6 pikseli - i drugi o rozmiarach nieparzystych - na przykład 5x5 pikseli.



Rys. 4.8. Ilustracja do wyjaśnienia symetrii w dziedzinie transformacji Fouriera (opis poniżej): z lewej dla obrazu o rozmiarach parzystych (6x6), z prawej dla obrazu o rozmiarach nieparzystych (5x5).

Czytelników zainteresowanych tym zagadnieniem zachęcamy do sprawdzenia odpowiednich symetrii dla obrazu o parzystej liczbie wierszy i nieparzystej liczbie kolumn (lub odwrotnie). Dla odpowiednich  $F$ -obrazów otrzymuje się symetrie, które można wyjaśnić w oparciu o rysunek 4.8. Pogrubione linie pokazują

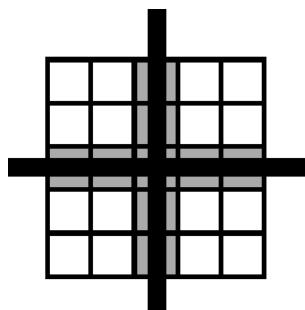
zują osie symetrii dla amplitudy, a ich punkt przecięcia określa punkt symetrii dla fazy.  $F$ -piksele oznaczone kolorem szarym nie są związane powyższymi symetriami. Są to  $F$ -piksele typu 1 i typu 2. Jednak każdy wiersz lub kolumna, która jest cała oznaczona na szaro, musi spełniać warunki symetrii dotyczące ciągów jednowymiarowych. Można to łatwo uzasadnić opierając się na zależnościach (4.31) i (4.32) oraz dwustopniowej procedurze transformacji wiersze-kolumny (lub kolumny-wiersze).



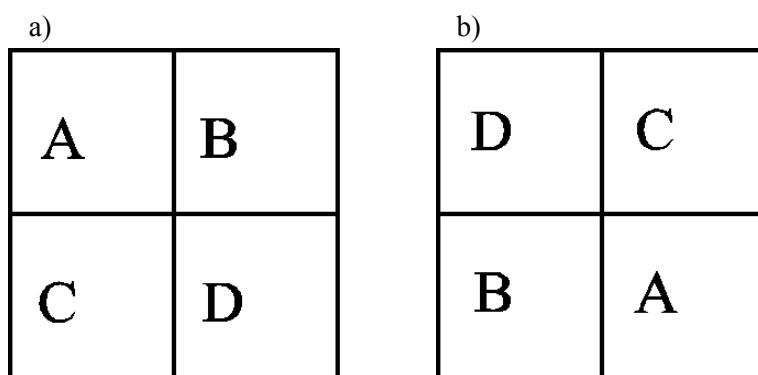
Rys. 4.9. Przesunięcia poszczególnych obszarów  $F$ -obrazów - z lewej pole  $F$ -obrazu o rozmiarach parzystych (6x6), z prawej o rozmiarach nieparzystych (5x5):  
a) pola  $F$ -obrazów przed przesunięciem b) pola  $F$ -obrazów po przesunięciu.

Podobnie jak w przypadku ciągów jednowymiarowych, także i tutaj często stosuje się przesunięcie w dziedzinie Fouriera. Tym razem jednak musi to być przesunięcie w dwóch wymiarach. Rysunek 4.9 pokazuje schematycznie, do czego praktycznie sprowadza się przesunięcie w przypadku obrazu o rozmiarach parzystych i nieparzystych. Wewnątrz danego obszaru oznaczonego tym samym kolorem nie ma żadnych przemieszczeń. Ważne jest tu stwierdzenie, że  $F$ -piksel odpowiadający składowej stałej przed wykonaniem przesunięcia w dziedzinie  $F$  znajduje się w  $F$ -obrazie w lewym górnym rogu - niezależnie od tego, czy obraz jest typu parzystego, czy nieparzystego. Natomiast po przesunięciu ten  $F$ -piksel znajduje się dokładnie w centrum obrazu nieparzystego, a w lewym górnym rogu

prawej dolnej ćwiartki  $F$ -obrazu parzystego. Na krawędziach  $F$ -obrazu po przesunięciu znajdują się  $F$ -piksele odpowiadające największym częstotliwościom odpowiednich kosinusoid, które posłużyły do wygenerowania ciągów bazowych. Jak widać z rysunku 4.8, dla obrazu typu parzystego do wykonania zarówno przesunięcia jak i przesunięcia odwrotnego można zastosować tę samą procedurę, podczas gdy dla obrazu typu nieparzystego dwukrotne zastosowanie tej samej procedury nie zakończy się otrzymaniem  $F$ -obrazu pierwotnego - czyli tego, który powstaje bezpośrednio po zastosowaniu wzoru (26). Po wykonaniu przesunięcia w dziedzinie  $F$  symetrie dla obrazu typu parzystego będą wyglądały identycznie jak przed przesunięciem, natomiast dla obrazu typu nieparzystego symetrie po przesunięciu pokazane są na rysunku 4.9 - rola pogrubionych linii i szarych  $F$ -pikseli analogiczna jak na rysunku 4.8. Podobnie jak w przypadku ciągów jednowymiarowych, po to by poprawnie zastosować wzór (4.27) na transformację odwrotną,  $F$ -obraz musi mieć przywróconą postać sprzed przesunięcia w dziedzinie  $F$  - za pomocą przesunięcia odwrotnego.

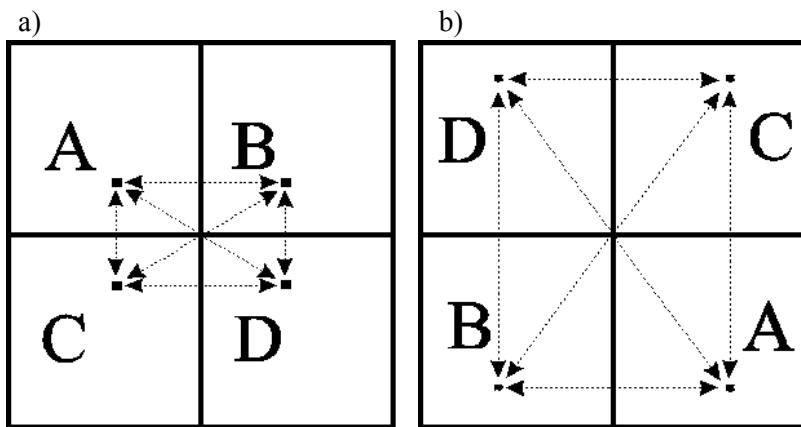


Rys. 4.10. Pole  $F$ -obrazu z rysunku 4.8 (z prawej) po wykonaniu przesunięcia, zaznaczono nowe osie symetrii.



Rys. 4.11. Uproszczona ilustracja przesunięcia w dziedzinie  $F$ , przy pominięciu szczegółów związanych z pojedynczymi  $F$ -pikselsami (litery nie są treścią  $F$ -obrazu a jedynie oznaczeniem jego fragmentów):

a)  $F$ -obraz przed przesunięciem, b)  $F$ -obraz po przesunięciu.



Rys. 4.12. Ilustracja symetrii zachodzących w uproszczonej (z pominięciem  $F$ -pikseli typu 1 i typu 2) interpretacji  $F$ -obrazu, czarne prostokąty oznaczają wybrane  $F$ -piksele, linie przerywane ze strzałkami zachodzące relacje symetrii:  
a)  $F$ -obraz przed przesunięciem, b)  $F$ -obraz po przesunięciu.

Rysunek 4.12 ukazuje relacje symetrii zachodzące pomiędzy  $F$ -pikselami należącymi do poszczególnych ćwiartek  $F$ -obrazu. Symetrie te są dwóch typów. Symetria osiowa - względem danej linii pionowej lub poziomej - oraz symetria punktowa - względem środka obrazu, wyznaczonego przez przecięcie linii symetrii pionowej i poziomej. Po powiększeniu do skali ukazującej pojedyncze  $F$ -piksele ujrzelibyśmy, że linie symetrii, a zatem i środek symetrii, położone są zgodnie z rys. 4.8. Pozostaniemy jednak przy skali rysunku 4.12. Wprowadźmy następujące oznaczenia, określające wszystkie piksele  $F$ -obrazu w odniesieniu do obszaru  $A$ :

$F_A(i, k)$  -  $F$ -piksel należący do obszaru  $A$  o położeniu wyznaczonym przez indeksy  $(i, k)$ ;

$F_{B/A}(i, k)$  -  $F$ -piksel z obszaru  $B$  o położeniu symetrycznym względem osi pionowej względem  $F$ -piksla o indeksach  $(i, k)$ ;

$F_{C/A}(i, k)$  -  $F$ -piksel z obszaru  $C$  o położeniu symetrycznym względem osi poziomej względem  $F$ -piksla o indeksach  $(i, k)$ ;

$F_{D/A}(i, k)$  -  $F$ -piksel z obszaru  $D$  o położeniu symetrycznym punktowo względem środka obrazu względem  $F$ -piksla o indeksach  $(i, k)$ .

Jeżeli teraz przyjmiemy, że przykładowo:  $A_{B/A}(i, k)$  oznacza amplitudę - patrz wzór (4.35) -  $F$ -piksla należącego do obszaru  $B$  i będącego w odpowiedniej relacji symetrii względem  $F$ -piksla obszaru  $A$  o indeksach  $(i, k)$  i analogicznie  $\varphi_{B/A}(i, k)$  oznacza fazę - ponownie wzór (4.35) - tego samego  $F$ -piksla należącego do obszaru  $B$ , to:

$$\begin{aligned} A_A(i, k) &= A_{B/A}(i, k) \\ A_{B/A}(i, k) &= A_{C/A}(i, k) \\ A_{C/A}(i, k) &= A_{D/A}(i, k) \end{aligned} \quad (4.38)$$

Zatem wszystkie cztery  $F$ -piksele wyszczególnione na rysunku 4.12 posiadają tę samą amplitudę. Relacje fazowe są następujące:

$$\begin{aligned} \varphi_A(i, k) &= -\varphi_{D/A}(i, k) \\ \varphi_{B/A}(i, k) &= -\varphi_{C/A}(i, k) \end{aligned} \quad (4.39)$$

Dla fazy nie ma związku pomiędzy  $\varphi_A(i, k)$  i  $\varphi_{B/A}(i, k)$  ani pomiędzy  $\varphi_{C/A}(i, k)$  i  $\varphi_{D/A}(i, k)$ . Zatem relacje fazowe - zależność (4.39) - dotyczą jedynie symetrii punktowej względem środka  $F$ -obrazu. Dzięki temu jednak wszystko się zgadza. Do określenia rozwinięcia (4.33) potrzebne są amplitudy z jednej ćwiartki  $F$ -obrazu, jednak fazy z jednej ćwiartki nie wystarczyłyby do jednoznacznego określenia faz kosinusoid jednowymiarowych generujących ciągi bazowe. Uwzględniając zależność (4.37) można otrzymać dwa równania na dwie szukane fazy jednowymiarowych kosinusoid:

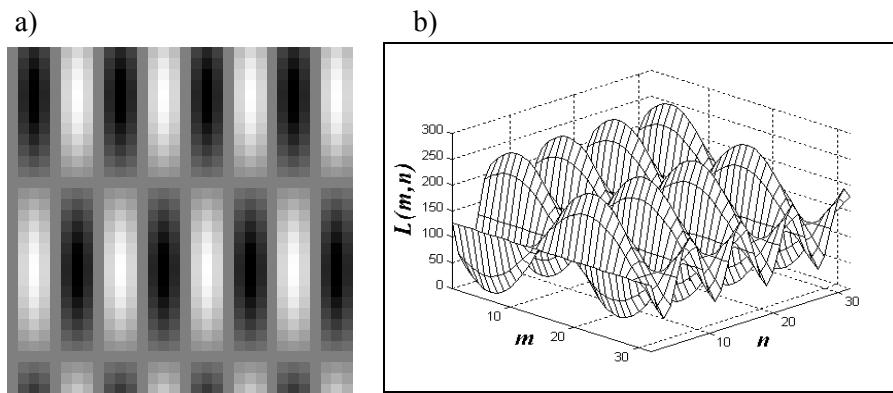
$$\begin{aligned} \varphi_A(i, k) &= \varphi_i + \varphi_k \\ \varphi_{C/A}(i, k) &= \varphi_i - \varphi_k \end{aligned} \quad (4.40)$$

Podsumowując przeprowadzone rozważania można stwierdzić, że do pełnego odtworzenia obrazu oryginalnego w oparciu o jego  $F$ -obraz wystarczy, jeżeli znane będą wartości amplitud  $F$ -pikseli typu 3 dowolnej ćwiartki oraz wartości faz  $F$ -pikseli typu 3 należących do dwóch ćwiartek stykających się ze sobą krawędziami, a także dodatkowo wartości wybranych  $F$ -pikseli typu 2 i typu 1. Wyboru należałoby dokonać traktując  $F$ -piksele typu 1 i typu 2 układające się w wiersze lub kolumny tak jak transformaty Fouriera rzeczywistych ciągów jednowymiarowych - w oparciu o zachodzące w takim przypadku odpowiednie symetrie.

### 4.3.2 Przykłady obrazów cyfrowych oraz ich $F$ -obrazów

Dalsze zagadnienia zostaną przedstawione w oparciu o przykłady. Przyjmijmy, że podlegający przetwarzaniu obraz cyfrowy - o nazwie np. „dwie\_fale” - powstał w wyniku zastosowania następującej zależności:

$$\begin{aligned} L(m, n) &= 128 + 127 \cdot \cos\left(\frac{2 \cdot \pi \cdot m}{32} + \frac{3 \cdot \pi}{4}\right) \cdot \cos\left(\frac{2 \cdot \pi \cdot n}{32/4} - \frac{\pi}{2}\right) \\ m &= 0, 1, \dots, 31; \quad n = 0, 1, \dots, 31 \end{aligned} \quad (4.41)$$



Rys. 4.13. Obraz cyfrowy „dwie\_fale”, 32x32 piksele, opisany zależnością (4.41):  
a) treść obrazu ukazana w poziomach szarości, b) obraz pokazany jako wykres funkcji określonej na dziedzinie dwuwymiarowej.

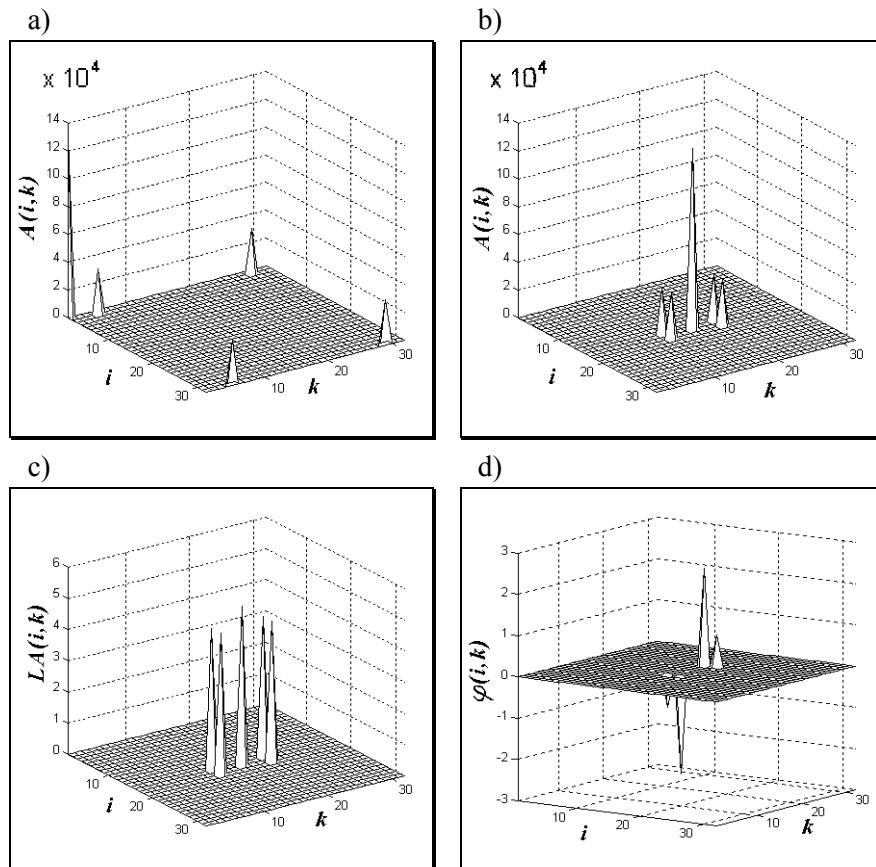
Przed prezentacją na rys. 4.13a elementy ciągu  $L$  (4.41) musiały być dodatkowo zaokrąglone do wartości całkowitych nie mniejszych niż 0 i nie większych niż 255. Obraz cyfrowy pokazany na obu rysunkach 4.13a i 4.13b to ten sam obraz. Rysunki wyglądają jednak inaczej. Po transformacji Fouriera obraz „dwie\_fale” jest nadal tym samym obrazem, jednak w innej reprezentacji, w innej postaci. W dziedzinie Fouriera reprezentacja, czyli  $F$ -obraz, jest już zespolona. Amplituda  $F$ -obrazu „dwie\_fale” pokazana jest na rysunku 4.14a. Rysunek 4.14b przedstawia amplitudę po przeprowadzeniu przesunięcia w dziedzinie  $F$ , a rysunek 4.14c po zlogarytmowaniu wartości amplitudy. Operacja logarytmowania amplitudy jest stosowana bardzo często ze względu na dość znaczne różnice w wartościach amplitud poszczególnych  $F$ -pikseli dla większości obrazów. Skala liniowa powodowałaby trudności w jednolitej interpretacji wartości o tak dużej rozpiętości. Niektóre wartości amplitud mogą mieścić się w przedziale od 0 do 1, co w wyniku logarytmowania mogłoby dać duże wartości ujemne. Aby tego uniknąć przed logarytmowaniem do amplitudy  $F$ -obrazu, oznaczonej jako  $A(i,k)$  - zgodnie ze wzorem (4.35) - dodaje się wartość <sup>24</sup>.

$$LA(i,k) = \log_{10}(A(i,k) + 1) \quad (4.42)$$

Rysunek 4.14d przedstawia fazę  $F$ -obrazu po przesunięciu w dziedzinie  $F$ .

Na wykresach z rysunku 4.14 i dalszych po przesunięciu w dziedzinie  $F$  nie zastosowano zmiany indeksów. Wynika to z tego, że odtąd będziemy traktować operację transformacji Fouriera i przesunięcia w dziedzinie tej transformacji łącznie, jako całość, zmierzającą do otrzymania  $F$ -obrazu określonego na tym samym polu indeksów co obraz oryginalny.

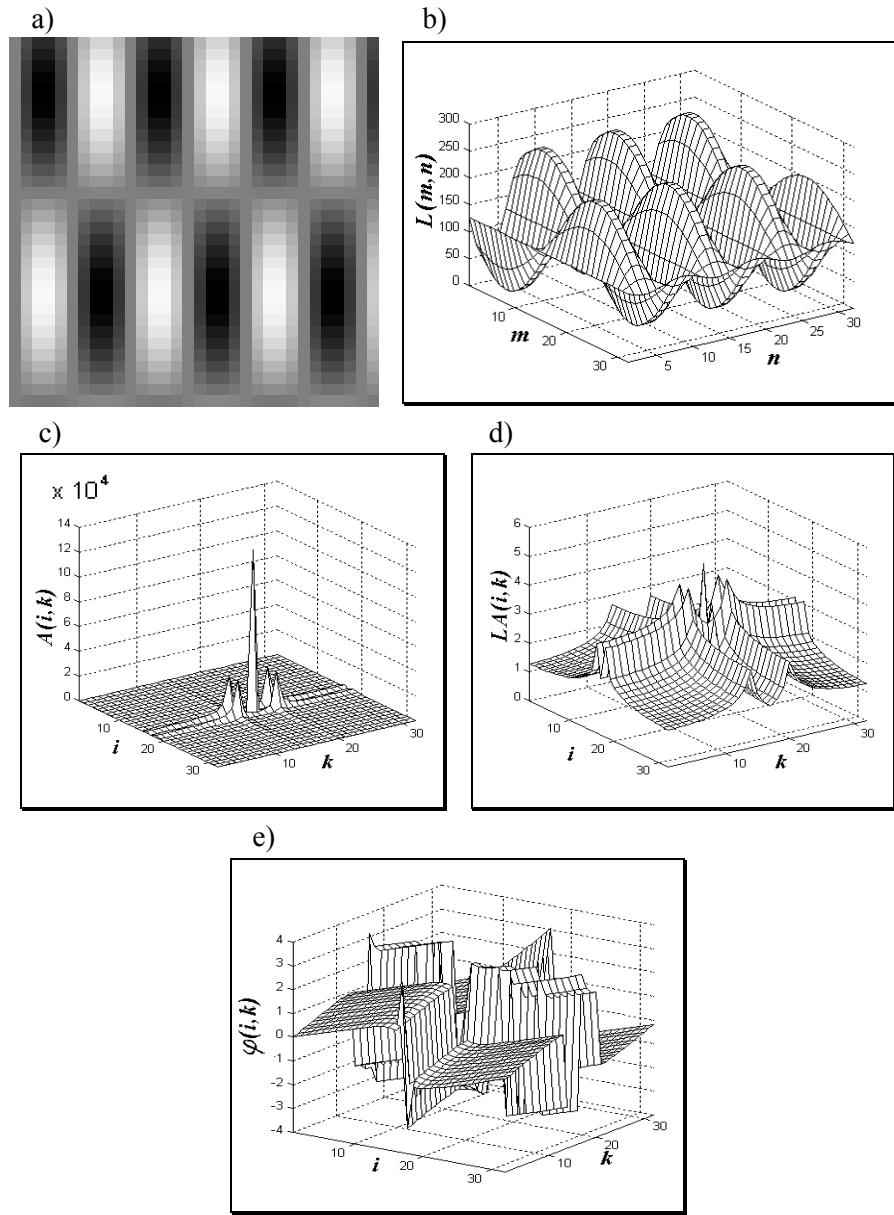
<sup>24</sup> Od tej pory wszystkie wykresy amplitud  $F$ -obrazów będą od razu prezentowane po operacji logarytmowania i nie będzie to więcej podkreślane - analogicznie jak w przypadku przesunięcia w dziedzinie  $F$ .



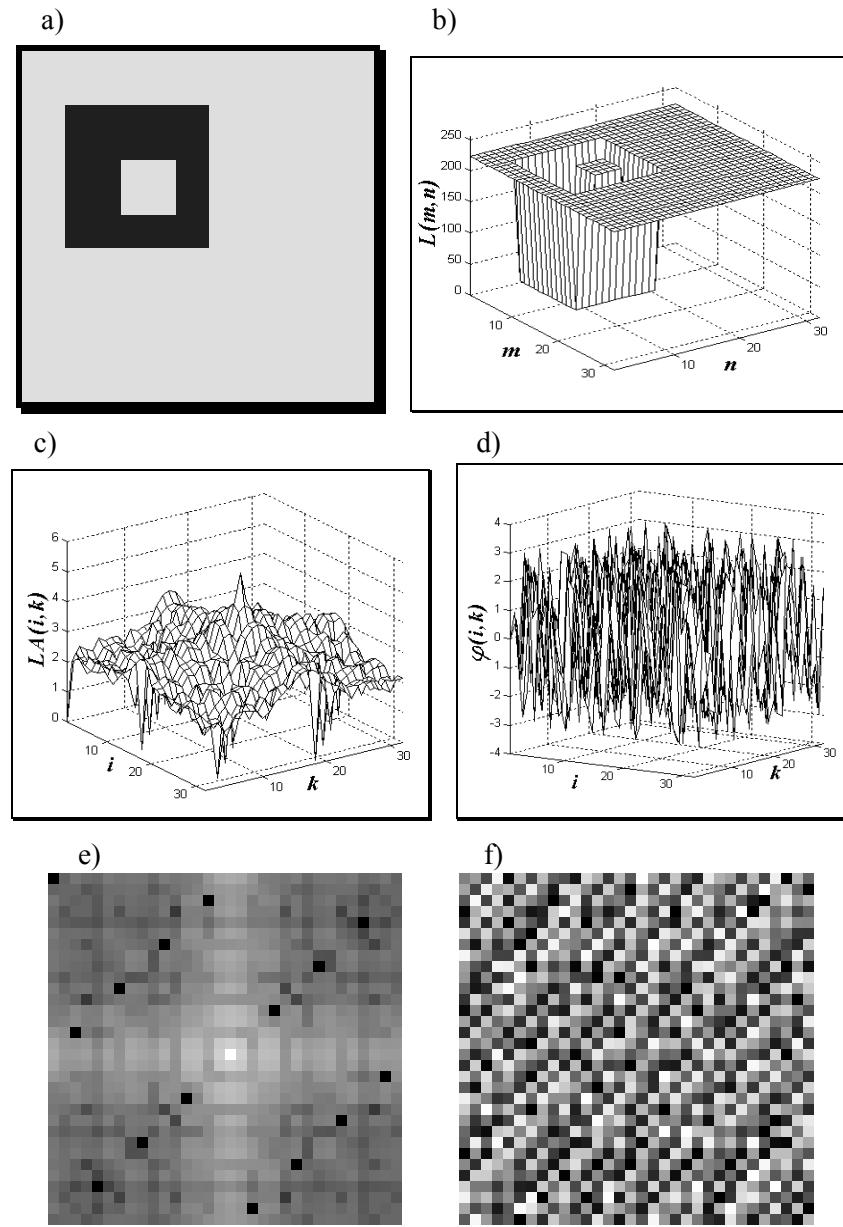
Rys. 4.14. Wykresy  $F$ -obrazu „dwie\_fale”: a) amplituda bezpośrednio po zastosowaniu wzoru (4.26), b) amplituda po przesunięciu w dziedzinie  $F$ , c) amplituda po przesunięciu w dziedzinie  $F$  i zastosowaniu operacji (4.42),  
d) faza po przesunięciu w dziedzinie  $F$ .

Rysunki 4.13 i 4.14 mogą sugerować, że transformacja Fouriera może się przydać do wykrywania pewnych charakterystycznych wzorców w obrazie. Gdyby każdy obraz można było przekształcić do postaci tak klarownych wykresów jak na rysunku 4.14, a następnie udało by się powiązać z wybranymi „słupkami” wykresów informację wizualną zawartą w obrazie, to można by łatwo wyodrębniać z obrazu wybrane fragmenty informacji. Zatem dla porównania warto obejrzeć bardzo podobny pod względem wizualnym obraz cyfrowy, który stanowi fragment obrazu z rys. 4.13a powiększony do tej samej rozdzielczości co obraz oryginalny. Treścią obu obrazów jest zatem ten sam wzorzec, obrazy różnią się jedynie dopasowaniem ramki do powtarzalności wzorca. W przypadku obrazu z rysunku 4.13a oba okresy - pionowy i poziomy - mieszczą się w obrazie całkowitą ilość razy, czego nie da się powiedzieć o obrazie z rysunku 4.15a. Niestety zazwyczaj relacja pomiędzy występującymi w obrazie okresowymi wzorcami i ramką

obrazu jest dość przypadkowa, nie jesteśmy zatem w stanie przewidzieć, czy o występowaniu danego wzorca będzie świadczyć postać  $F$ -obrazu taka jak na rysunku 4.14, czy taka jak na rysunku 4.15.



Rys. 4.15. Obraz „dwie\_fale2”: a) prezentacja obrazu w poziomach szarości, b) obraz w postaci wykresu funkcji, c) amplituda  $F$ -obrazu (po przesunięciu w dziedzinie  $F$ ), d) amplituda  $F$ -obrazu (po przesunięciu w dziedzinie  $F$  i operacji logarytmowania według (4.42)), e) faza  $F$ -obrazu.



Rys. 4.16. Obraz „kwadrat” o rozdzielcości 32x32 piksele: a) prezentacja obrazu w poziomach szarości (ramka nie należy do treści obrazu), b) obraz w postaci wykresu funkcji, c) amplituda  $F$ -obrazu w postaci wykresu funkcji, d) faza  $F$ -obrazu w postaci wykresu funkcji, e) amplituda  $F$ -obrazu w poziomach szarości f) faza  $F$ -obrazu w poziomach szarości.

Zawartości obrazów „dwie\_fale” i „dwie\_fale2” są dla obserwatora raczej mało interesujące. Zanim jednak zostanie pokazany  $F$ -obraz standardowego obrazu „Lena”,

zostanie wykorzystany jeszcze jeden sztuczny obraz „kwadrat” (rysunek 4.16a). Treść obrazu nie jest już tak trudna w interpretacji wizualnej jak w dotychczasowych obrazach („dwie\_fale” i „dwie\_fale2”). Za to  $F$ -obraz, pomimo stosunkowo prostej zawartości obrazu, jest już dość skomplikowany - patrz rysunek 4.16c i 4.16d oraz 4.16e i 4.16f. Na rysunku tym - części „e” i „f” - zaprezentowano kolejną możliwość: prezentację  $F$ -obrazu w postaci takiej samej jak obrazu oryginalnego, czyli po zastąpieniu liczb poziomami szarości. Ponieważ  $F$ -obraz jest zespolony, zatem w miejsce jednego obrazu o 256 poziomach szarości otrzymuje się dwie części  $F$ -obrazu, których przybliżeniem są obrazy o 256 poziomach szarości. Obraz amplitudy  $F$ -obrazu można traktować ponownie jako nowy obraz oryginalny i oczywiście dla niego też można znaleźć postać  $F$ -obrazu. Tak samo reprezentacja w poziomach szarości fazy  $F$ -obrazu może być od tego momentu traktowana jak kolejny obraz cyfrowy. Trzeba jednak pamiętać, że dla przedstawienia amplitudy lub fazy w postaci poziomów szarości konieczne są dwie dodatkowe operacje:

- 1) dostosowanie zakresu do przedziału od 0 do 255, indeks  $p$  oznacza wartość przeznaczoną do prezentacji w postaci poziomów szarości:  
*dla amplitudy:*

$$A_p(i, k) = \frac{A(i, k) - \min_{i,k}(A(i, k))}{\max_{i,k}(A(i, k)) - \min_{i,k}(A(i, k))} \cdot 255 \quad (4.43)$$

*dla fazy:*

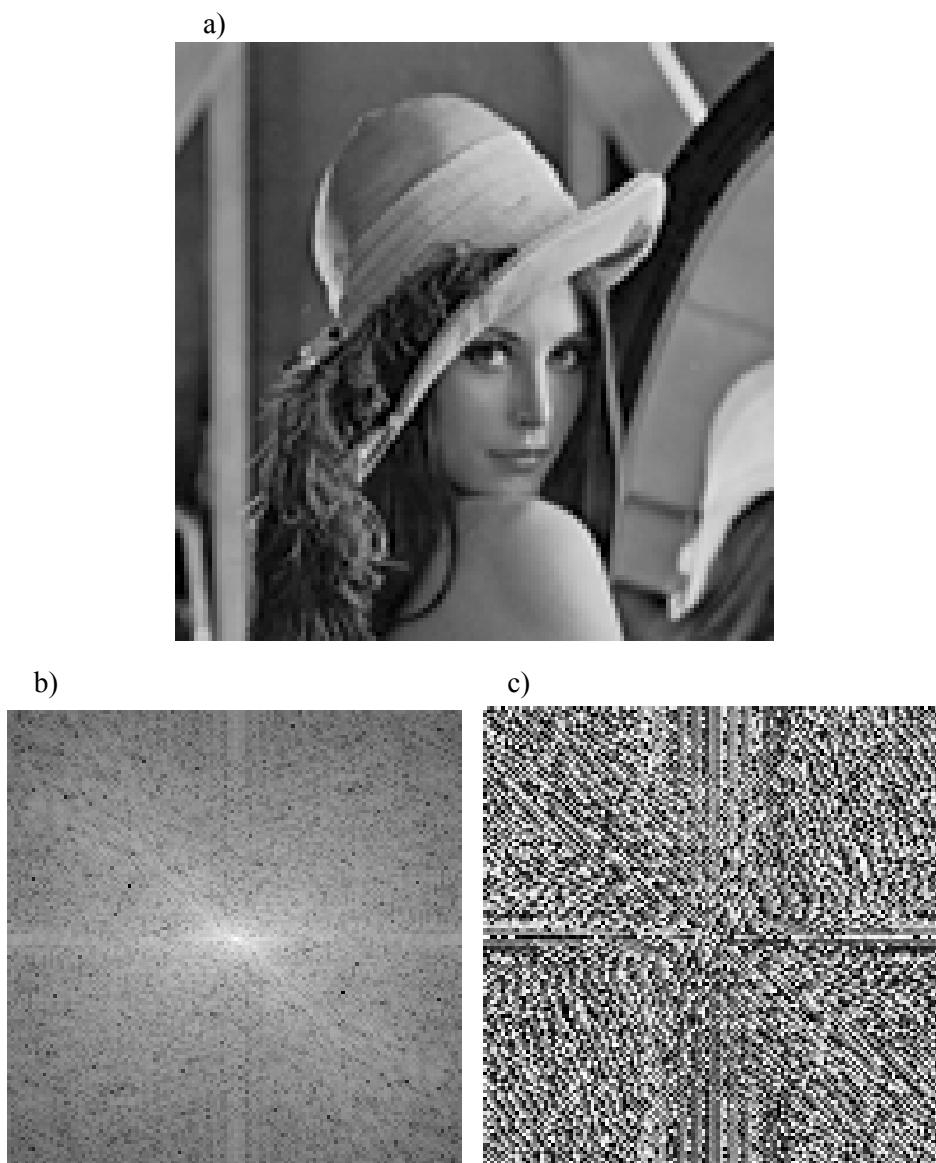
$$\varphi_p(i, k) = \frac{\varphi(i, k) - \min_{i,k}(\varphi(i, k))}{\max_{i,k}(\varphi(i, k)) - \min_{i,k}(\varphi(i, k))} \cdot 255 \quad (4.44)$$

- 2) zaokrąglenie tak otrzymanych liczb z zakresu od 0 do 255 do postaci całkowitej.

Jeżeli wynik operacji 1) nie jest nigdzie zapamiętywany, to po operacji 2) obraz oryginalny nie może już być dokładnie odtworzony, ponieważ po operacji 2) reprezentacja w postaci  $F$ -obrazu jest nieodwracalnie zniekształcona. Często jednak można szybko zorientować się w pewnych cechach  $F$ -obrazu właśnie korzystając z wizualizacji w postaci poziomów szarości (lub wizualizacji różnokolorowej). W przypadku amplitudy zazwyczaj minimalna wartość przed skalowaniem (4.43) niewiele różni się od zera, więc zależność (4.43) zastępuje się uproszczoną:

$$A_p(i, k) = \frac{A(i, k)}{\max_{i,k}(A(i, k))} \cdot 255 \quad (4.45)$$

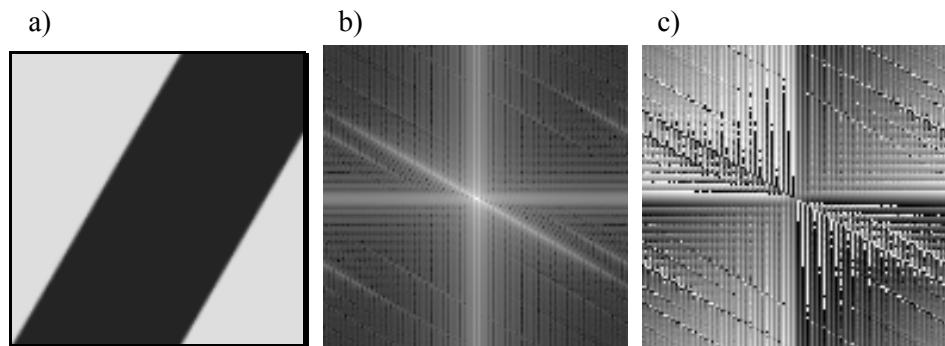
W przypadku fazy, ze względu na odpowiednie symetrie, średnia wartość znajduje się dokładnie w środku przedziału, zatem poziom szarości odpowiadający liczbie 127 reprezentuje zerową fazę. Rysunek 4.17 pokazuje po raz kolejny obraz „Lena” w rozdzielczości 128x128 pikseli oraz reprezentację tego obrazu w dziedzinie  $F$ , w postaci dwóch obrazów: dla amplitudy i dla fazy. Warto i tu zaobserwować symetrie w obu częściach  $F$ -obrazu.



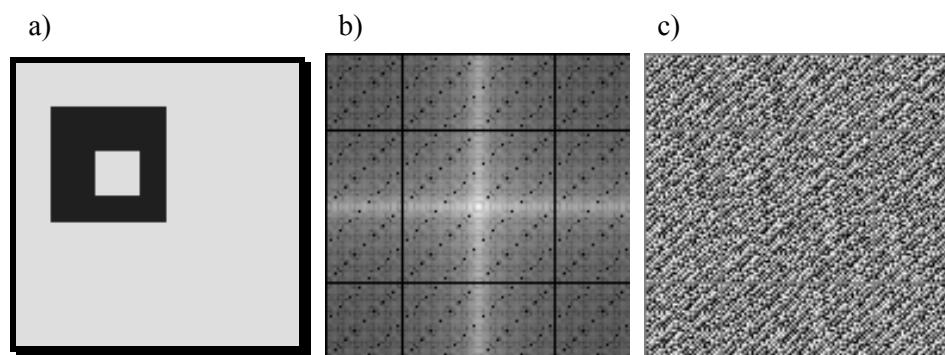
Rys. 4.17. Obraz „Lena”, o rozdzielczości 128x128:  
a) obraz oryginalny, b) amplituda  $F$ -obrazu, c) faza  $F$ -obrazu.

### 4.3.3 Przykłady charakterystycznych związków pomiędzy treścią obrazu i F-obrazem przedstawionym w postaci poziomów szarości

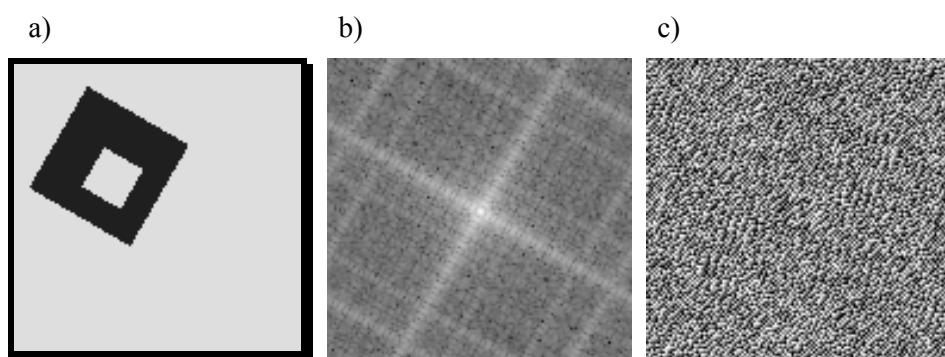
W rozdziale tym będzie pokazane, w jaki sposób pewne cechy obrazu mogą być obserwowane na jego  $F$ -obrazie oglądany w postaci poziomów szarości. Jako przykłady wykorzystane będą obrazy o rozdzielcości  $128 \times 128$  pikseli. Rysunek 4.18 prezentuje obraz o bardzo prostej treści, dla którego szczególnie wyraźnie widoczne są symetrie amplitudy i fazy  $F$ -obrazu. Bardzo czytelny jest także związek pomiędzy kierunkiem krawędzi w obrazie (rys. 4.18a) i kierunkiem linii widocznych w amplitudzie (rys. 4.18b) i fazie (rys. 4.18c)  $F$ -obrazu. Kierunki te są wzajemnie prostopadłe. Rysunki 4.19-4.21 pokazują dalsze przykłady związków pomiędzy ułożeniem krawędzi w obrazie i liniami widocznymi szczególnie wyraźnie w amplitudzie  $F$ -obrazu. Rysunek 4.21 ilustruje ponadto, jak dwa elementy obrazu wnoszą niezależnie swoje cechy do  $F$ -obrazu. Efekt sumowania w dziedzinie  $F$  jest wyraźnie widoczny dzięki temu, że oba kwadraty posiadają ten sam kolor (poziom szarości). Na rysunkach 4.22-4.24 pokazano natomiast, jak ten sam kształt, umieszczony w różnych położeniach na obrazie, wpływa na widok  $F$ -obrazu. Tym razem warto zwrócić uwagę na zmiany w fazie. Z kolei rysunek 4.25 zamieszczono dla porównania z rys. 4.22 - oba obrazy zawierają okręgi o tej samej grubości, ale o różnych średnicach; oba umieszczone centralnie. Obserwując rysunki 4.18-4.25 można dojść do wniosku, że w dziedzinie Fouriera zaobserwować można elementy analogiczne do elementów obrazu oryginalnego. Zatem można by w dziedzinie Fouriera zastosować odpowiednie filtry, wycinające wybrane pasmo - na przykład biegnące w skos przez pole  $F$ -obrazu – i w ten sposób zostaną wyeliminowane krawędzie obrazu oryginalnego o tym samym nachyleniu. Owszem, można tak zrobić, jednak o wiele prościej jest te krawędzie eliminować bezpośrednio w dziedzinie obrazu. Ponadto ingerując w obraz w jego pierwotnej dziedzinie można od razu obserwować, jak dana operacja wpływa na treść obrazu, podczas gdy dokonując manipulacji w dziedzinie  $F$  nigdy nie można być pewnym, czy nie powstanie szkodliwy wizualnie efekt uboczny tych manipulacji. Nieco później pokażemy odpowiednie przykłady, a teraz przedstawiony zostanie jeden z nich - połączenia obrazu „Lena” z okręgiem z rysunku „okrag\_3” - rysunek 4.26. Ciekawe, jaki procent czytelników wolałby stać przed zadaniem zlokalizowania i ewentualnego usunięcia obiektu znieksztalcającego obraz „Lena” w oparciu o  $F$ -obraz, a ilu wolałoby pracować na obrazie w dziedzinie oryginalnej?



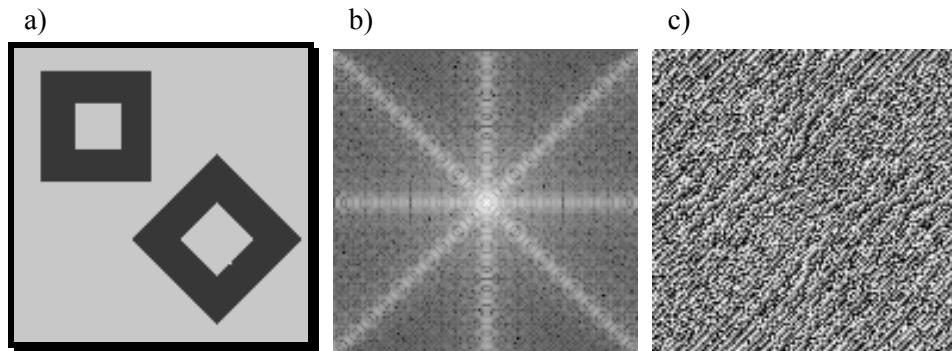
Rys. 4.18. Ilustracja powiązania wyróżnionych kierunków w obrazie i  $F$ -obrazie  
a) obraz „pasek”, b) amplituda obrazu „pasek”, c) faza obrazu „pasek”.



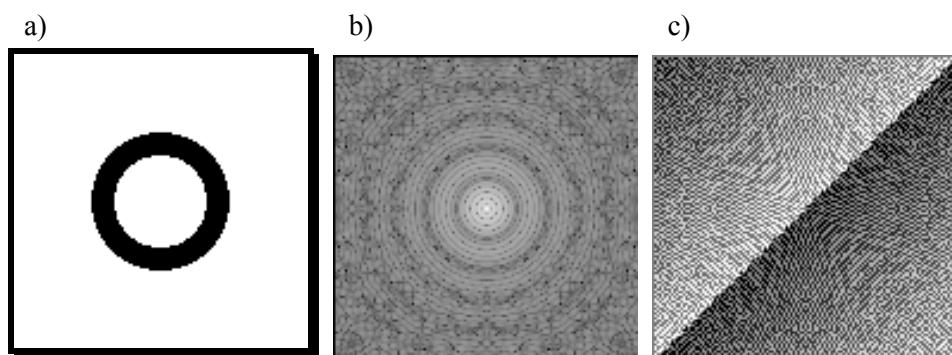
Rys. 4.19. Ilustracja związku pomiędzy ułożeniem krawędzi na obrazie i widokiem odpowiedniego  $F$ -obrazu:  
a) obraz „kwadrat2”, b) amplituda obrazu „kwadrat2”, c) faza obrazu „kwadrat2”.



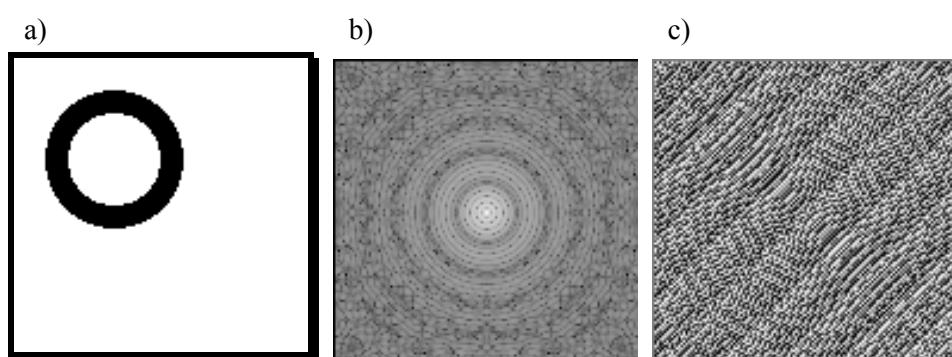
Rys. 4.20. Ilustracja związku pomiędzy ułożeniem krawędzi na obrazie i widokiem odpowiedniego  $F$ -obrazu:  
a) obraz „kwadrat3”, b) amplituda obrazu „kwadrat3”, c) faza obrazu „kwadrat3”.



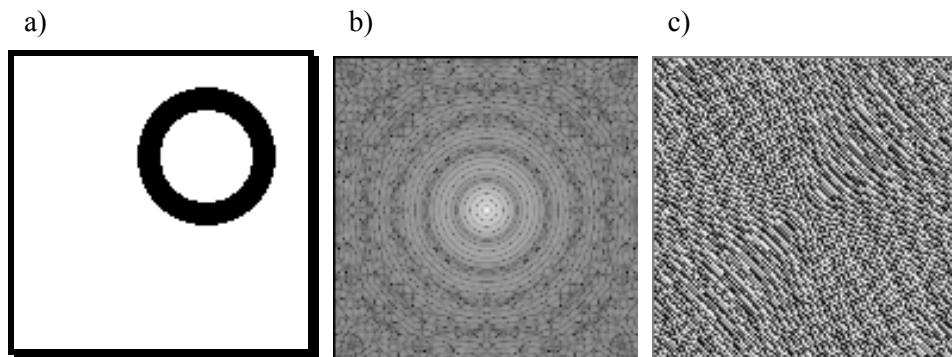
Rys. 4.21. Ilustracja związku pomiędzy ułożeniem krawędzi na obrazie i widokiem odpowiedniego  $F$ -obrazu: a) obraz „dwa\_kwadraty”, b) amplituda obrazu „dwa\_kwadraty”, c) faza obrazu „dwa\_kwadraty”.



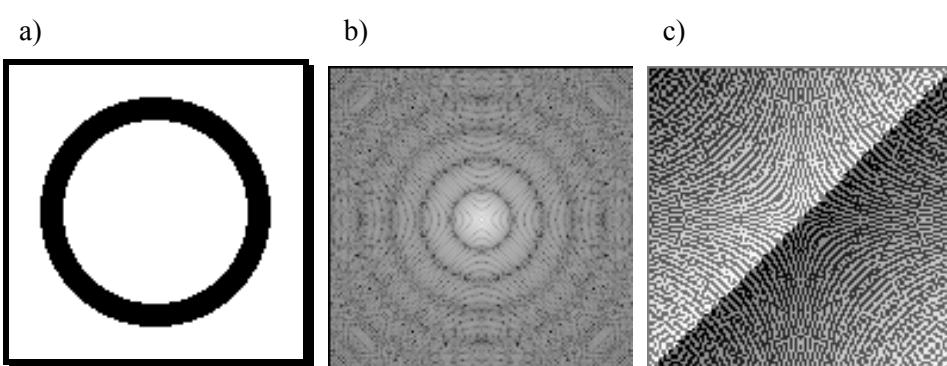
Rys. 4.22. Ilustracja związku pomiędzy ułożeniem okręgu na obrazie i widokiem odpowiedniego  $F$ -obrazu:  
a) obraz „okrag\_1”, b) amplituda obrazu „okrag\_1”, c) faza obrazu „okrag\_1”.



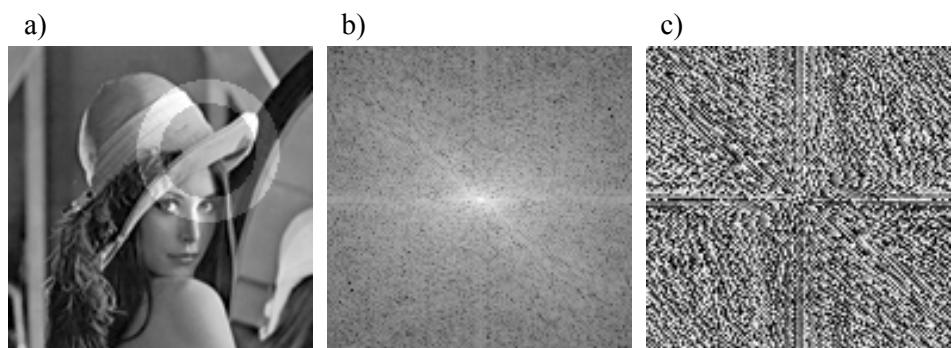
Rys. 4.23. Ilustracja związku pomiędzy ułożeniem okręgu na obrazie i widokiem odpowiedniego  $F$ -obrazu:  
a) obraz „okrag\_2”, b) amplituda obrazu „okrag\_2”, c) faza obrazu „okrag\_2”.



Rys. 4.24. Ilustracja związku pomiędzy ułożeniem okręgu na obrazie i widokiem odpowiedniego  $F$ -obrazu:  
a) obraz „okrag\_3”, b) amplituda obrazu „okrag\_3”, c) faza obrazu „okrag\_3”.



Rys. 4.25. Ilustracja związku pomiędzy ułożeniem oraz średnicą okręgu na obrazie i widokiem dąpowiedniego  $F$ -obrazu:  
a) obraz „okrag\_4”, b) amplituda obrazu „okrag\_4”, c) faza obrazu „okrag\_4”.



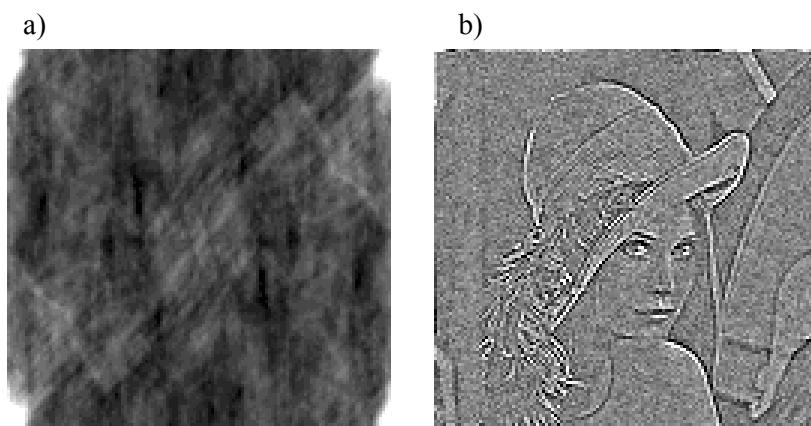
Rys. 4.26. Obraz „Lena” zniekształcony przez umieszczenie na nim okręgu z obrazu „okrag\_3”:  
a) obraz, b) amplituda obrazu, c) faza obrazu.

#### 4.3.4 Zawartość informacji wizualnej w poszczególnych składowych F-obrazu

Oglądając podane przykłady można dojść do wniosku, że główna informacja zawarta w obrazie - kształty okręgów, prostokątów, nachylenia krawędzi - reprezentowana jest przez amplitudową część *F*-obrazu. Dlatego też warto przeprowadzić eksperyment rozdzielenia amplitudy i fazy *F*-obrazu i obejrzeć rezultat wykonania odwrotnej transformacji z tak powstałych *F*-obrazów. Jeżeli zachowane zostaną odpowiednie symetrie, to zarówno obraz odtworzony wyłącznie w oparciu o informację amplitudową, jak i fazową będzie posiadał jedynie część rzeczywistą. Amplitudową i fazową część *F*-obrazu można otrzymać przeprowadzając poniższe operacje dla wszystkich indeksów *i* oraz *k*

$$A(i, k) = |F(i, k)| \quad (4.46)$$

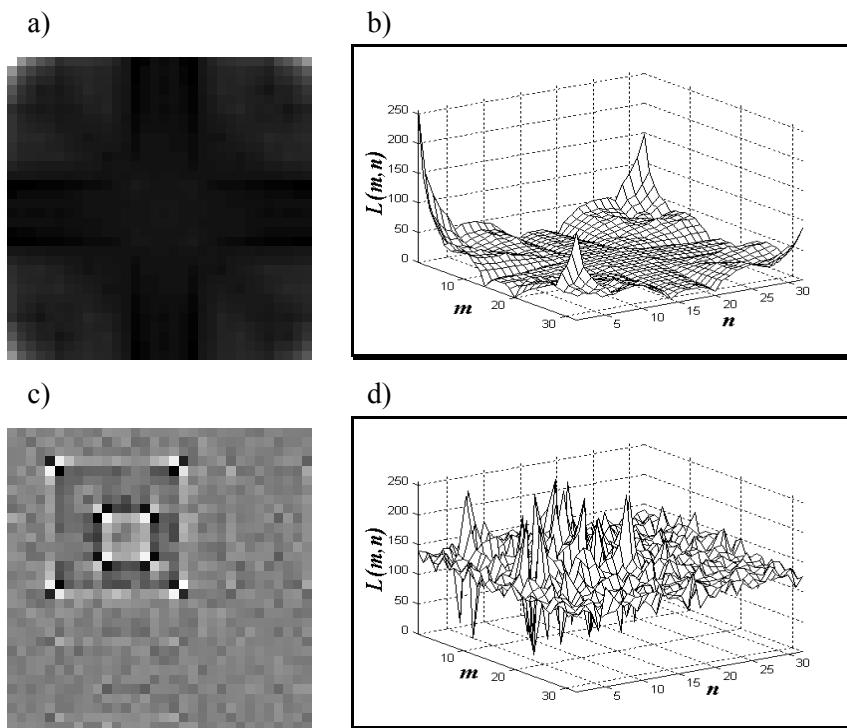
$$\varphi(i, k) = \frac{F(i, k)}{A(i, k)} \quad (4.47)$$



Rys. 4.27. Obraz odtworzony dla oryginalnego obrazu „Lena”, 128x128: a) w oparciu o informację amplitudową, po rozciągnięciu do pełnego zakresu poziomów szarości przedziału 0-20, b) w oparciu o informację fazową, po rozciągnięciu do pełnego zakresu poziomów szarości przedziału 40-175.

Obrazy otrzymane poprzez odwrotną transformację Fouriera z *F*-obrazów (4.46) i (4.47) są wprawdzie zawsze rzeczywiste, jednak ich wartości są zazwyczaj bardzo źle dopasowane do przedziału 0-255. W związku z tym należy dobrą, najlepiej metodą obserwacji i porównań, nowy zakres, tak by na odtworzonym obrazie zarysowały się kształty zawarte w obrazie oryginalnym. Wynik powyższego eksperymentu dla obrazu „Lena”, 128x128 pikseli, pokazano na rysunku 4.27. W obu przypadkach najpierw cały odtworzony zakres wartości pikseli został odwzorowany na zakres 0-255 i następnie z tak otrzymanego zakresu wy-

bierano fragment, którego rozciągnięcie do zakresu 0-255 dawało subiektywnie najlepsze wyniki. „Subiektywnie” oznacza tutaj, że głównym kryterium było w tym przypadku wrażenie wizualne. Rysunek 4.28 demonstruje wynik analogicznych zobrazowań dla obrazu „kwadrat” - patrz rys. 4.16a - o rozdzielczości 32x32 piksele.

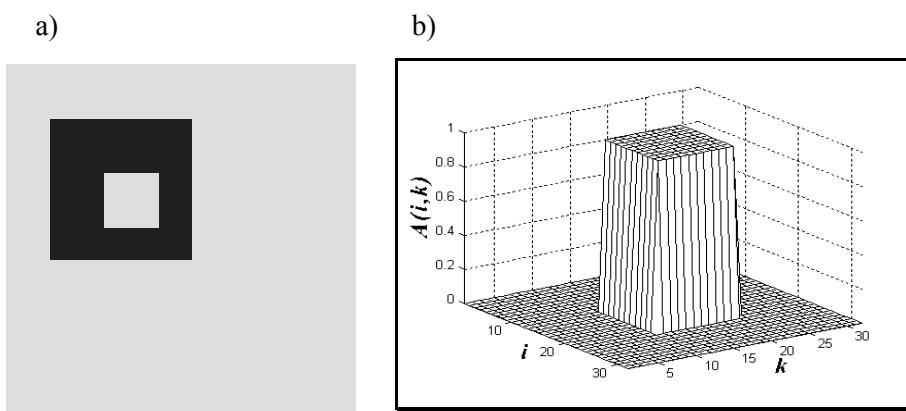


Rys. 4.28. Obraz odtworzony dla oryginalnego obrazu „kwadrat”, 32x32, w obu przypadkach po dostosowaniu zakresu wartości do przedziału 0-255, bez dodatkowego dopasowywania wartości: a) obraz odtworzony w oparciu o informację amplitudową, b) wykres funkcji obrazu z rysunku a), c) obraz odtworzony w oparciu o informację fazową, d) wykres funkcji obrazu z rysunku c).

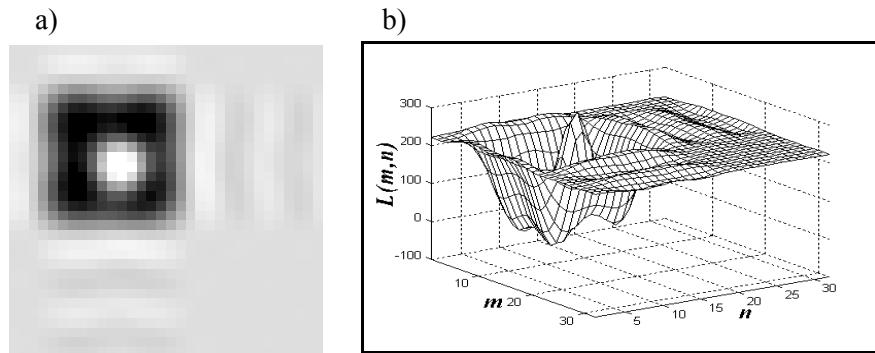
Wynik porównania wypada dla obu testów zdecydowanie na korzyść fazy. Należy jednak pamiętać, że amplituda otrzymana według zależności (4.46) stanowi  $F$ -obraz rzeczywisty. Zatem, biorąc pod uwagę podobieństwo transformacji prostej i odwrotnej, można wywnioskować, że po odtworzeniu nastąpiło przestrzenne nałożenie się informacji zawartej pierwotnie w obrazie, tak by obraz odtworzony w dziedzinie pierwotnej przejawiał odpowiednie symetrie - widoczne szczególnie wyraźnie na rys. 4.28b. Zatem obrazy na rysunkach 4.27a i 4.28a zawierają nałożone cztery wersje oryginalnego obrazu przekształcone geometrycznie przez odpowiednie symetrie. Całość jest zupełnie nieczytelna. Takie nałożenie się informacji, praktycznie uniemożliwiające odtworzenie informacji oryginalnej, nosi nazwę „aliasingu”.

### 4.3.5 Filtracja obrazów w dziedzinie Fouriera

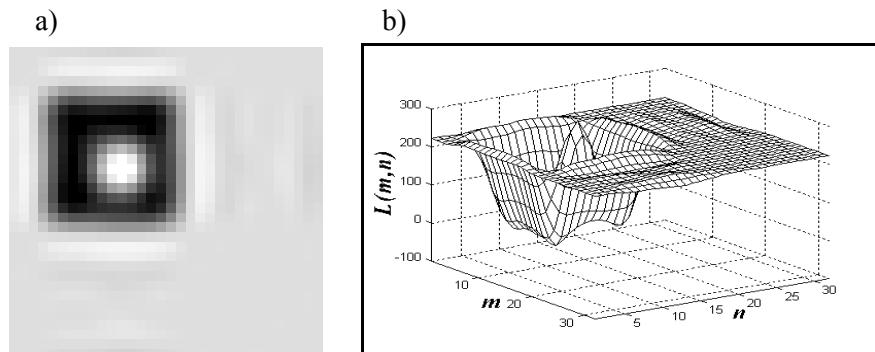
W wielu dziedzinach bardzo popularnym narzędziem przetwarzania sygnałów są filtry cyfrowe. Również w przypadku przetwarzania obrazów cyfrowych można zdefiniować pojęcie filtru oparte na właściwościach  $F$ -obrazów i związkach pomiędzy  $F$ -obrazami i obrazami. Filtr cyfrowy przeznaczony do filtracji obrazów można zinterpretować jako  $F$ -obraz, a filtrację jako mnożenie punktowe dwóch  $F$ -obrazów - jednego, pochodzącego od filtrowanego obrazu, i drugiego, będącego filtrem. Dla przykładu weźmy obraz „kwadrat”, który wraz z odpowiadającym mu  $F$ -obrazem przedstawiono na rysunku 4.16. Sam obraz powtórzono na rysunku 4.29a. Rysunek 4.29b pokazuje wykres amplitudy  $F$ -obrazu, który zostało potraktowane jako filtr. Faza filtru jest wszędzie równa zero. Amplituda filtru przyjmuje tylko dwie wartości 0 i 1. Obszar wypełniony jedynkami będziemy dalej nazywali oknem częstotliwościowym. Oznacza to, że po pomnożeniu części  $F$ -pikseli obrazu „kwadrat” pozostanie wyzerowana, a część nie ulegnie zmianie. Dla filtru z rysunku 4.19b zachowane zostaną tylko składowe obrazu „kwadrat” otaczające składową stałą, czyli odpowiadające niższym częstotliwościom. Można zatem taki filtr nazwać dolnoprzepustowym. Po wykonaniu mnożenia w dziedzinie  $F$  powrócono do dziedziny pierwotnej i rysunek 4.30 pokazuje obraz oraz jego wykres, otrzymany po filtracji za pomocą przykładowego filtru dolnoprzepustowego. Wartości otrzymanej w dziedzinie pierwotnej funkcji wykraczają poza zakres 0-255 i przed prezentacją w postaci poziomów szarości (rys. 4.30a) wartości przekraczające dopuszczalny zakres musiały zostać odpowiednio ograniczone. Charakterystyczną cechą idealnych filtrów - określonych w postaci  $F$ -pikseli o wartościach 0 i 1 - jest powstanie oscylacji w obrazie otrzymanym po filtracji.



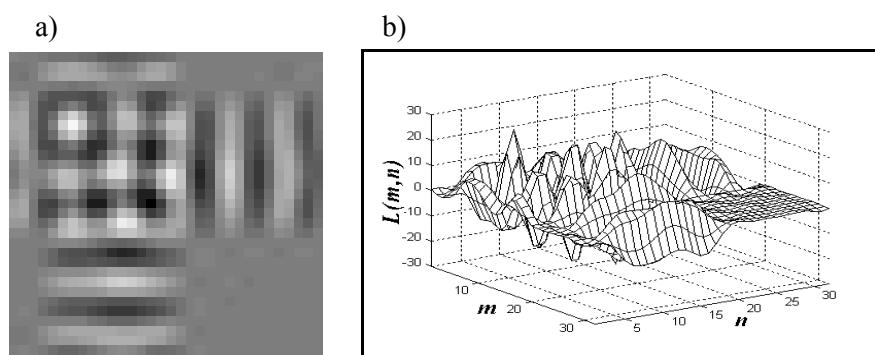
Rys. 4.29. Z lewej obraz „kwadrat”, z prawej wykres amplitudy  $F$ -obrazu, który został wykorzystany jako filtr dolnoprzepustowy.



Rys. 4.30. Obraz „kwadrat” oraz jego wykres otrzymany po filtracji filtrem z rys. 4.29b.



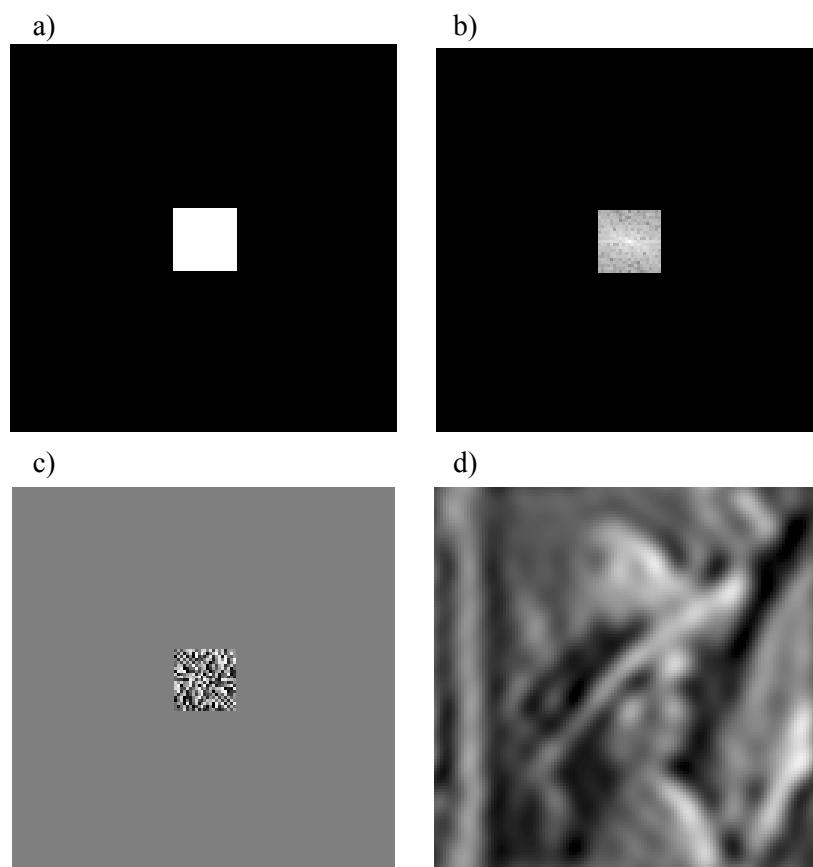
Rys. 4.31. Obraz wraz z wykresem odtworzony z części rzeczywistej powstałej w dziedzinie pierwotnej po filtracji obrazu „kwadrat” filtrem podobnym do przedstawionego na rys. 4.29b, ale z przesuniętym obszarem wypełnionym jedynkami.



Rys. 4.32. Obraz wraz z wykresem odtworzony z części urojonej powstałej w dziedzinie pierwotnej po filtracji obrazu „kwadrat” filtrem podobnym do przedstawionego na rys. 4.29b ale z przesuniętym oknem częstotliwościowym. Rysunek a) powstał po dodaniu stałej 128 do wartości funkcji przedstawionej na wykresie b).

Filtr z rysunku 4.29a spełnia opisane wcześniej warunki symetrii. Stosunkowo łatwo jest jednak pomylić się w położeniu jedynek filtru. Rysunki 4.31 i 4.32 pokazują rezultat filtracji filtrem z celowo przesuniętym oknem częstotliwościowym o jeden  $F$ -piksel w pionie i o jeden w poziomie. W rezultacie w dziedzinie pierwotnej powstał obraz zespolony - z częścią rzeczywistą i urojoną. Powstanie części urojonej obrazu jest zazwyczaj efektem niepożądanym, należy go więc unikać.

Na rysunkach od 4.33 do 4.40 pokazano wyniki filtracji dwóch przykładowych obrazów „Lena” (patrz rys. 4.17a) oraz „dwa\_kwadraty” (patrz rys. 4.21a) różnymi filtrami zdefiniowanymi w postaci zero-jedynkowych  $F$ -obrazów. Dzięki temu, że filtry są zero-jedynkowe, obrazy otrzymane po filtracji pokazują, jaka zawartość informacji wizualnej zawarta jest w zachowanej części  $F$ -obrazu. Taka obserwacja stanowi ważne uzupełnienie dotychczasowych rozważań na temat związków między obrazem i  $F$ -obrazem.

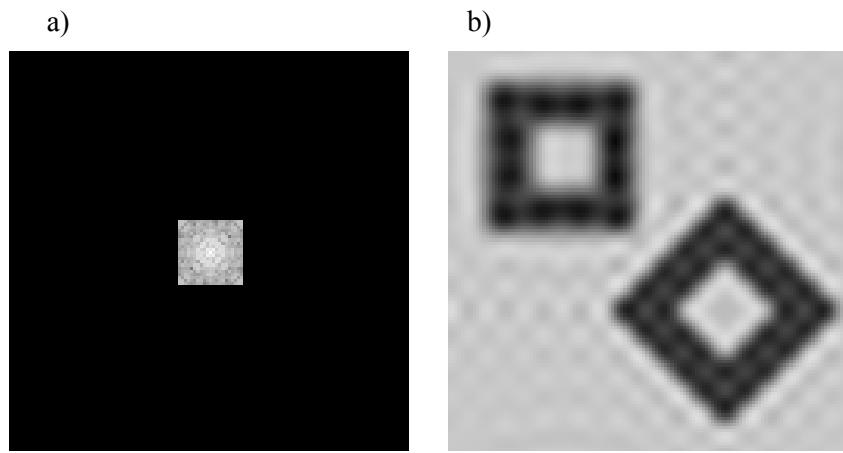


Rys. 4.33. Filtracja dolnoprzepustowa obrazu „Lena”: a)  $F$ -obraz filtru, b) amplituda  $F$ -obrazu „Lena” po filtracji, c) faza  $F$ -obrazu „Lena” po filtracji, d) obraz po filtracji otrzymany w dziedzinie pierwotnej.

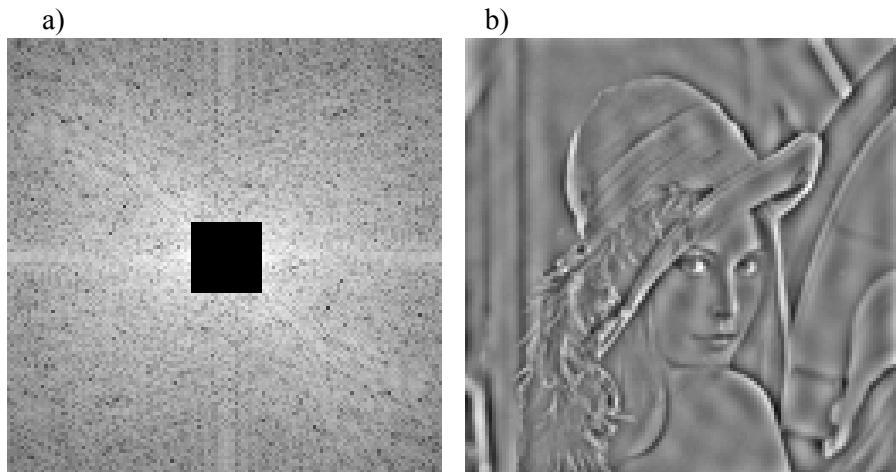
Rysunek 4.33a przedstawia  $F$ -obraz filtru w postaci poziomów szarości. Ponieważ  $F$ -obraz jest zero-jedynkowy, więc by cokolwiek było widać, wszystkie wartości zostały pomnożone przez stałą 255. Ponadto w przypadku  $F$ -obrazu zawierającego wyłącznie 0 i 1 część rzeczywista jest zawsze równa amplitudzie, a faza oraz część urojona są wszędzie równe zero. Dlatego też dla pełnego opisu takiego filtru w postaci  $F$ -obrazu wystarczy podać na przykład jego część rzeczywistą - analogicznie jak dla obrazu cyfrowego w dziedzinie pierwotnej. Po wykonaniu filtracji, czyli mnożenia  $F$ -obrazów,  $F$ -obraz wynikowy będzie zawierał zera wszędzie tam, gdzie filtr posiadał wartości zerowe, natomiast  $F$ -piksele filtrowanego obrazu (zwykle o wartościach zespolonych) przypadające na obszar okna, mnożone przez wartość 1, pozostaną nie zmienione. Dlatego też, jeżeli przedstawi się amplitudę lub fazę  $F$ -obrazu po filtracji, to zwykle od razu widoczne jest, jak musiał wyglądać filtr. Wynika to z faktu, że niezwykle rzadko  $F$ -obraz pochodzący z obrazu cyfrowego o sensowej treści wizualnej zawiera obszary o zerowej amplitudzie czy fazie.

Zatem wyzerowanie amplitudy lub fazy w danym obszarze  $F$ -obrazu filtrowanego jest zwykle wyraźnie widoczne i świadczy o zerach występujących w analogicznym obszarze  $F$ -obrazu będącego filtrem. Rysunki 4.33b oraz 4.33c w powiązaniu z rys. 4.33a stanowią ilustrację powyższego wywodu. Rysunek 4.33d pokazuje obraz odtworzony po przeprowadzeniu filtracji dolnoprzepustowej, demonstrując jednocześnie, jaka część informacji wizualnej obrazu „Lena” zawarta jest w szczegółowej, dolnopasmowej części  $F$ -obrazu.

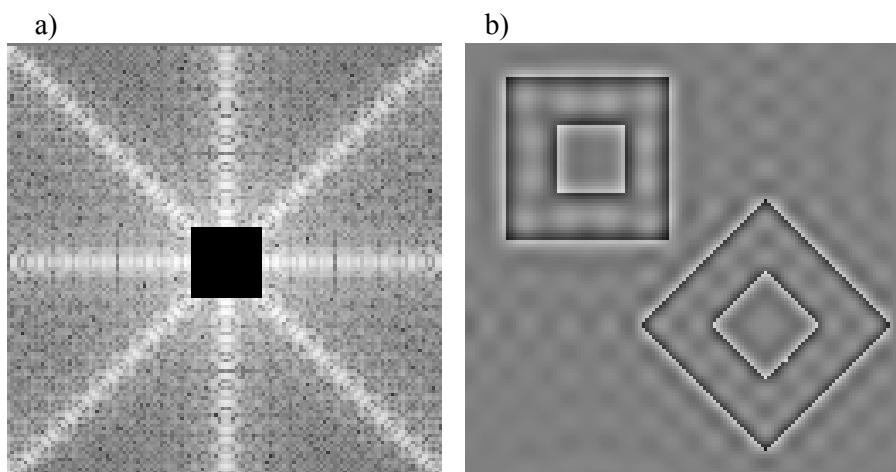
Dolnoprzepustowa filtracja obrazu „dwa\_kwadraty” zilustrowana jest już tylko w postaci amplitudy  $F$ -obrazu po filtracji oraz otrzymanego po filtracji obrazu - rysunek 4.34. Z rysunku 4.34a można odczytać postać filtru, z kolei znając  $F$ -obraz filtru i fazę pokazaną na rys. 4.21c można zobaczyć, jak wygląda zachowana przez filtr część fazy.



Rys. 4.34. Filtracja dolnoprzepustowa obrazu „dwa\_kwadraty”: a) amplituda  $F$ -obrazu po filtracji, b) obraz w dziedzinie pierwotnej otrzymany po filtracji.



Rys. 4.35. Filtracja górnoprzepustowa obrazu „Lena”:  
a) amplituda  $F$ -obrazu po filtracji, b) obraz w dziedzinie pierwotnej otrzymany po filtracji (oraz po dodaniu stałej 128).



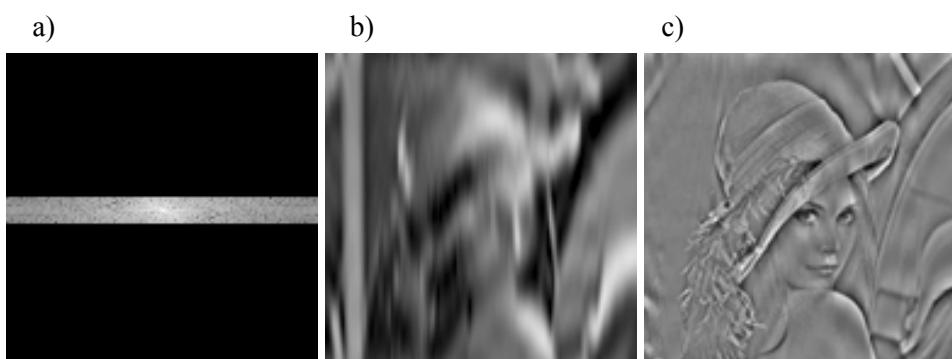
Rys. 4.36. Filtracja górnoprzepustowa obrazu „dwa\_kwadraty”:  
a) amplituda  $F$ -obrazu po filtracji, b) obraz w dziedzinie pierwotnej otrzymany po filtracji (oraz po dodaniu stałej 128).

Rysunki 4.35 i 4.36 pokazują zastosowanie filtru górnoprzepustowego. Jest to filtr komplementarny w stosunku do wykorzystanego powyżej filtru dolnoprzepustowego. Oznacza to, że w  $F$ -obrazie filtru dolnoprzepustowego zamieniono zera na jedynki, a jedynki na zera. Jak widać, dodanie  $F$ -obrazów obu komplementarnych filtrów da  $F$ -obraz zawierający same jedynki. Taki filtr nie zmieniłby nic w filtrowanym  $F$ -obrazie. Wynika z tego, że cała informacja wizualna zawarta w obrazie może być rozłożona na dwie (lub więcej) części w oparciu o wybór fragmentów  $F$ -obrazu. W przypadku gdy filtr eliminuje składową stałą,

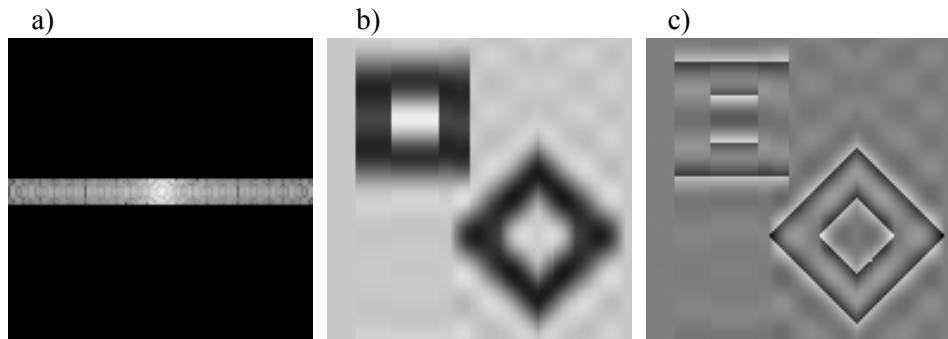
do prezentacji w postaci odcienni szarości obraz odtworzony musi mieć wartości wszystkich pikseli powiększone o pewną stałą - gdyby ten zabieg pominać, to około połowa wartości byłaby ujemna i musiałaby zostać zaokrąglona do zera. W praktyce, w przypadku 8-bitowego zapisu pikseli, do wszystkich pikseli dodaje się wartość 127 lub 128.

Nazwy filtr dolno- i górnoprzepustowy pochodzą z przetwarzania sygnałów jednowymiarowych, gdzie wystarczy podać jedną wartość oddzielającą pasmo dolne i górne. W przypadku obrazów, czyli ciągów dwuwymiarowych, można rozmaicie wybierać rozkład dolnych i górnych częstotliwości. Na rysunkach od 4.29 do 4.36 filtry określone były w postaci kwadratowych pól zer lub jedynek. Można jednak część dolnopasmową określić w postaci koła, elipsy, rombu, prostokąta itd. Za każdym razem obraz po filtracji dolnoprzepustowej będzie nieco inny. Pewnym szczególnym przypadkiem może być filtr o jakimś wyróżnionym kierunku (filtry kwadratowe wyróżniają kierunki wzdłuż przekątnej obrazu, ale w bardzo niewielkim stopniu). Można się umówić, że jeżeli obszar jedynek  $F$ -obrazu filtru będzie figurą wypukłą i będzie zawierał składową stałą, to będzie to filtr dolnoprzepustowy, natomiast filtr będący dopełnieniem filtru dolnoprzepustowego można nazwać filtrem górnoprzepustowym. Jest to jednak podział bardzo umowny.

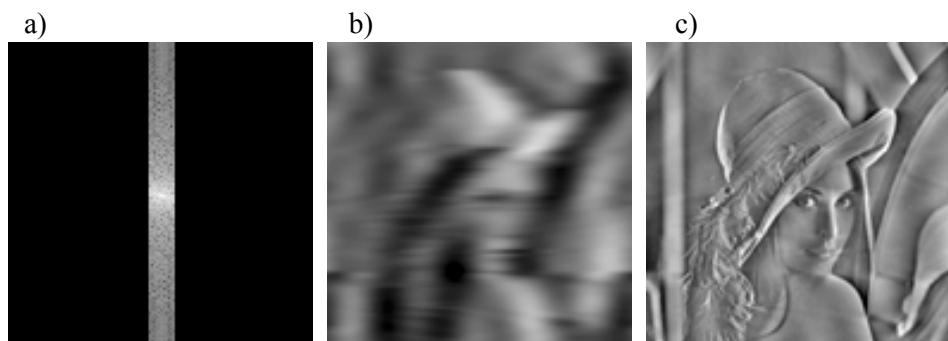
Rysunek 4.37 pokazuje wynik filtracji obrazu „Lena” filtrem dolnoprzepustowym o wyraźnej preferencji jednego kierunku oraz filtrem do niego komplementarnym. Tak określony filtr górnoprzepustowy eksponuje krawędzie poziome. Jeszcze wyraźniej jest to widoczne dla takiej samej filtracji obrazu „dwa\_kwadraty” przedstawionej na rysunku 4.38. Dla porównania rysunki 4.39 i 4.40 pokazują wynik filtracji dla obróconej o kąt prosty orientacji filtru.



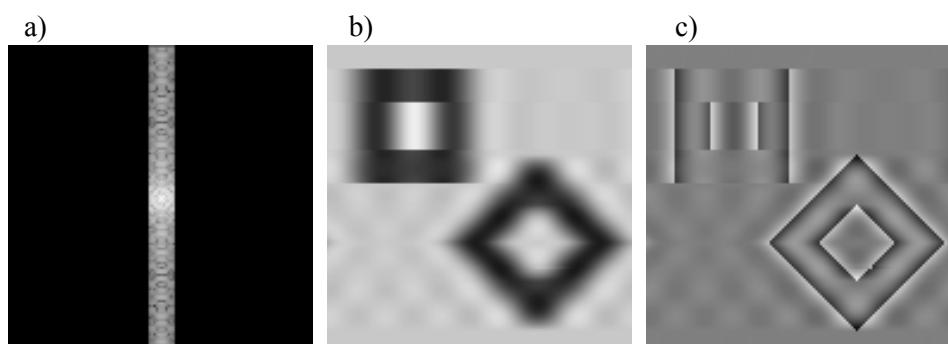
Rys. 4.37. Filtracja obrazu „Lena” z wyróżnionym kierunkiem poziomym: a) amplituda  $F$ -obrazu po filtracji dolnoprzepustowej, b) odtworzony obraz po filtracji dolnoprzepustowej, c) odtworzony obraz po komplementarnej filtracji górnoprzepustowej (oraz dodaniu stałej 128).



Rys. 4.38. Filtracja obrazu „dwa\_kwadraty” z wyróżnionym kierunkiem poziomym:  
a) amplituda  $F$ -obrazu po filtracji dolnoprzepustowej, b) odtworzony obraz po filtracji dolnoprzepustowej, c) odtworzony obraz po komplementarnej filtracji górnoprzepustowej (oraz dodaniu stałej 128).

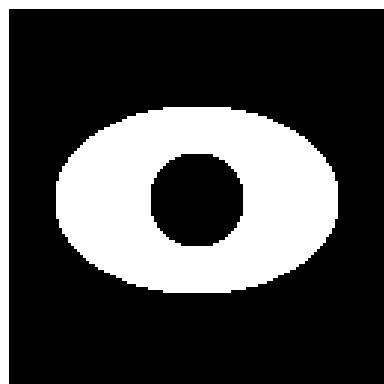


Rys. 4.39. Filtracja obrazu „Lena” z wyróżnionym kierunkiem pionowym: a) amplituda  $F$ -obrazu po filtracji dolnoprzepustowej, b) odtworzony obraz po filtracji dolnoprzepustowej, c) odtworzony obraz po komplementarnej filtracji górnoprzepustowej (oraz dodaniu stałej 128).

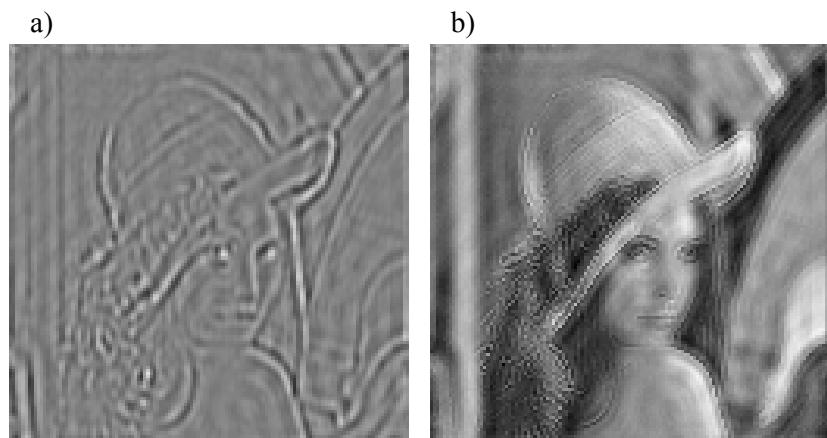


Rys. 4.40. Filtracja obrazu „dwa\_kwadraty” z wyróżnionym kierunkiem pionowym:  
a) amplituda  $F$ -obrazu po filtracji dolnoprzepustowej, b) odtworzony obraz po filtracji dolnoprzepustowej, c) odtworzony obraz po komplementarnej filtracji górnoprzepustowej (oraz dodaniu stałej 128).

Filtracja zero-jedynkowa nie musi być dolno- lub górnoprzepustowa. Można wybrać dowolny fragment dwuwymiarowej dziedziny  $F$ , jako obszar jedynek lub zer filtru, pamiętając jedynie o zachowaniu odpowiednich symetrii. Rysunek 4.41 pokazuje w postaci poziomów szarości (po pomnożeniu przez 255) filtr zero-jedynkowy, który zachowuje wybrane pasmo  $F$ -obrazu. Wyniki filtracji obrazów „Lena” i „dwa\_kwadraty” za pomocą tego filtra oraz filtra komplementarnego pokazują rysunki 4.42 i 4.43.



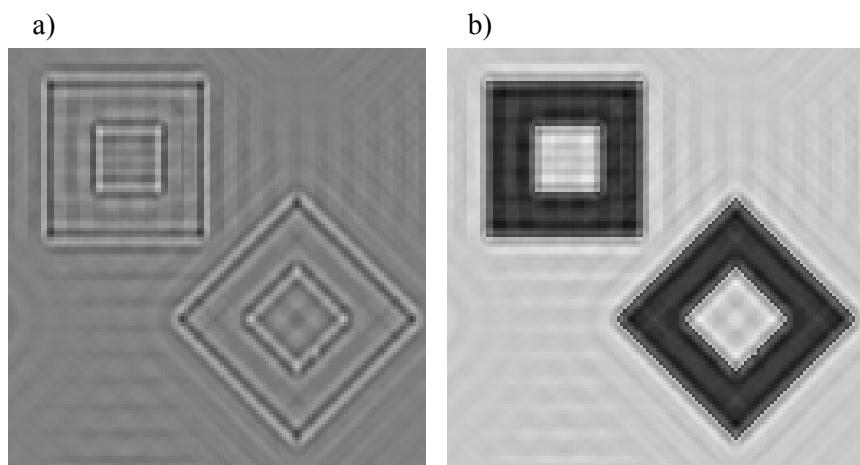
Rys. 4.41.  $F$ -obraz przykładowego filtra pasmowego (po pomnożeniu przez 255).



Rys. 4.42. Wynik filtracji obrazu „Lena”: a) filtrem z rys. 4.41, b) filtrem komplementarnym do filtru z rys. 4.41.

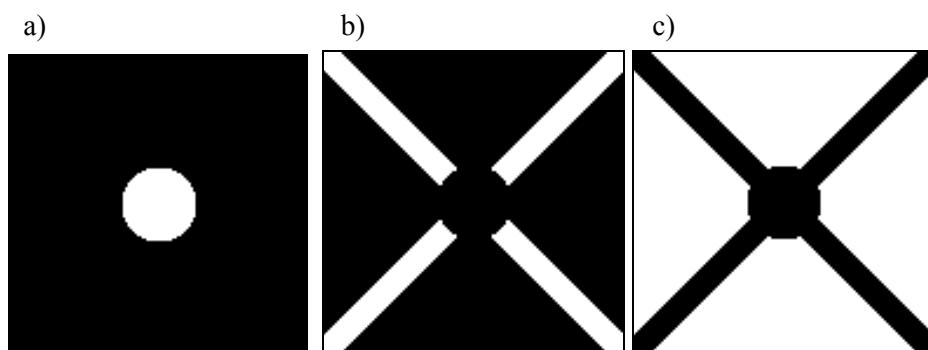
Warto jeszcze raz podkreślić, że w każdym przedstawionym przykładzie filtracji za pomocą filtrów komplementarnych dwa odtworzone obrazy zawierają łącznie pełną informację cyfrową, jaką była zawarta w obrazie pierwotnym przed filtracją. Jeżeli dodamy oba obrazy odtworzone w dziedzinie pierwotnej, zanim przeprowadzimy niezbędne do wyświetlenia w postaci poziomów szarości operacje, to otrzymamy obraz oryginalny. Jeżeli natomiast oba obrazy zostały już

dostosowane do wymogów prezentacji w postaci odcieni szarości (liczby całkowite z przedziału 0-255 w przypadku reprezentacji 8-bitowej), to po ich dodaniu obraz będzie bardzo podobny do oryginalnego, jednak mogą wystąpić widoczne różnice. Trzeba także pamiętać, że obraz w postaci poziomów szarości powstały po odfiltrowaniu składowej stałej musi zawierać dodaną wartość stałą np. 128. Zatem po dodaniu go do drugiego obrazu, powstałego po filtracji komplementarnej, trzeba tę stałą odjąć.

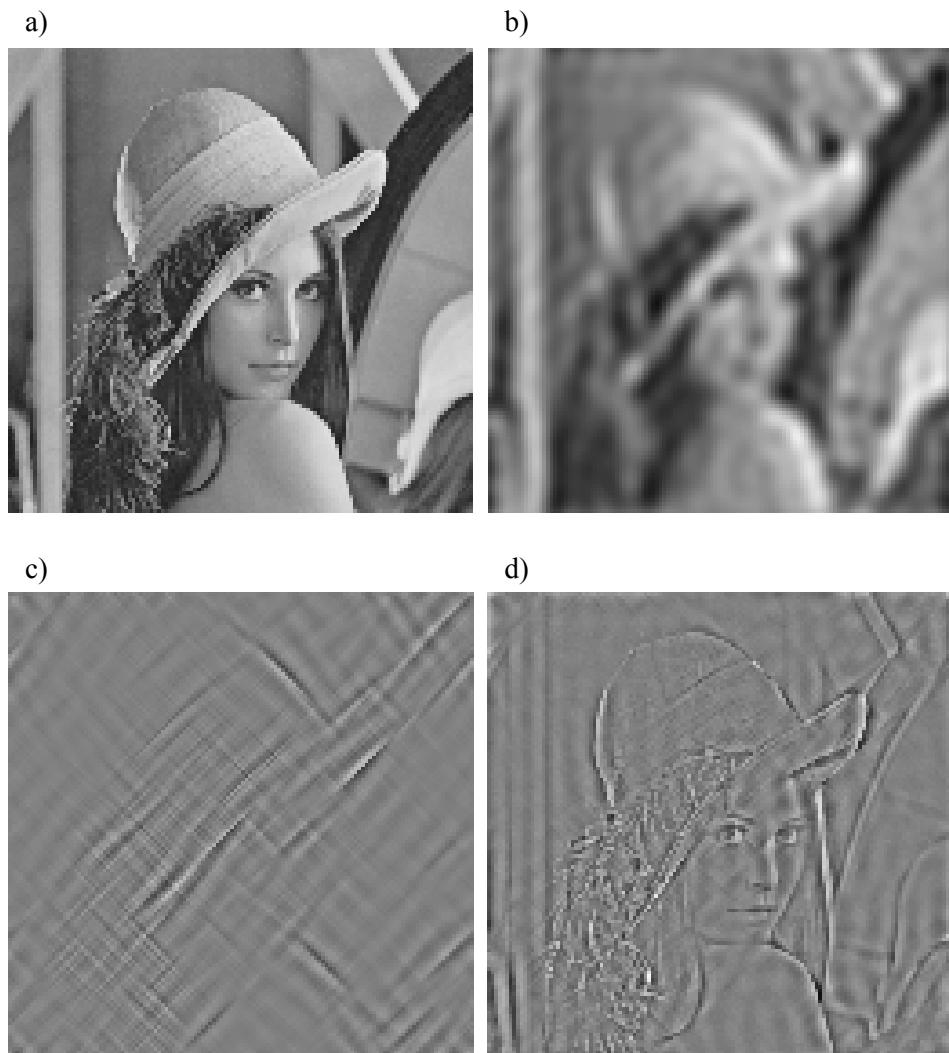


Rys. 4.43. Obraz „dwa\_kwadraty” po filtracji: a) filtrem z rys. 4.41, b) filtrem komplementarnym do filtru z rys. 4.41.

Zestaw filtrów, tworzących po dodaniu  $F$ -obraz zawierający same jedynki, może składać się z większej ilości zero-jedynkowych  $F$ -obrazów. Rysunek 4.44 pokazuje przykład zestawu trzech filtrów zero-jedynkowych rozkładających obraz cyfrowy na trzy składowe podobrazy - patrz rysunki 4.45 i 4.46. Każdy z filtrów określa wybrane podpasmo.

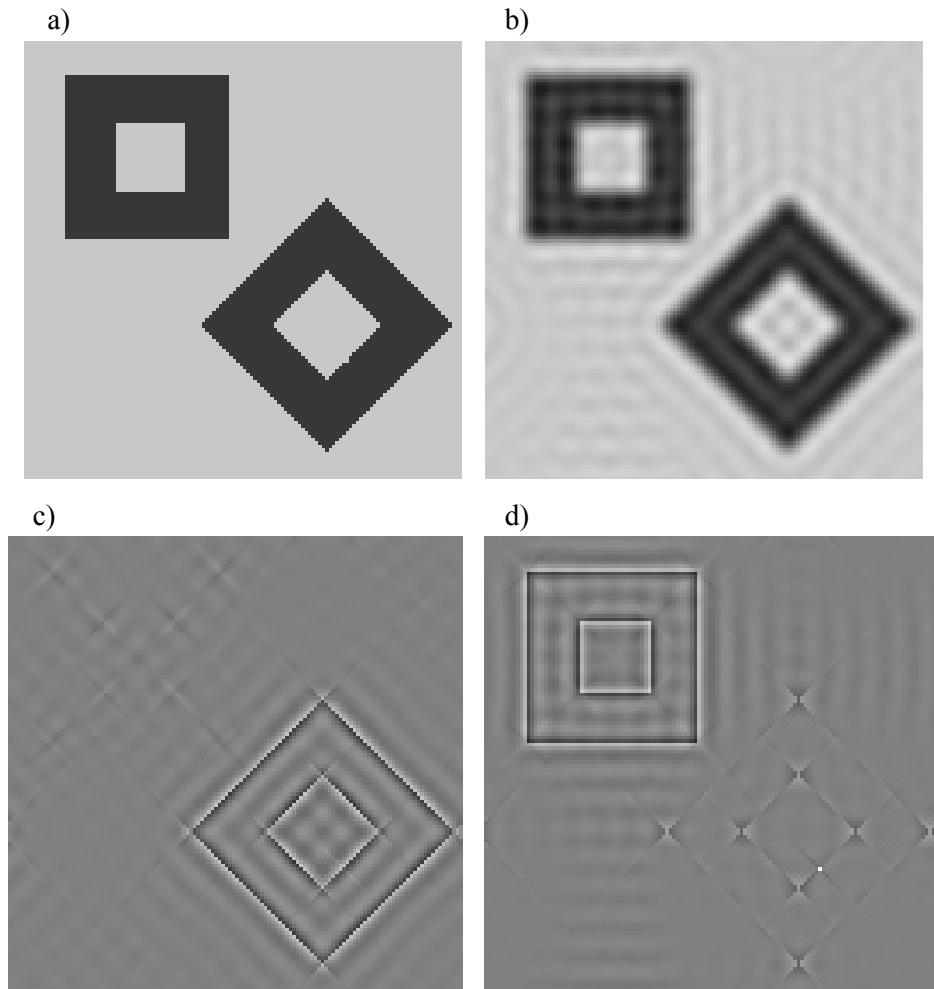


Rys. 4.44. Zestaw filtrów do rozkładu obrazu cyfrowego na trzy podpasma: a) filtr dolnoprzepustowy, b) filtr eksponujący krawędzie ukośne, c) filtr powstały jako uzupełnienie filtrów a) i b.).



Rys. 4.45. Rozkład obrazu „Lena” na trzy podobrazy w oparciu o filtry podpasmowe z rys. 4.44: a) obraz oryginalny, b) obraz otrzymany po filtracji filtrem z rys. 4.44a), c) obraz otrzymany po filtracji filtrem z rys. 4.44b) (plus stała 128), d) obraz otrzymany po filtracji filtrem z rys. 4.44c) (plus stała 128).

Łatwo sobie wyobrazić rozkłady w oparciu o rozmaite geometryczne (odpowiednio symetryczne) podziały pola  $F$ -obrazu. Trzeba jednak przestrzec optymistów, że w przypadku obrazów o wiele łatwiej jest wymyślić sobie podział obszaru na filtry podpasmowe niż uzasadnić jego użyteczność. Podział na podpasma znajduje zastosowanie głównie w dziedzinie kompresji obrazów cyfrowych.



Rys. 4.46. Rozkład obrazu „dwa\_kwadraty” na trzy podobrazy w oparciu o filtry podpasmowe z rys. 4.44: a) obraz oryginalny, b) obraz otrzymany po filtracji filtrem z rys. 4.44a, c) obraz otrzymany po filtracji filtrem z rys. 4.44b (plus stała 128), d) obraz otrzymany po filtracji filtrem z rys. 4.44c (plus stała 128).

#### 4.3.6 Niezerowa faza filtru

W dotychczasowych rozważaniach brane były pod uwagę filtry zero-jedynkowe, czyli o zerowej fazie. W przetwarzaniu sygnałów jednowymiarowych, a w szczególności sygnałów ciągłych, znany jest warunek liniowości fazy. Na kilku przykładach pokażemy teraz, że w przypadku obrazów faza powinna być nie tylko jak najbardziej zbliżona do liniowej, ale wręcz jak najbardziej zerowa. Oczywiście nie zawsze - podobnie jak w przypadku filtrów dla ciągłych sygnałów jednowymiarowych - postawiony warunek można dokładnie spełnić. Ważne jest jednak

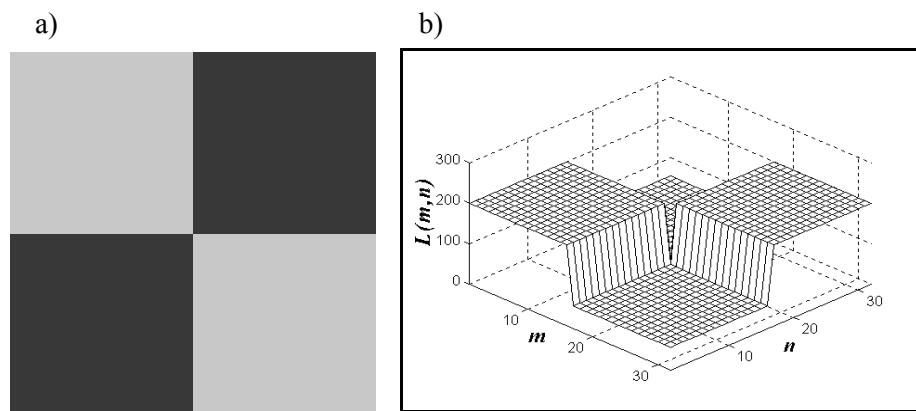
by wiedzieć, jaki jest ideał, do którego powinno się dążyć. Zasadnicza różnica pomiędzy zwykle przetwarzanymi sygnałami jednowymiarowymi i obrazem cyfrowym polega na tym, że obraz ma stosunkowo niewielki zakres zmienności indeksów. Już dość duży obraz cyfrowy rozciąga się na polu 512x512 pikseli. Jednowymiarowy ciąg wartości pochodzący z próbkowania na przykład sygnału akustycznego będzie miał co najmniej dziesiątki tysięcy wartości. Jeżeli ten ciąg jednowymiarowy zostanie przesunięty o kilka, czy nawet kilkadziesiąt, próbek w lewo lub w prawo, to zwykle po odtworzeniu w postaci dźwięku nie będzie to specjalnie dokuczliwe. Natomiast przesunięcie obrazu nawet o jeden, czy dwa piksele może w istotny sposób zmienić jego wygląd - część pikseli znajdzie się poza ramką obrazu i te „wysunięte” z jednej strony ramki piksele w wyniku cyklicznego przesunięcia „wsuną” się do obrazu z drugiej strony. Przykładowo, to, co było z lewej strony obrazu znajdzie się po prawej, a to co było na górze obrazu będzie po przesunięciu w jego dole, co daje zwykle niezbyt pożąany efekt.

Rysunek 4.47 przedstawia kolejny obraz opracowany sztucznie, specjalnie dobrany dla zilustrowania opisywanych właściwości filtracji. Na rysunku 4.48 pokazano wykres idealnie liniowej fazy filtru posiadającego amplitudę wszędzie równą 1. Fazę tę wyznaczono tak, by płaszczyzna jej wykresu przechodziła w punkcie składowej stałej przez wartość zero oraz by nachylenie tej płaszczyzny wyznaczone w kierunku wiersza wynosiło:

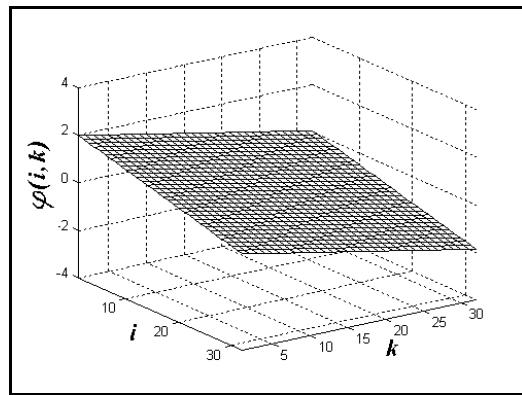
$$\left. \frac{\Delta\phi}{\Delta k} \right|_{i=\text{const}} = -\frac{2}{N} \quad (4.48)$$

natomiast w kierunku kolumny:

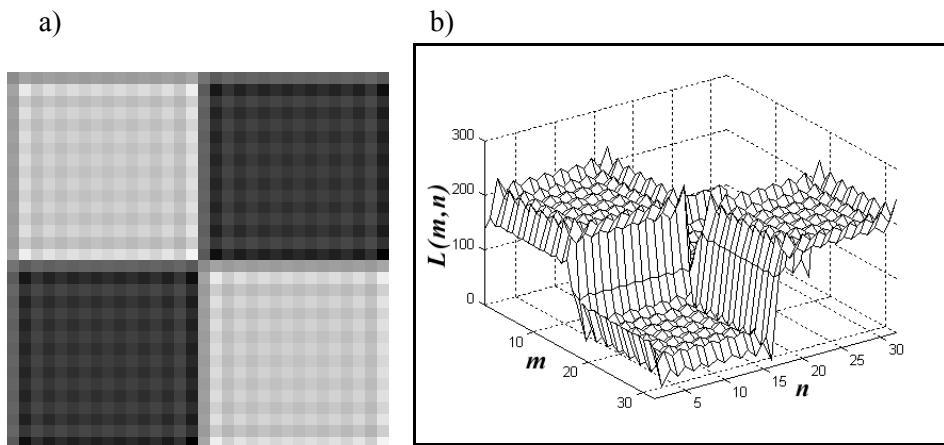
$$\left. \frac{\Delta\phi}{\Delta i} \right|_{k=\text{const}} = -\frac{2}{M} \quad (4.49)$$



Rys. 4.47. Obraz „Szach” o rozmiarach 32x32 piksele oraz jego wykres.



Rys. 4.48. Wykres fazy  $F$ -obrazu będącego filtrem o amplitudzie wszędzie równej 1.

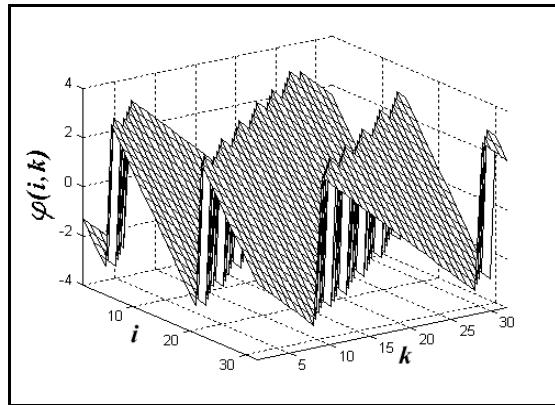


Rys. 4.49. Obraz otrzymany po filtracji filtrem jedynkowym o fazie z rys. 4.48:  
a) w postaci poziomów szarości, b) w postaci wykresu funkcji.

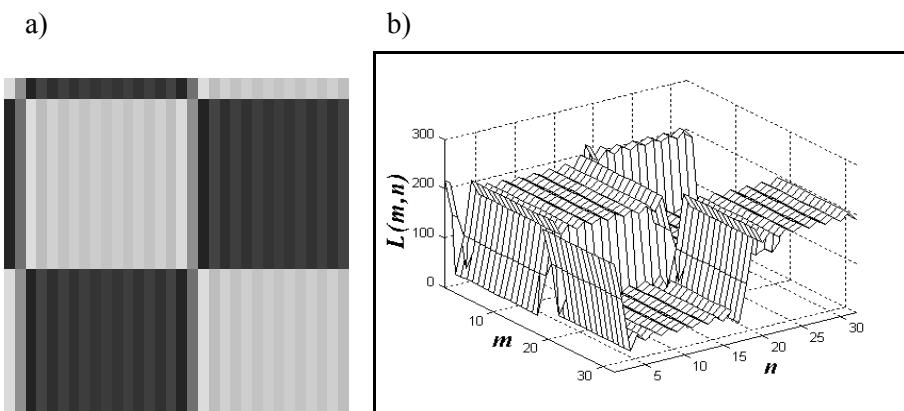
Pomimo tego, że faza filtra jest liniowa i zerowa w punkcie składowej stałej, obraz odtworzony po filtracji - patrz rys. 4.49 - zawiera nie tylko cykliczne przesunięcie w pionie i w poziomie, ale także zniekształcenia o charakterze zafalowań. Rysunek 4.50 pokazuje wykres fazy innego filtra o amplitudzie wszędzie równej jeden. W tym przypadku nachylenia funkcji fazy określono następująco:

$$\left. \frac{\Delta\varphi}{\Delta k} \right|_{i=\text{const}} = -\frac{10}{N} \quad (4.50)$$

$$\left. \frac{\Delta\varphi}{\Delta i} \right|_{k=\text{const}} = -\frac{4 \cdot \pi}{M} \quad (4.51)$$



Rys. 4.50. Wykres fazy filtru jedynkowego o nachyleniach kierunkowych według wzorów (4.50) i (4.51).



Rys. 4.51. Obraz otrzymany po filtracji filtrem jedynkowym o fazie z rys. 4.50:  
a) w postaci poziomów szarości, b) w postaci wykresu funkcji.

Wprawdzie wykres przedstawiony na rysunku 4.50 nie jest płaszczyzną, jednak faza jest liniowa. Łamany kształt wykresu wynika z faktu, że faza wyznaczona według nachyleń (4.50) i (4.51) rozciąga się poza zakres  $[-\pi, +\pi]$ . Ponieważ wszystkie zależności związane z transformacją Fouriera są okresowe względem fazy, z okresem  $2 \cdot \pi$ , dlatego zwykłe wykresy fazy przedstawia się w postaci cyklicznie sprowadzonej do zakresu  $[-\pi, +\pi]$  lub  $[0, +2 \cdot \pi]$ . Na rysunku 4.50 zastosowano wariant pierwszy. W obrazie odtworzonym po filtracji - rysunek 4.51 - zafalowania występują jedynie w kierunku poziomym (a ich efektem są pionowe pasy). Zniknęły zniekształcenia w kierunku, w którym faza miała nachylenie określone przez (4.51). Ponadto widać, że cykliczne przesunięcie w kierunku pionowym wynosi dokładnie dwa piksele. Dociekliwy Czytelnik łatwo wyjaśni zarówno efekt zafalowań, jak i przesunięcia cyklicznego, opierając się na właści-

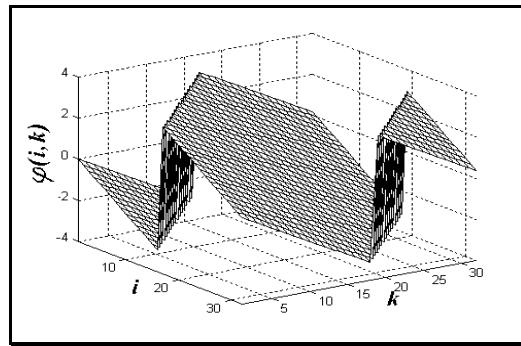
wościach próbkowanych kosinusoid, z których z kolei powstały funkcje bazowe do tworzenia  $F$ -obrazów. Reasumując, aby nie powstawały zafalowania, faza filtru powinna być liniowa o nachyleniach:

$$\left. \frac{\Delta\varphi}{\Delta k} \right|_{i=\text{const}} = -\frac{k_\varphi \cdot 2 \cdot \pi}{N} \quad (4.52)$$

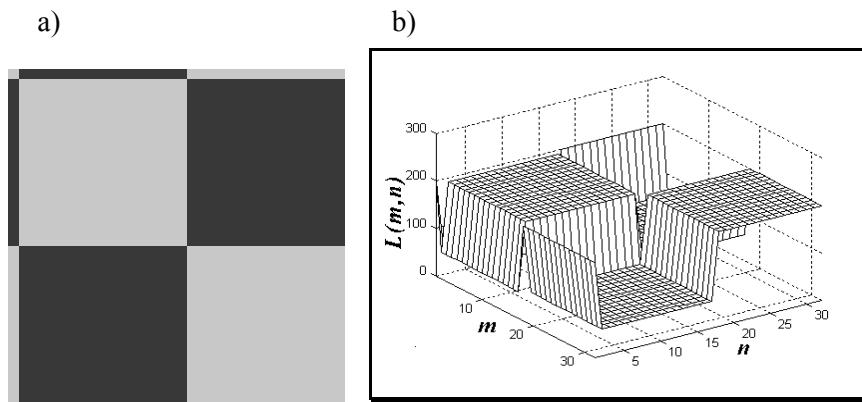
$$\left. \frac{\Delta\varphi}{\Delta i} \right|_{k=\text{const}} = -\frac{i_\varphi \cdot 2 \cdot \pi}{M} \quad (4.53)$$

gdzie  $k_\varphi$  oraz  $i_\varphi$  to dowolne liczby całkowite, dodatnie lub ujemne.

Jeżeli przyjmiemy przesunięcia w lewo i w dół jako dodatnie, to przesunięcie cykliczne w poziomie wyrażone w pikselach będzie wynosiło dokładnie tyle, ile wynosi współczynnik  $k_\varphi$ , natomiast w pionie tyle, ile wynosi współczynnik  $i_\varphi$ .



Rys. 4.52. Wykres fazy filtru jedynkowego o nachyleniach kierunkowych według wzorów (4.52) i (4.53) przy obu współczynnikach równych 1.



Rys. 4.53. Obraz otrzymany po filtracji filtrem jedynkowym o fazie z rys.4.52:  
a) w postaci poziomów szarości, b) w postaci wykresu funkcji.

Rysunek 4.52 pokazuje wykres fazy filtru o stałej, równej jeden amplitudzie, przy nachyleniach - wyznaczonych dla obu współczynników we wzorach (4.52) i (4.53) - równych 1. Obraz będący wynikiem filtracji przedstawiono na rysunku 4.53.

#### 4.3.7 Odpowiedź impulsowa filtru

Dość istotną rolę w problematyce filtracji odgrywa pojęcie odpowiedzi impulsowej filtru. Jak wskazuje nazwa, jest to wynik otrzymany po filtracji danym filtrem danych wejściowych będących impulsem jednostkowym. W przypadku obrazów cyfrowych impuls jednostkowy jest to obraz cyfrowy, posiadający wszędzie wartości zero z wyjątkiem lewego górnego narożnika, o indeksach (0,0), gdzie wartość piksela wynosi jeden - stąd nazwa „impuls jednostkowy”. Dla rozmiaru obrazu 6x6 pikseli obraz impulsu jednostkowego pokazany jest na rysunku 4.54.

1	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0

Rys. 4.54. Przykład obrazu cyfrowego, o rozmiarach 6x6 pikseli, będącego impulsem jednostkowym.

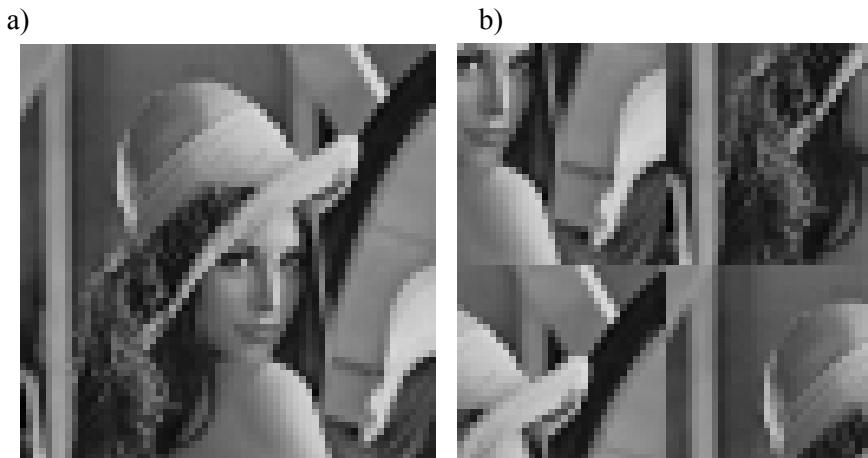
Każdy filtr zdefiniowany w dziedzinie Fouriera jest także jednoznacznie określony przez swoją odpowiedź impulsową. Wynika to choćby z tego, że odpowiedź impulsowa filtru po poddaniu jej transformacji Fouriera daje  $F$ -obraz będący definicją filtru w dziedzinie  $F$ . Łatwo skojarzyć, że odpowiedź impulsową można wobec tego otrzymać w inny sposób - podając definicję filtru w dziedzinie  $F$  odwrotnej transformacji Fouriera. Opis filtru w postaci odpowiedzi impulsowej jest przydatny do przeprowadzania filtracji bezpośrednio w dziedzinie obrazu, bez przechodzenia do dziedziny  $F$ . Taka filtracja nie polega już na mnożeniu punktowym, ale na operacji splatania obrazu filtrowanego i odpowiedzi impulsowej filtru. O splocie była już wcześniej mowa, jednak przyjrzymy się temu zagadnieniu jeszcze raz w podrozdziale 4.3.8. Zanim to nastąpi, konieczne będzie zwroć uwagę na właściwość okresowości transformacji Fouriera. Dwuwymiarowy ciąg, będący obrazem cyfrowym, jest zastępowany reprezentacją pochodzącą z próbkowanych funkcji kosinus. Wszystkie te funkcje są okresowe i jeżeli udało się dany obraz zastąpić odpowiednio skalowanymi i prze-

suniętymi próbками funkcji kosinus, to można równie dobrze przyjąć, że takie zastąpienie będzie także dokładnie pasować do dowolnej ilości okresów tworzących obraz okresowy. Na rysunku 4.55 pokazano fragment okresowego obrazu powstałego z obrazu „Lena”.

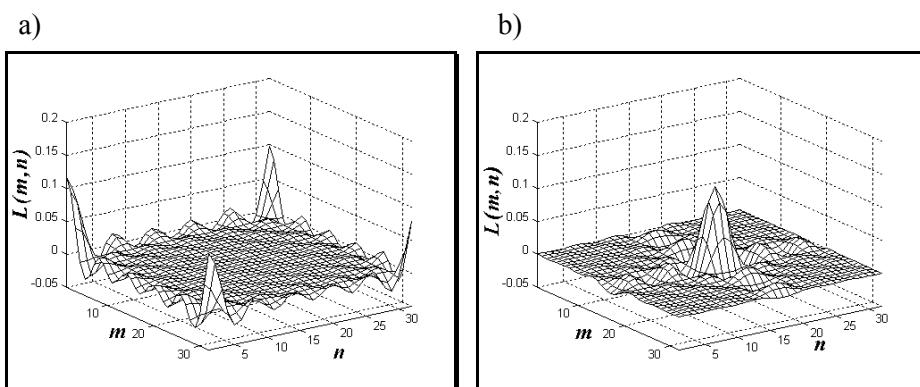


Rys. 4.55. Dwuwymiarowy obraz okresowy i dwa sposoby wyboru jednego reprezentacyjnego okresu.

Jak widać, okresowość dotyczy obu wymiarów. Na rysunku 4.55a zaznaczono sposób wybrania jednego okresu, który jednoznacznie określa cały obraz okresowy. W tym przypadku okres ten odpowiada znanemu już nam obrazowi „Lena”. W ogólności można ten okres wybrać dowolnie. Praktyczne interesujący jest jednak sposób pokazany na rys. 4.55b. Oba obrazy będące równoważnymi jedno-okresowymi reprezentacjami obrazu okresowego pokazano na rysunku 4.46. Jeżeli dowolny obraz, który poddajemy transformacji Fouriera można przedstawić w wersji okresowej, to dotyczy to także obrazu będącego odpowiedzią impulsową filtra. Okazuje się, że w przypadku odpowiedzi impulsowej przesunięcie ramki wyznaczającej jeden okres z obrazu okresowego, tak by piksel, który był w lewym górnym rogu obrazu, znalazł się jak najbliżej centrum, posiada istotne znaczenie praktyczne. Rysunek 4.57a pokazuje wykres odpowiedzi impulsowej dla filtra dolnoprzepustowego zdefiniowanego w dziedzinie  $F$  - zgodnie z rysunkiem 4.29b - bezpośrednio po odwrotnej transformacji Fouriera. Natomiast rysunek 4.57b przedstawia tę samą odpowiedź impulsową, ale po wyborze okresu podstawowego, tak by piksel będący pierwotnie w lewym górnym rogu znalazł się w centrum obrazu. Takie przesunięcie nie jest niczym nowym - jest to dokładnie to samo, co przesunięcie w dziedzinie  $F$ , stosowane dla umieszczenia składowej stałej w centrum  $F$ -obrazu, tyle, że tym razem to przesunięcie zostało uzasadnione okresowością obrazu. Analogiczna okresowość w dziedzinie  $F$  może posłużyć jako uzasadnienie przesunięcia w tej dziedzinie.



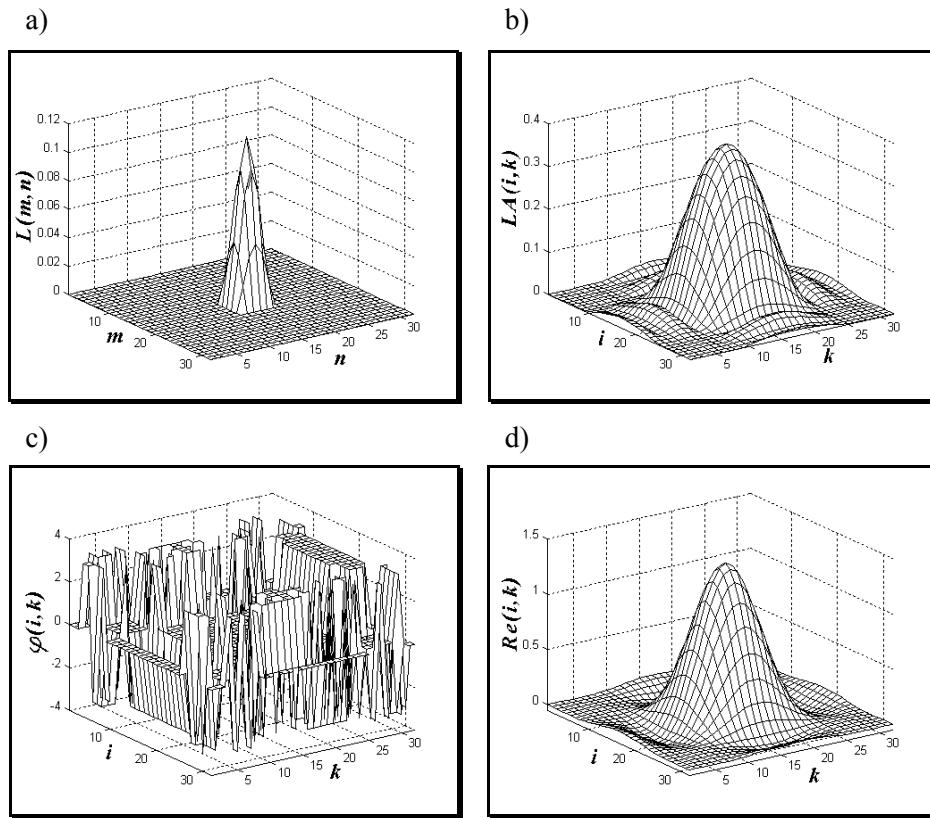
Rys. 4.56. Jeden okres obrazu okresowego: a) odpowiadający pierwotnemu obrazowi cyfrowemu, b) po przesunięciu okresu reprezentacyjnego tak, by lewy górny narożnik obrazu pierwotnego znalazł się w centrum nowego obrazu.



Rys. 4.57. Odpowiedź impulsowa filtra dolnoprzepustowego z rysunku 4.29b: a) po wyznaczeniu odwrotnej transformacji Fouriera z  $F$ -obrazu filtru, b) po przesunięciu lewego, górnego narożnika do centrum, przy wykorzystaniu okresowości obrazu.

Jak widać z wykresu na rysunku 4.57b, pewne fragmenty odpowiedzi impulsowej - skupione po przesunięciu wokół centrum - są wyraźnie większe od pozostały. Dlatego też często po prostu pomija się wartości oddalone nieco od centrum odpowiedzi impulsowej - tak jakby miały wartość zero. Prowadzi to do znacznego zmniejszenia pola odpowiedzi impulsowej, które trzeba uwzględnić przy wyznaczaniu splotu, co poważnie redukuje ilość wykonywanych obliczeń. Takie ograniczenie odpowiedzi impulsowej musi jednak wpłynąć na postać filtru w dziedzinie  $F$ . Rysunek 4.58 pokazuje, jak obcięcie pola odpowiedzi impulsowej do rozmiaru 5x5 pikseli - pozostałe wartości wyzerowano - wpłynęło na postać  $F$ -obrazu filtru. Jak widać z wykresu amplitudy, na rysunku 4.58b, filtr nie jest

już zero-jedynkowy, podział na pasmo przepustowe i zaporowe nie jest ostry, pojawiły się też pewne zafalowania. Zafalowania te, powstałe z obcięcia odpowiedzi impulsowej, znane są pod nazwą efektu Gibbsa. Rysunek 4.58c pokazuje wykres fazy. Wygląda on bardzo chaotycznie i nieczytelnie, istotne jest jednak, że wartości, jakie przyjmuje faza, pochodzą - z niewielkim błędem - wyłącznie ze zbioru  $\{-\pi, 0, +\pi\}$ . Gdyby komputer liczył dokładnie, to zbiór ten zawierałby tylko dwa elementy 0 i na przykład  $-\pi$ . Po przypomnieniu sobie sposobów zapisu liczb zespolonych - wzór (4.18) lub (4.35) - można skojarzyć, że faza plus lub minus  $\pi$  świadczy o zmianie znaku części rzeczywistej oraz o zerowej wartości urojonej. W takim przypadku o wiele czytelniejsze jest przedstawienie  $F$ -obrazu w postaci wykresu części rzeczywistej, jak na rys. 4.58d. Niekiedy może się okazać, że  $F$ -obraz posiada jedynie niezerową część urojoną - wtedy jej wykresem należy się posłużyć zamiast dwóch wykresów: amplitudy i fazy.



Rys. 4.58. Obcięcie pola odpowiedzi impulsowej z rys. 4.57b do obszaru 5x5 pikseli:  
 a) wykres odpowiedzi impulsowej po redukcji pola, b) amplituda  $F$ -obrazu filtru wynikająca z redukcji pola odpowiedzi impulsowej, c) faza  $F$ -obrazu filtru po zredukowaniu pola odpowiedzi impulsowej, d) część rzeczywista  $F$ -obrazu filtru opisanego amplitudą b) i fazą c).

### 4.3.8 Filtracja splotowa

W rozdziale 4.3.5 była mowa o filtracji obrazu w dziedzinie  $F$  przez punktowe mnożenie odpowiednich  $F$ -obrazów. Jednak obraz jest zwykle określony w dziedzinie pierwotnej i również po filtracji interesuje nas raczej jego wygląd w tej samej dziedzinie. Filtr, jak się okazało, może być także określony w dziedzinie pierwotnej przez swoją odpowiedź impulsową. W takiej sytuacji może warto przeprowadzać filtrację bezpośrednio w dziedzinie obrazu, bez stosowania transformacji w przód i odwrotnej. Operacją dokładnie równoważną filtracji w dziedzinie  $F$ , realizowanej za pomocą punktowego mnożenia, jest w dziedzinie obrazu operacja splotania, przy założeniu okresowości obrazu i tej samej okresowości odpowiedzi impulsowej filtra. Przy tym założeniu wynik operacji musi być także obrazem okresowym - o tym samym okresie. Wystarczy zatem wziąć pod uwagę jeden, odpowiednio wybrany okres. Jeżeli odpowiedź impulsowa jest mniejsza niż pojedynczy okres obrazu okresowego - czyli po prostu przetwarzany obraz cyfrowy - to nic nie stoi na przeszkodzie, by uzupełnić tę odpowiedź impulsową zerami. Taki iloczyn splotowy, który zakłada okresowość czynników, nazywa się splotem kołowym, w odróżnieniu od splotu liniowego, który wyznacza się teoretycznie w nieskończonym zakresie, a praktycznie tylko tam gdzie wynik może być różny od zera. Splot liniowy nie odpowiada dokładnie iloczynowi transformat, jednak w przypadku odpowiedzi impulsowej o niewielkim rozmiarze pola, zawierającym niezerowe wartości, różnica często nie jest istotna - splot liniowy jest niemal wszędzie równy splotowi kołowemu. Jak już napisano w rozdziale 3, wyliczanie splotu sprowadza się do wymnożenia odpowiedniej maski przez odpowiadające tej masce pole obrazu - punkt po punkcie - dodaniu wyników mnożeń i przypisaniu rezultatu pikselowi obrazu wynikowego znajdującemu się w zaznaczonym na masce splotowej położeniu. Zwykle jest to położenie centralne w masce. Maskę filtra i odpowiedź impulsową filtra można łatwo wzajemnie powiązać. Założymy, że dysponujemy odpowiedzią impulsową w postaci otrzymanej po przesunięciu punktu  $(0,0)$  do środka obrazu. Należy określić pole - najczęściej prostokątne lub kwadratowe - w którym mieszą się wszystkie niezerowe wartości tej odpowiedzi. Niekiedy wartości bliiskie zeru traktuje się jak zerowe, co, jak pokazano, daje pewne zniekształcenie filtra. Po wyznaczeniu pola niezerowych wartości odpowiedzi impulsowej następuje odwrócenie kolejności wierszy i kolumn. Należy jedynie wyróżnić piksel, który znajdował się pierwotnie w punkcie  $(0,0)$  przesuniętym do środka obrazu. Otrzymane w wyniku odwrócenia kolejności pole pikseli stanowi maskę, a wyróżniony piksel określa, gdzie należy zapisywać wynik operacji mnożenia obrazu i maski. Rysunek 5.59 pokazuje przykładową centralną (czyli nie wyzerowaną) część odpowiedzi impulsowej i odpowiadającą jej maskę. Piksel o wartości 5 został wyróżniony.

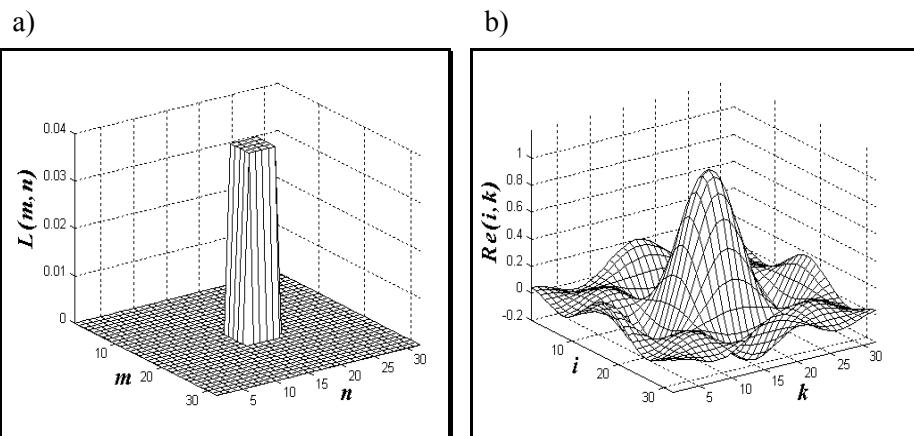
<b>1</b>	<b>-2</b>	<b>3</b>
<b>-4</b>	<b>5</b>	<b>-6</b>
<b>7</b>	<b>-8</b>	<b>9</b>

<b>9</b>	<b>-8</b>	<b>7</b>
<b>-6</b>	<b>5</b>	<b>-4</b>
<b>3</b>	<b>-2</b>	<b>1</b>

Rys. 5.59. Przykład odpowiedzi impulsowej filtru i maski do wyznaczania splotu.

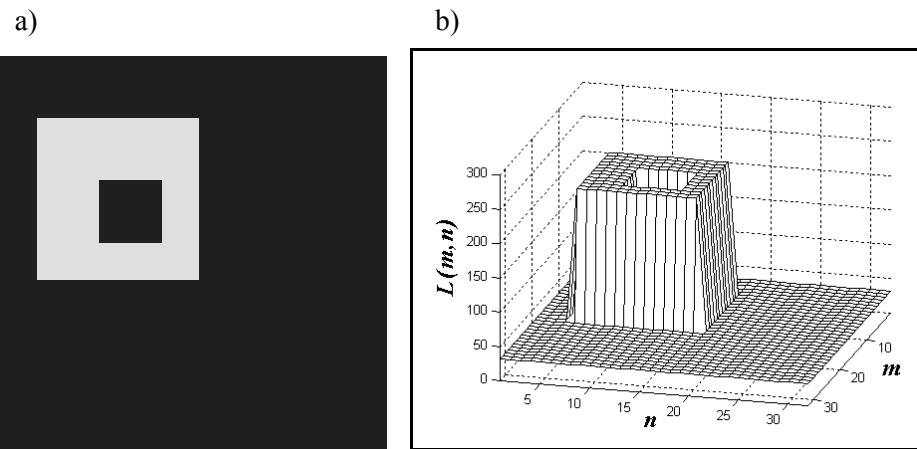
Najistotniejsze jednak jest to, że zazwyczaj filtry do przetwarzania obrazów wcale nie są definiowane w dziedzinie  $F$ . Opracowuje się maskę i stosuje bezpośrednio do wyznaczania splotu. Odpowiedź impulsową wyznacza się po to tylko, żeby z niej wyznaczyć postać częstotliwościową filtru. Postać ta może niekiedy pomóc w uzasadnieniu, dlaczego dany filtr zachowuje się w taki, a nie inny sposób, choć zwykle nie jest ani niezbędną przy opracowaniu filtru, ani przy jego stosowaniu. Rysunek 4.60 pokazuje często stosowany filtr dolnoprzepustowy.



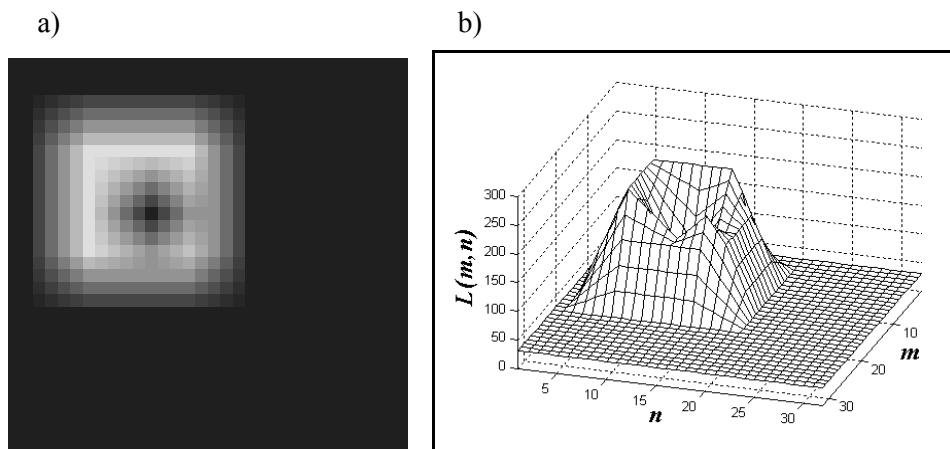
Rys. 4.60. Filtr zdefiniowany za pomocą odpowiedzi impulsowej zawierającej jedynki w centralnym polu  $5 \times 5$  pikseli, podzielone przez skalującą wartość 25:  
a) wykres odpowiedzi impulsowej, b) część rzeczywista  $F$ -obrazu filtru.

Ponieważ maska o rozmiarach  $5 \times 5$  jest symetryczna, więc jest jednocześnie odpowiedzią impulsową. Wartości maski, pierwotnie zawierającej 25 jedynek, zostały przeskalowane, tak aby ich suma była równa 1. Ten bardzo skuteczny i popularny filtr nie jest, jak widać, idealnym filtrem dolnoprzepustowym. Dla przydatności tego filtru o wiele ważniejsze jest uzasadnienie postaci maski oparte na uśrednianiu niż niedoskonała częstotliwościowa postać filtru. Bardzo istotną

zaletą tego filtra - zdefiniowanego w dziedzinie pierwotnej, bez usiłowań dopasowania go do częstotliwościowego ideału - jest brak oscylacji w obrazie będącym wynikiem filtracji. Rysunki 4.61 i 4.62 pokazują obraz oryginalny i wynik filtracji tego obrazu za pomocą filtra z rysunku 4.60.



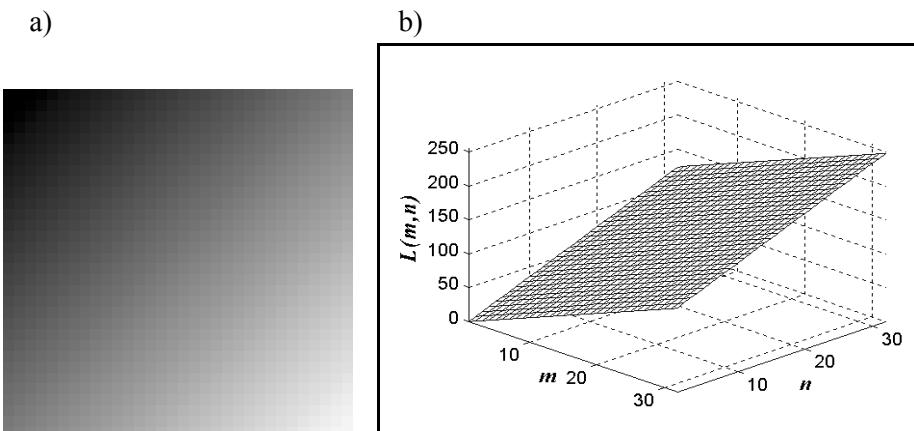
Rys. 4.61. Obraz 32x32 piksele do ilustracji filtracji dolnoprzepustowej filtrem uśredniającym: a) w postaci odcieni szarości, b) w postaci wykresu.



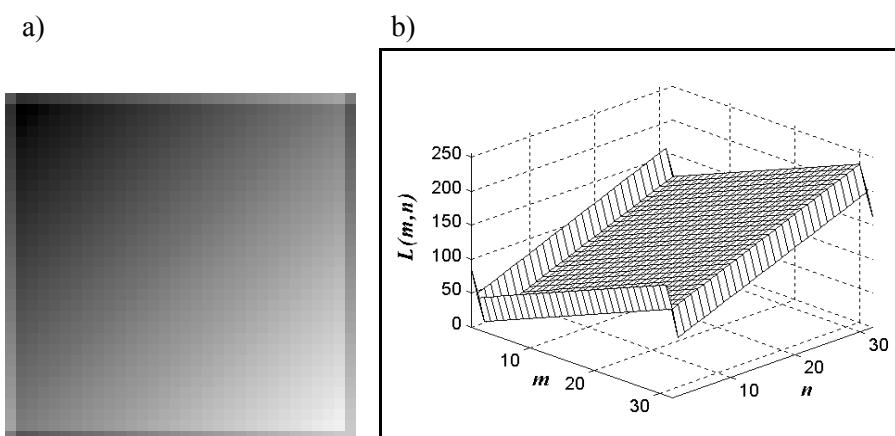
Rys. 4.62. Obraz z rysunku 4.61 po filtracji filtrem z rys. 4.60:  
a) w postaci odcieni szarości, b) w postaci wykresu.

Na zakończenie tego podrozdziału warto zwrócić uwagę na pewien szczegół, który jest tym większy, im większa jest maska filtru. Chodzi o wynik filtracji w okolicy brzegu obrazu. Jeżeli przeprowadzimy filtrację w dziedzinie częstotliwości (lub też zastosujemy równoważny splot kołowy oparty na okresowości kosinusoid), to będzie to odpowiadało sytuacji, jak gdyby maska wystająca poza ramkę oryginalnego obrazu „nasunęła” się na kolejny okres tego obrazu. Czyli na

przykład nastąpi uśrednienie elementów znajdujących się po przeciwnych stronach obrazu. Rysunki 4.63 i 4.64 pokazują taką sytuację. Nie jest to zwykle efekt pożądany. Dlatego w trakcie filtracji za pomocą maski zakłada się, że obraz poza ramką jest poszerzony o wartości pikseli znajdujące się na brzegu obrazu oryginalnego. Rysunek 4.65 przedstawia takie poszerzenie obrazu „Lena” o rozmiarach 128x128 pikseli. Rysunek pokazuje powiększenie obrazu z pewną przesadą - wystarczyłoby ono do filtracji za pomocą maski o rozmiarach 255x255. W praktyce maski są znacznie mniejsze, a ponadto nie jest wcale konieczne pamiętać obrazu tak powiększonego. Wystarczy, jeśli program realizujący filtrację daną maską wykryje wysunięcie maski poza ramkę i podstawi w miejsce brakujących pikseli wartości pikseli z brzegu obrazu.



Rys. 4.63. Obraz oryginalny do ilustracji okresowości filtracji w dziedzinie  $F$ :  
a) w postaci odcieni szarości, b) w postaci wykresu.



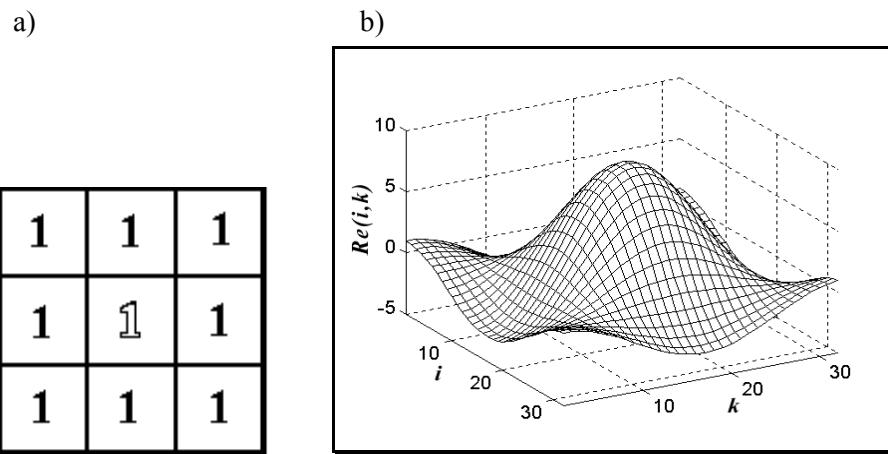
Rys. 4.64. Obraz z rys. 4.63 otrzymany po filtracji filtrem uśredniającym przeprowadzonej w dziedzinie częstotliwości (równoważnej splotowi kołowemu, czyli okresowemu): a) w postaci poziomów szarości, b) w postaci wykresu.



Rys. 4.65. Obraz „Lena” po powiększeniu pola obrazu za pomocą powielenia wartości pikseli z brzegu obrazu, w celu umożliwienia wyznaczenia wyniku filtracji za pomocą maski w pobliżu brzegu obrazu.

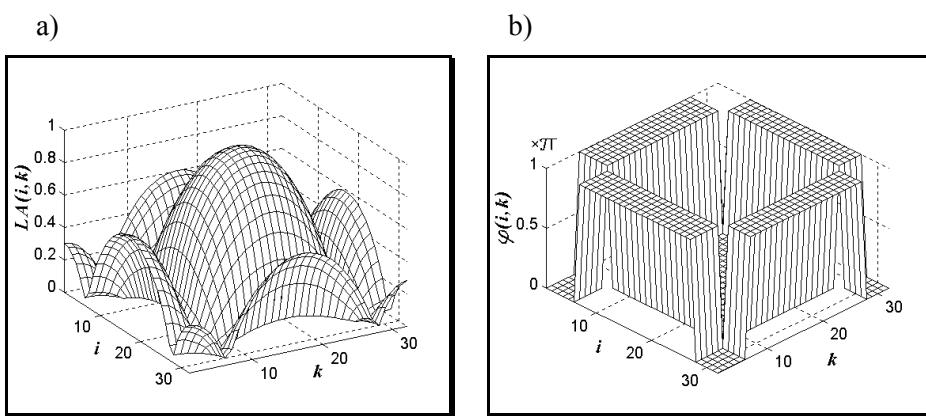
#### 4.3.9 *F*-obrazy wybranych filtrów splotowych

Jak już wspomniano, szereg filtracji realizowanych jest wprost za pomocą masek. Maski te można wyznaczyć intuicyjnie albo za pomocą zależności nie odwołujących się w ogóle do transformacji Fouriera. W kontekście dyskusji na temat zastosowania transformacji Fouriera do przetwarzania obrazów warto jednak przyjrzeć się *F*-obrazom odpowiadającym kilku przykładom spośród najczęściej stosowanych masek. Rysunek 4.66a pokazuje maskę uśredniającą, którą potocznie nazywa się filtrem dolnoprzepustowym. Jej *F*-obraz najlepiej przedstawić w postaci wykresu części rzeczywistej - rys. 4.66b - ponieważ część urojona jest wszędzie równa zero.



Rys. 4.66. Uśredniający filtr dolnoprzepustowy:  
a) maska filtru, b) część rzeczywista  $F$ -obrazu.

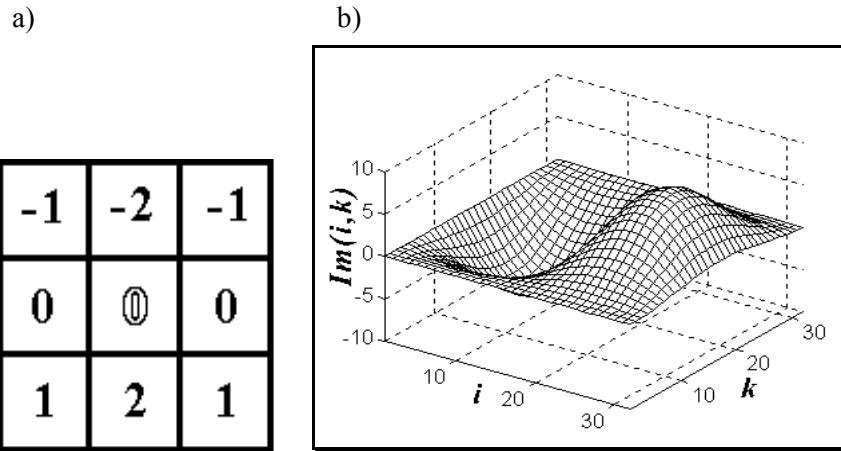
Tradycyjne wykresy amplitudy i fazy okazują się w tym przypadku nieco mniej czytelne - patrz rys. 4.67. Wykres z rys. 4.66b posiada wyraźne cechy filtru dolnoprzepustowego, jednak gdyby ktoś zaczynał od definiowania filtru w dziedzinie częstotliwości, z pewnością nie określiłby właśnie tak filtru dolnoprzepustowego.



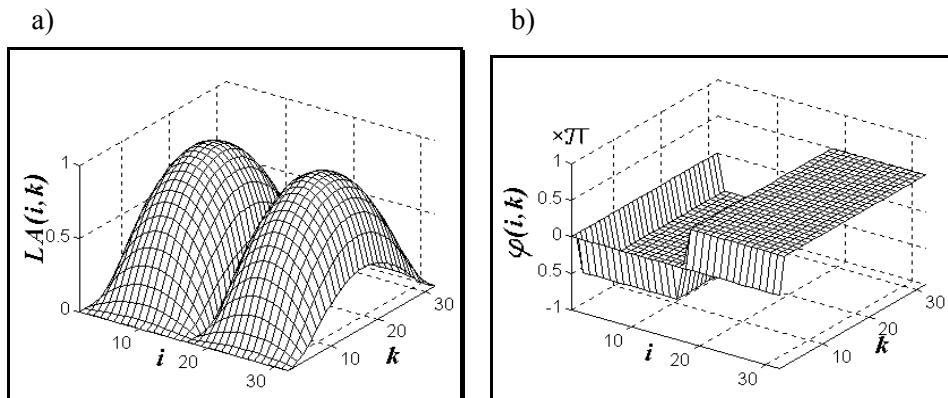
Rys. 4.67. Wykresy  $F$ -obrazu maski uśredniającej z rys. 4.66a: a) amplituda, b) faza.

Na rysunku 4.68 pokazano maskę gradientu Sobela oraz część urojoną  $F$ -obrazu tej maski. Rysunek 4.69 przedstawia wykresy amplitudy i fazy dla tej maski. Maska Sobela ma za zadanie eksponować krawędzie. Wykresy  $F$ -obrazu potwierdzają, że filtr eliminuje całkowicie składową stałą oraz wszystkie wartości  $F$ -pikseli leżących na linii przechodzącej przez  $F$ -piksel składowej stałej. Jaka jest jednak różnica pomiędzy tym  $F$ -obrazem a filtrami zero-jedynkowymi z rozdziału 4.3.5, które miały cechy preferujące wybrane kierunki? Zastosowanie

filtrów zero-jedynkowych powoduje powstanie oscylacji i jak wykazuje praktyka, są one niezbyt przydatne do zadań wymagających skutecznego eksponowania krawędzi.

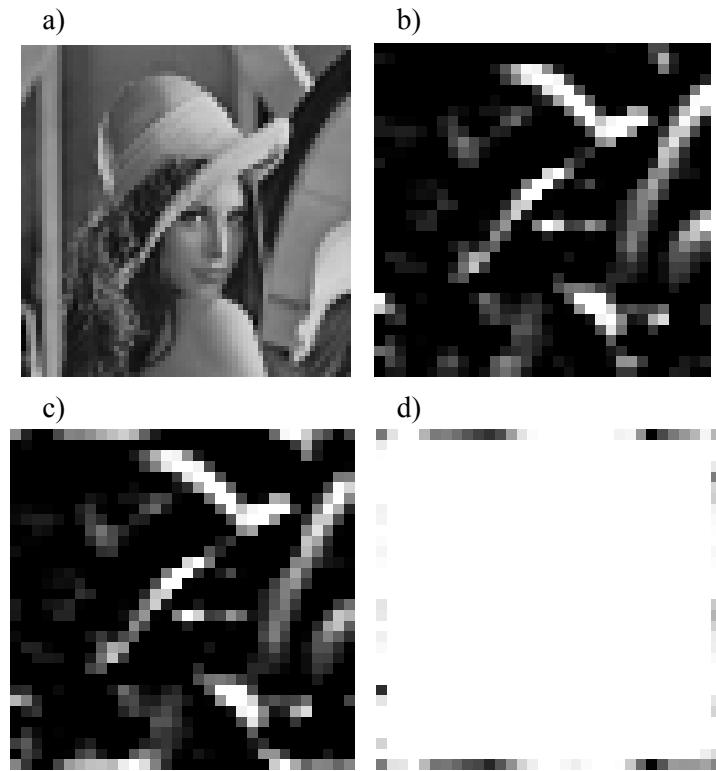


Rys. 4.68. Filtr gradientu Sobela eksponujący krawędzie poziome:  
a) maska filtru, b) część urojona F-obrazu filtru.

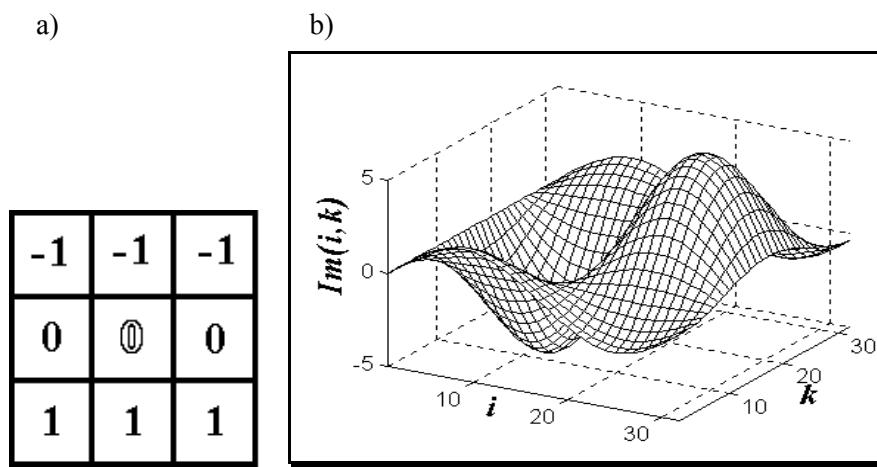


Rys. 4.69. Wykresy F-obrazu maski Sobela z rys. 4.68a: a) amplituda, b) faza.

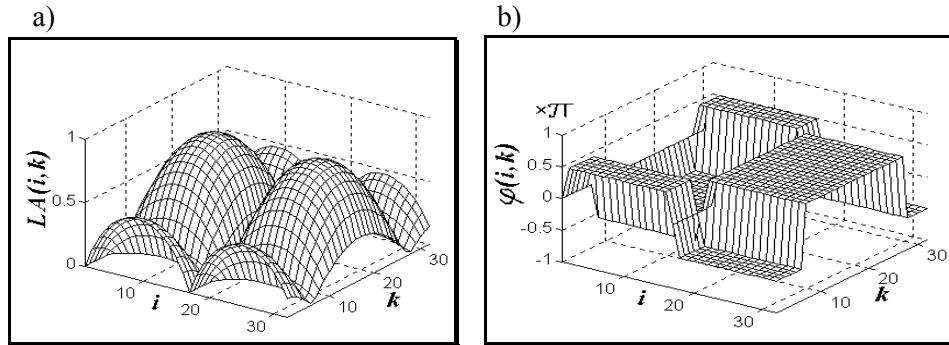
Rysunek 4.70 przedstawia jeszcze raz, tym razem na przykładzie filtru maskowego Sobela, różnicę pomiędzy splotem kołowym - zakładającym okresowość obrazu - oraz splotem z powiększeniem obrazu przez powielenie pikseli brzegowych. Na rysunku 4.70d wyraźnie widać, że różnica sięgająca na głębokość jednego piksela w głębi obrazu pojawiła się tylko wzduż linii ramki. Wynika to z rozmiaru maski 3x3. Dalsze rysunki od 4.71 do 4.77 prezentują związki innych często stosowanych masek przeznaczonych do eksponowania krawędzi obrazu z ich reprezentacją częstotliwościową. Z wykresów wynika wyraźnie, że wszystkie te filtry mają charakter górnoprzepustowy.



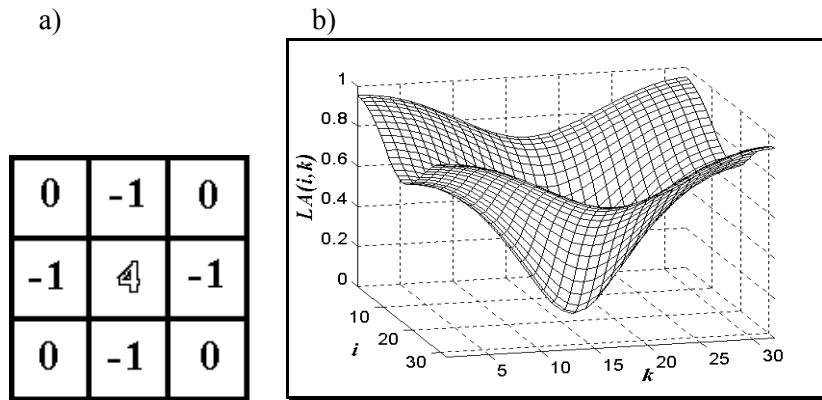
Rys. 4.70. Filtracja obrazu „Lena” o rozdzielczości  $64 \times 64$  za pomocą filtra z rys. 4.68:  
 a) obraz oryginalny, b) obraz po filtracji maską z założeniem powięlenia pikseli  
 brzegowych obrazu, c) obraz po filtracji za pomocą splotu kołowego,  
 d) negatyw modułu różnicy obrazów b) i c).



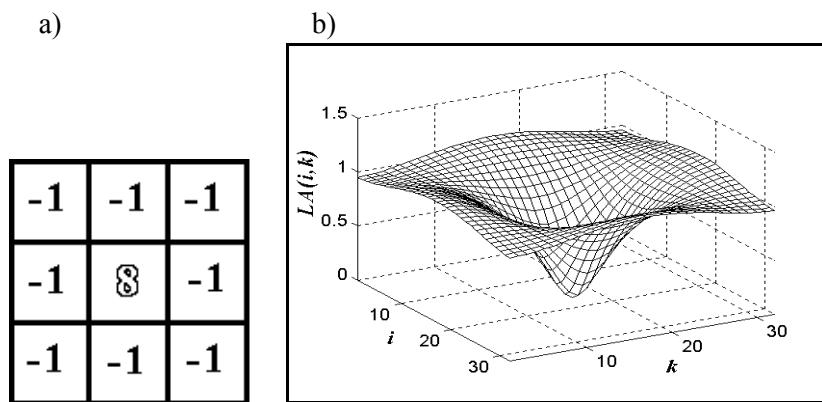
Rys. 4.71. Maska Prewitta do wyznaczania gradientu: a) zawartość maski, b) wykres  
 części urojonej opisującej  $F$ -obraz dla filtra odpowiadającego masce.



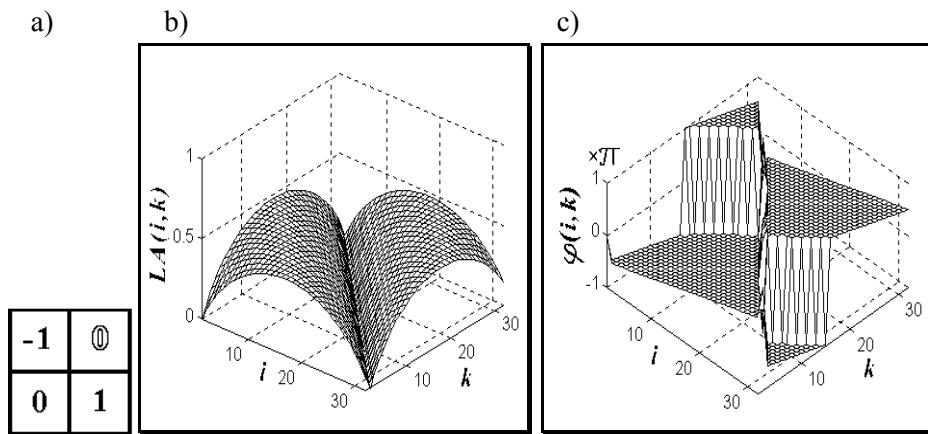
Rys. 4.72.  $F$ -obraz filtru dla maski Prewitta z rysunku 4.71a: a) wykres amplitudy,  
b) wykres fazy.



Rys. 4.73. Maska realizująca Laplasjan w oparciu o sąsiedztwo 4-punktowe i jej  $F$ -obraz: a) zawartość maski, b) wykres amplitudy opisujący w sposób kompletny  $F$ -obraz filtru odpowiadającego masce.



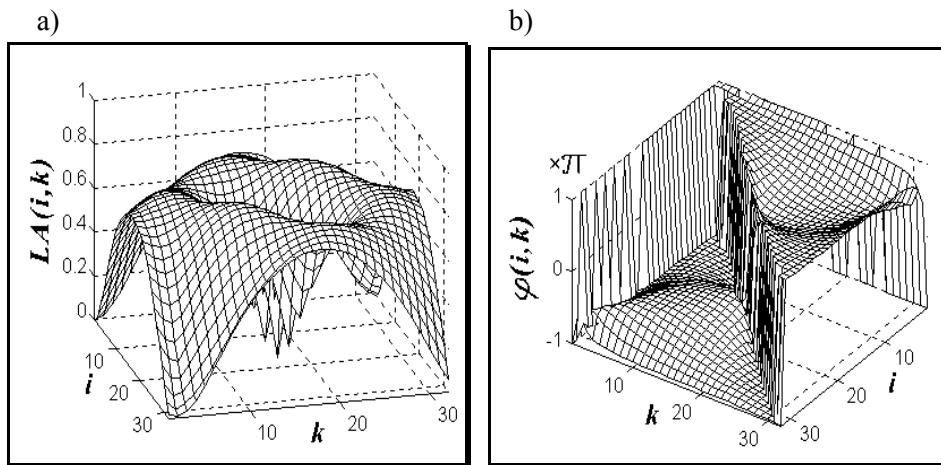
Rys. 4.74. Maska realizująca Laplasjan w oparciu o sąsiedztwo 8-punktowe i jej  $F$ -obraz: a) zawartość maski, b) wykres amplitudy opisujący w sposób kompletny  $F$ -obraz filtru odpowiadającego masce.



Rys. 4.75. Maska gradientu Robertsa i jej  $F$ -obraz: a) zawartość maski, b) wykres amplitudy  $F$ -obrazu filtru odpowiadającego masce, c) wykres fazy  $F$ -obrazu filtru odpowiadającego masce.

1	1	1
-1	-2	1
-1	-1	1

Rys. 4.76. Maska gradientu kierunkowego.



Rys. 4.77.  $F$ -obraz filtru odpowiadającego masce gradientu kierunkowego z rys. 4.76:  
a) wykres amplitudy, b) wykres fazy.

Można zauważyć, jak maski eksponujące gradienty skośne (rysunek 4.74 oraz rysunki 4.75 i 4.76) pomimo podobieństw, jeśli chodzi o przekształcanie obrazu, różnią się postaciami  $F$ -obrazów. W przypadku gradientów skośnych faza nie jest zerowa, a może nawet być nieznacznie nieliniowa - jak w przypadku filtru o  $F$ -obrazie pokazanym na rysunku 4.76. Jedne filtry posiadają  $F$ -obrazy zawierające wyłącznie niezerową część rzeczywistą, inne wyłącznie część urojoną. Maski realizujące Laplasjan - rys. 4.73 i 4.74 - są definiowane w dziedzinie  $F$  wyłącznie za pomocą amplitudy, co oznacza, że ich część rzeczywista jest wszędzie dodatnia. Jak wynika już nawet z tych niewielu przykładów, niezwykle trudno byłoby uzasadnić postać większości filtrów w dziedzinie  $F$ , gdybyśmy nie mogli odwołać się do postaci odpowiadających im masek. Dziedzina częstotliwościowa niezwykle rzadko może zatem służyć w praktyce do definiowania filtrów tego typu, czy do analizy ich przydatności do przetwarzania obrazów. Ze względu na szacunek do transformacji Fouriera warto wiedzieć, czego można się spodziewać w dziedzinie  $F$  w przypadku danej maski. Zwykle jednak skuteczność danego filtru wynika z wyglądu jego odpowiedzi impulsowej (maski) w dziedzinie pierwotnej obrazu.

#### 4.3.10 Rozplatanie, czyli filtracja odwrotna

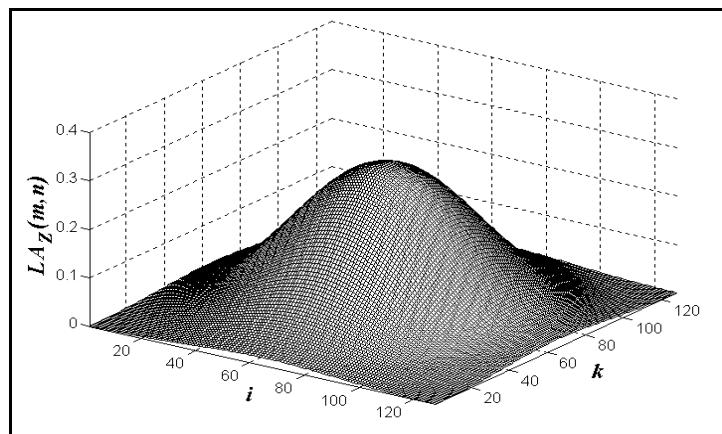
Jednym z możliwych praktycznych zastosowań filtracji opartej na interpretacji częstotliwościowej jest filtracja odwrotna, zwana także rozplataniem. Łatwo się domyślić, że chodzi o odtworzenie wyglądu obrazu, tak by jak najbardziej przypominał obraz sprzed filtracji. Zwykle filtracja, do której poszukuje się odwrotnej, nie jest realizowana celowo, ale jest niepożądany efektem ubocznym jakiegoś etapu procesu rejestracji obrazu. Jeżeli ten szkodliwy efekt da się opisać jako filtrację, to problem wydaje się być stosunkowo prosty. Wystarczyłoby znieksztalcony obraz poddać filtracji filtrem, którego każdy  $F$ -piksel stanowić będzie odwrotność  $F$ -piksela filtru modelującego znieksztalcone obrazu. Nie jest to jednak takie proste. Nie każde znieksztalcone obrazu można opisać jako filtrację. Operacja, którą da się opisać  $F$ -obrazem filtru, musi być po pierwsze liniowa, po drugie musi być niezmienna względem przesunięcia. O niezmienności względem przesunięcia i liniowości była już w tej książce mowa. Dla przypomnienia powtórzmy, że pierwsza cecha oznacza, że dany obiekt widoczny na obrazie powinien być jednakowo znieksztalcony, niezależnie od tego, w jakim położeniu się znalazł względem ramki obrazu. Jak widać, nie spełnia tego wymagania wiele znieksztalceń geometrycznych pochodzących od układów optycznych. Liniowość dowolnej operacji wykonanej na obrazach - oznaczmy tę operację jako  $O[\bullet]$  - można dla dowolnych dwóch obrazów cyfrowych  $L_1$  i  $L_2$  oraz dowolnych liczb  $a_1$  i  $a_2$  opisać jako następującą równość:

$$O[a_1 \cdot L_1 + a_2 \cdot L_2] = a_1 \cdot O[L_1] + a_2 \cdot O[L_2] \quad (4.54)$$

Oznacza to, że ostateczny wynik będzie ten sam, niezależnie od tego, czy dodamy obrazy przed operacją „O”, czy też dodamy wyniki po operacji wykonanej na tych obrazach odziennie. Wynik nie będzie także zależał od tego, czy pomnożymy wszystkie piksele przez jakąś stałą przed operacją, czy też pomnożymy przez tę stałą dopiero wynik operacji. Zatem jeżeli filtr liniowy ma modelować zniekształcenia, to powinny one mieć identyczny charakter niezależnie od tego, czy obraz jest ciemny, czy jasny.

Zależność (4.55) przedstawia sposób wyznaczania filtra odwrotnego - przyjęto, że  $F_Z$  oznacza  $F$ -obraz filtru zniekształcającego, a  $F_O$  filtra odtwarzającego:

$$F_O(i, k) = \frac{1}{F_Z(i, k)} \quad (4.55)$$



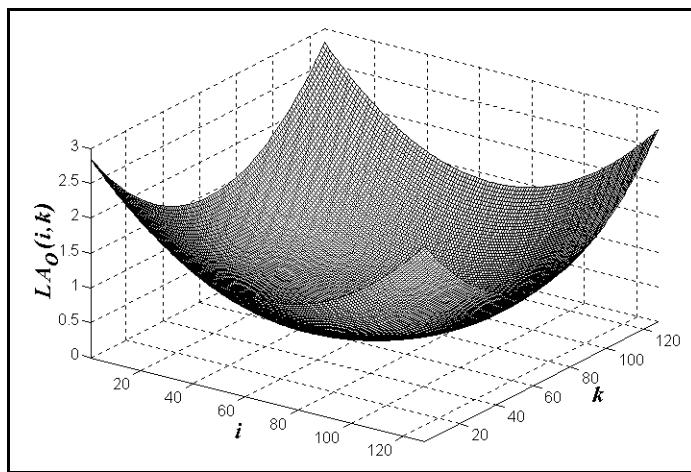
Rys. 4.78. Amplituda filtru modelującego zniekształcenia.



Rys. 4.79. Obraz oryginalny - a) oraz zniekształcony filtrem z rys 4.78 - b).

Jak wynika z (4.55), jeżeli znieksztalcenie obrazu uda się zamodelować w postaci filtracji liniowej niezmiennej względem przesunięcia, to, aby możliwe było wyznaczenie odwrotności  $F$ -pikseli, filtr modelujący znieksztalcenia powinien mieć wszystkie  $F$ -piksele różne od zera. Zatem obraz znieksztalcony filtrem zero-jedynkowym nie będzie mógł być wiernie odtworzony - po filtracji takim filtrem część informacji jest bezpowrotnie tracona. Rys. 4.78 pokazuje amplitudę filtra dolnoprzepustowego o zerowej fazie, a rys. 4.79 znieksztalcony przez filtrację tym filtrem obraz „Lena”.

Na rysunku 4.80 przedstawiono amplitudę filtra odwrotnego, wyznaczonego według zależności (4.55). Należy zwrócić uwagę, że w tym przypadku znany jest dokładnie filtr znieksztalcający, co w praktyce rzadko się zdarza. Filtr znieksztalcający może zazwyczaj jedynie w przybliżeniu oddawać sposób powstawania znieksztalczeń.



Rys. 4.80. Amplituda odtwarzającego filtra odwrotnego do filtra znieksztalcającego z rys. 4.78.

Naturalnie przeprowadzenie filtracji znieksztalcającej i odwrotnej, przy założeniu nieskończonej precyzji zapisu wartości pikseli obrazu znieksztalconego, musi prowadzić do idealnego odtworzenia obrazu sprzed znieksztalcenia. Jednak symulując sytuację bardziej zbliżoną do rzeczywistości, wartości pikseli obrazu znieksztalconego zostały zapisane z 8-bitową precyzją - taki obraz, o skończonej dokładności, dostaje się przecież zwykle do przetwarzania. Okazuje się, że kwantyzacja wartości prowadzi do pewnych znieksztalczeń w odtworzonym obrazie - patrz rysunek 4.81.

Zatem filtracja odwrotna za pomocą odwrotnego filtra, wyznaczonego w dziedzinie  $F$  według zależności (4.55), kryje w sobie szereg pułapek. Jeżeli jednak umiemy zamodelować filtr znieksztalcający, to możemy odtwarzać obraz znieksztalcony inaczej, bez wyznaczania filtra odwrotnego - metodą iteracyjną.



Rys. 4.81. Obraz odtworzony z obrazu z rys. 4.79b przez filtr odwrotny pokazany na rys. 4.80.

Oznaczmy przez  $L_p$  obraz odtworzony po  $p$ -tej iteracji, natomiast przez  $\Psi[L, L_Z]$  filtrację obrazu  $L$  za pomocą filtra znieksztalcającego  $L_Z$ . Zarówno obraz jak i filtr mogą być również określone w dziedzinie  $F$ . Wynik filtracji  $\Psi$  niech zawsze będzie obrazem w dziedzinie pierwotnej. Jak widać, powyższa notacja nie określa, w jakiej dziedzinie zdefiniowany jest filtr, a w jakiej obraz i czy filtrację realizuje się przez punktowe mnożenie  $F$ -obrazów, czy przez splot obrazu z odpowiednią impulsową filtrem - w tym przypadku nie jest to po prostu istotne:

$$\Psi[L, L_Z] = \Psi[L, F_Z] = \Psi[F, L_Z] = \Psi[F, F_Z] \quad (4.56)$$

Jeżeli na początku podstawimy za  $L_1$  obraz znieksztalcony, który ma być poddany przetworzeniu, to algorytm, rozpoczynający iteracje od wartości  $p=1$ , można opisać następującym równaniem rekurencyjnym:

$$L_{p+1} = L_p + (L_1 - \Psi[L_p, L_Z]) \quad (4.57)$$

Podejście to ma wadę polegającą na konieczności wielokrotnego (zwykle kilkukrotnego) powtórzenia filtracji obrazu, jednak parę istotnych zalet wadę tę może w poważnym stopniu zrekompensować. Metoda ta dopuszcza określenie filtra znieksztalcającego tylko w dziedzinie obrazu - zatem można wybrać odpowiedź impulsową o stosunkowo niewielkim rozmiarze maski. W rezultacie do realizacji kolejnych kroków algorytmu można stosować filtrację splotową, bez przechodzenia do dziedziny Fouriera. Ponadto dzięki temu, że nie jest konieczne korzystanie z filtra odwrotnego, nie powstaje problem wyznaczania odwrotności

dla wartości zerowych lub bliskich zera. Właśnie te wartości bliskie zera w  $F$ -obrazie filtra znieksztalcającego powodują zazwyczaj pojawienie się niespodziewanych efektów ubocznych, jak na rys. 4.81. Na rysunku 4.82 pokazano wynik rozplatania iteracyjnego według zależności (4.57) dla obrazu znieksztalconego filtrem z rys. 4.78. Już szósta iteracja dała wynik względnie zadowalający. Zbytne przedłużanie realizacji algorytmu prowadzi do powstania zaszumienia obrazu - co w przypadku prezentowanego przykładu uwidocznioło się dość wyraźnie dopiero po ponad 20 iteracjach. Dla innych obrazów i filtrów znieksztalcających szum może ujawnić się już w kilku iteracjach po osiągnięciu zadowalającego obrazu odtworzonego. Należy podkreślić, że rozplatanie metodą iteracyjną jest znacznie mniej ryzykowne od filtracji filtrem odwrotnym, a ponadto może być realizowane bez jakiegokolwiek odwoływanego się do transformacji Fouriera.



Rys. 4.82. Obraz odtworzony po kolejnych iteracjach metody rekurencyjnej według wzoru (4.57): a) po pierwszej iteracji, b) po 6 iteracjach, c) po 24 iteracjach, d) po 40 iteracjach.

### 4.3.11 Podsumowanie

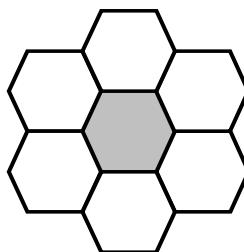
Pomimo tego, że rozdział ten jest dość obszerny, można go traktować jedynie jako wstęp do szeregu dalszych zagadnień związanych z zastosowaniem transformacji Fouriera do przetwarzania obrazów. Jednym z głównych celów przedstawionego opracowania było wyjaśnienie podstawowych cech transformacji Fouriera w odniesieniu do obrazów i pokazanie, że mimo poczesnego miejsca transformacji Fouriera wśród technik przetwarzania sygnałów ciągłych i cyfrowych, zastosowanie jej do przetwarzania obrazów wymaga poważnego zastanowienia. Bardzo łatwo jest bowiem po prostu poszerzyć odpowiednie wzory o jeden wymiar i odnieść wrażenie, że opracowało się nową technikę przetwarzania obrazów. Należy dokładnie zdawać sobie sprawę ze słabych stron takiego podejścia i ewentualnych pułapek, by po włożeniu sporego wysiłku nie okazało się, że opracowano „kwadratowe koło”. Dobre zrozumienie istotnych cech dyskretniej dwuwymiarowej transformacji Fouriera może jednak zaowocować w postaci bardzo przydatnych metod - transformacja Fouriera może być wykorzystana w zaawansowanych technikach przetwarzania obrazów o charakterze specjalistycznym, na przykład obrazów medycznych, satelitarnych lub astronomicznych, do analizy tekstur czy do wyznaczania wymiaru fraktalnego. Dość znaczną analogię do transformacji Fouriera wykazuje szereg innych transformacji, powszechnie stosowanych na przykład do kompresji obrazów, takich jak transformacja kosinusowa czy falkowa (*ang. Discrete Wavelet Transform*).

## 5 Przekształcenia morfologiczne

### 5.1 Uwagi wstępne

Przekształcenia morfologiczne są jednymi z najważniejszych operacji w komputerowej analizie obrazu, gdyż - odpowiednio kombinowane w zestawy - pozwalają na najbardziej złożone operacje, związane z analizą kształtu elementów obrazu, ich wzajemnego położenia oraz umożliwiają złożone procesy symulacji. Podstawowe przekształcenia morfologiczne są bowiem punktem wyjściowym do tworzenia bardziej złożonych operacji, związanych z analizą kształtu obiektów, oraz ich wzajemnego rozmieszczenia. Niestety największą ich wadą jest wielka złożoność obliczeniowa, na skutek której rozpowszechniły się one w analizatorach obrazu dopiero w drugiej połowie lat 80-tych<sup>25</sup>.

Fundamentalnym pojęciem przekształceń morfologicznych jest tzw. *element strukturalny obrazu*<sup>26</sup>. Jest to pewien wycinek obrazu (przy dyskretnej reprezentacji obrazu - pewien podzbiór jego elementów) z wyróżnionym jednym punktem (tzw. punktem centralnym). Najczęściej stosowanym elementem strukturalnym jest koło o promieniu jednostkowym. Przy wyborze rzeczywistego elementu strukturalnego nie bez znaczenia jest, dyskutowany wcześniej w rozdziale 2, wybór siatki elementów obrazu. Różnicę tę obrazują rysunki 5.1 i 5.2, przedstawiające ten sam element strukturalny przy zastosowaniu najpopularniejszych siatek:



Rys. 5.1 Element strukturalny (koło o promieniu jednostkowym) na siatce heksagonalnej.

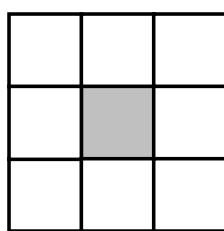
W ogólnym przypadku siatka heksagonalna jest lepsza, gdyż zapewnia następujące korzyści:

---

<sup>25</sup> Pierwszym analizatorem obrazu wykorzystującym operacje morfologiczne był TAS (*Texture Analysis System*), stworzony w latach 70-tych przez firmę Leitz przy współpracy z Centrum Morfologii Matematycznej w Fontainbleau.

<sup>26</sup> ang. *structuring element*

- podstawowy element strukturalny jest bardziej zbliżony do koła, analizowanego teoretycznie w większości przekształceń;
- podstawowy element strukturalny składa się z siedmiu elementów, w porównaniu do dziewięciu elementów przy siatce kwadratowej, co zmniejsza złożoność obliczeniową;
- wynik przekształcenia morfologicznego opartego o siatkę heksagonalną jest bardziej zgodny z intuicją, gdyż nie preferuje ono wyraźnie dwóch “abstrakcyjnych” kierunków: poziomego i pionowego.



Rys. 5.2. Element strukturalny (koło o promieniu jednostkowym) na siatce kwadratowej.

Pomimo tych zalet najczęściej stosuje się siatkę kwadratową (jak wspomniano już w rozdziale 2), gdyż jest ona tradycyjnie przyjmowana jako podstawowa i znacznie prostsza w realizacji. Zgodnie z tym faktem w dalszej części tego rozdziału, w szczególności we wszystkich przykładach, przyjęta zostanie siatka kwadratowa.

## 5.2 Ogólny algorytm przekształcenia morfologicznego

Intuicyjnie można powiedzieć, że każde przekształcenie morfologiczne polega na:

- przyłożeniu centralnego punktu kolejno do wszystkich punktów obrazu;
- sprawdzeniu, czy lokalna konfiguracja punktów odpowiada układowi, zapisanemu w elemencie strukturalnym;
- wykonaniu, w przypadku zgodności konfiguracji punktów, operacji określonej dla danego przekształcenia.

W sposób bardziej ogólny i formalnie poprawniejszy przekształcenie morfologiczne zapisać można następującym algorytmem:

1. Element strukturalny jest przemieszczany po całym obrazie i dla każdego punktu obrazu wykonywana analiza koincydencji punktów obrazu i elementu strukturalnego, przy założeniu, że badany punkt obrazu jest punktem centralnym elementu strukturalnego;

2. W każdym punkcie obrazu następuje sprawdzenie, czy rzeczywista konfiguracja pikseli obrazu w otoczeniu tego punktu zgodna jest z wzorcowym elementem strukturalnym.
3. W przypadku wykrycia zgodności wzorca pikseli obrazu i szablonu elementu strukturalnego - następuje wykonanie pewnej (ustalonej) operacji na badanym punkcie. Zazwyczaj jest to prosta zmiana koloru lub nasycenia jasności tego punktu, chociaż w ogólnym przypadku można założyć wykonanie dowolnej operacji.

Warto odnotować pewną dość istotną cechę przekształceń morfologicznych, odróżniającą je od wszystkich innych, wcześniej omawianych przekształceń i filtracji obrazów. Otóż przekształcenia punktowe transformują każdy punkt obrazu w taki sam sposób, bez względu na to, jakich ma on sąsiadów. Filtry konwolucyjne, medianowe, logiczne i inne - uzależniają wynik od sąsiedztwa danego punktu, ale przekształcenie jest wykonywane zawsze, nawet jeżeli wartość obrazu w danym punkcie nie ulegnie zmianie. Przekształcenia morfologiczne natomiast przekształcają tylko tę część punktów obrazu, których otoczenie jest zgodne z elementem strukturalnym, co pozwala na szczególnie subtelne planowanie przekształceń.

## 5.3 Typowe przekształcenia morfologiczne

### 5.3.1 Erozja

Erozja jest podstawowym przekształceniem morfologicznym. Aby zdefiniować operację erozji zakłada się, że istnieje nieregularny obszar X i koło B o promieniu  $r$ , które będzie elementem strukturalnym. Jako punkt środkowy elementu strukturalnego przyjmuje się środek koła B. Wówczas erozję figury X elementem B można zdefiniować na dwa różne sposoby:

- figura zerodowana to zbiór środków wszystkich kół o promieniu  $r$ , które w całości zawarte są we wnętrzu obszaru X;
- koło B przetacza się po **wewnętrznej** stronie brzegu figury. Kolejne położenia środka koła B wyznaczają brzeg figury zerodowanej.

W implementacji komputerowej erozja jednostkowa polega na usunięciu wszystkich tych punktów obrazu o wartości 1, które posiadają choć jednego sąsiada o wartości 0. Erozję można także interpretować matematycznie jako tzw. filtr minimalny, to znaczy taki operator, w którym każdemu punktowi przypisuje się **minimum** z wartością jego sąsiadów.

Dla zachowania zgodności z opisami innych, bardziej złożonych operacji morfologicznych, warto tu wskazać, że operację erozji można wykonać posługując się elementem strukturalnym przedstawionym na rys. 5.3.

1	1	1
1	1	1
1	1	1

Rys. 5.3. Element strukturalny.

Ponieważ rysunków podobnych do rysunku 5.3., definiujących różne elementy strukturalne, będziemy dalej rozważać dosyć dużo - celowe jest wprowadzenie tutaj pewnej konwencji notacyjnej, którą będziemy stale dalej stosować. I tak na rysunkach wyobrażających elementy strukturalne poszczególne symbole będą miały następujące znaczenia:

- 0** - piksel o szarości mniejszej od tła („zgaszony”),
- 1** - piksel o szarości większej od tła („zapalony”),
- x** - piksel o dowolnej szarości (zapalony lub zgaszony - jest to bez znaczenia).

Jeżeli punkt centralny i otoczenie elementu strukturalnego (rys. 5.3) zgadza się z lokalną konfiguracją punktów obrazu to nowa wartość punktu centralnego obrazu przyjmuje wartość 1. W przeciwnym przypadku punkt centralny przyjmuje wartość 0.

Najczęściej erozję przeprowadza się za pomocą elementu strukturalnego, w postaci punktu centralnego i jego otoczenia lepiej lub gorzej naśladującego koło, ale można też używać innych elementów strukturalnych, np. pary punktów, odcinków, trójkątów, itp.

Elementy liniowe jako elementy strukturalne są przydatne do wyszukiwania fragmentów obrazu zorientowanych liniowo w tym samym kierunku, co element strukturalny. Może to mieć różne ciekawe zastosowania. Na przykład, jeżeli potrzebujemy metodami komputerowej analizy obrazu sprawdzić, czy w badanym przez nas obrazie przedstawiającym materiał włóknisty (na przykład próbki tekstylne) nie ma zakłóceń przebiegu włókien, które powinny być proste i równoległe, to erozja elementem strukturalnym liniowym równoległym do założonego kierunku przebiegu włókien usunie z obrazu wszystkie te fragmenty, które są odchylone od wybranego kierunku. Różnica obrazu wyjściowego i zerodowanego pozwoli wykryć poszukiwane miejsce zakłóceń.

Erozja jako przekształcenie obrazu jest wykonywana jako samodzielne (autonomiczne) przekształcenie obrazu albo pełni rolę pomocniczą (jest punktem odniesienia lub elementem składowym dla innych przekształceń). Ponieważ erozja jest operacją prostą i stosunkowo łatwą do wyobrażenia - na jej podstawie omówione zostaną cechy wspólne także dla innych przekształceń morfologicznych. Najpierw spróbujemy zaproponować pewną formalną definicję erozji, która będzie nam potem przydatna.

### **Definicja I**

Zakładamy, że obraz wyjściowy zawiera pewien obszar (figure) X, wyróżniający się pewną charakterystyczną cechą (np. odróżniającą się od tła jasnością). Figura X po wykonaniu operacji erozji (często określana krótko jako figura zerodowana) to **zbiór punktów centralnych wszystkich elementów strukturalnych, które w całości mieszą się we wnętrzu obszaru X**. Miarą stopnia erozji jest wielkość elementu strukturalnego. Im większy rozmiar elementu strukturalnego - tym większa część brzegu podlegającej erozji figury zostaje usunięta.

### **Definicja II**

Erozję można traktować jako filtr minimalny. Dzięki temu pojęcie erozji można rozszerzyć na obrazy posiadające wiele stopni szarości, a nawet kolorowe. W przypadku obrazu posiadającego wiele poziomów jasności operację erozji wygodnie zapisać jest jako:

$$L'(m, n) = \min_{m_i, n_i \in B(m, n)} (L(m_i, n_i))$$

gdzie:  $L(m, n)$  - jasność punktu o współrzędnych  $(m, n)$ ;

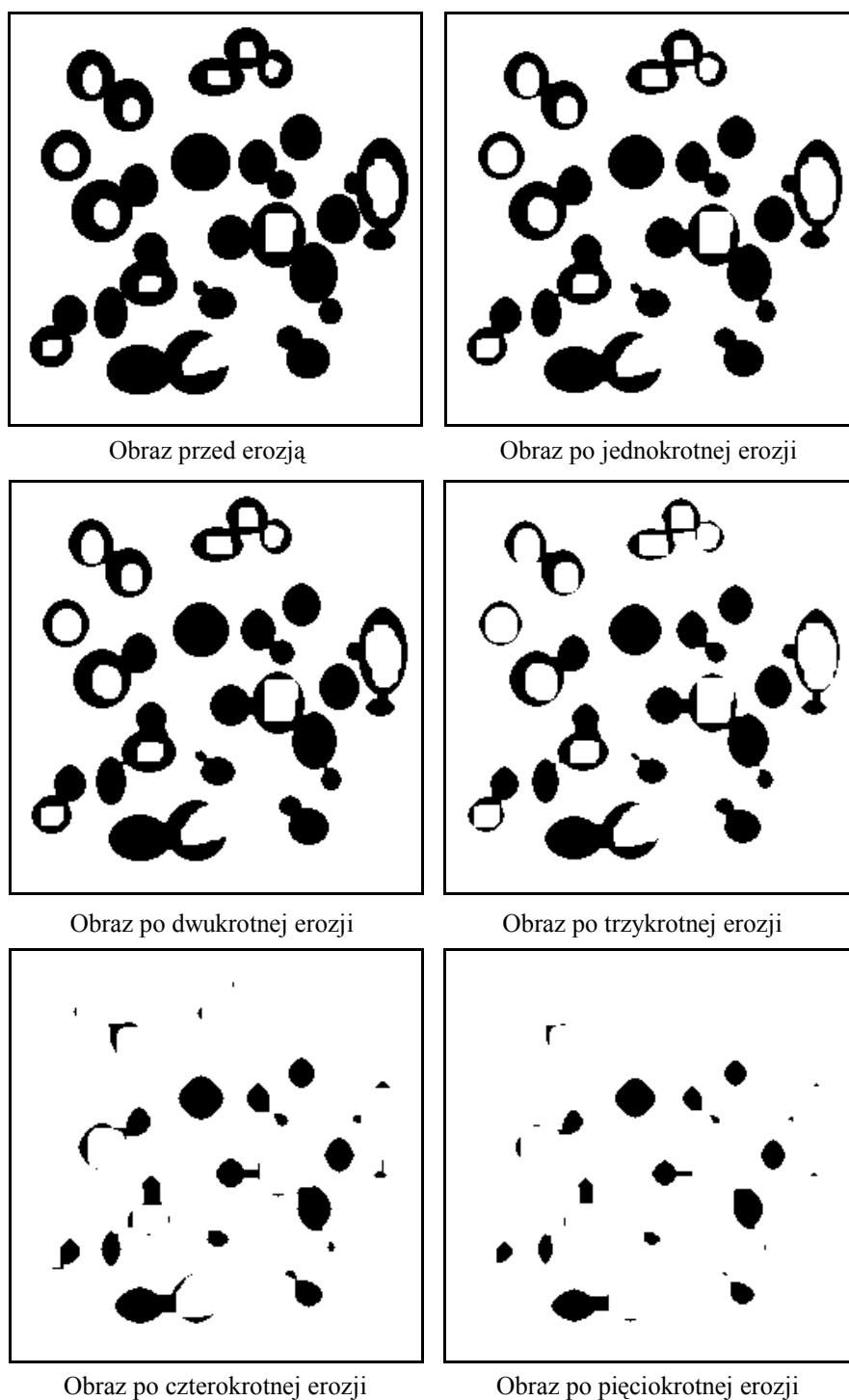
$B(m, n)$  - element strukturalny z punktem centralnym o współrzędnych  $(m, n)$ .

Erozję obrazu kolorowego można zdefiniować jako złożenie niezależnych erozji przeprowadzonych na jego poszczególnych składowych (RGB<sup>27</sup> lub Hue-Sat-Lum<sup>28</sup>). Wynik erozji obrazu kolorowego może jednak znacznie odbiegać od intuicyjnie spodziewanego rezultatu!

Przebieg procesu erozji dla prostego obrazu ziaren prześledzić można na rysunku 5.4. Kolejne kadry na tym obrazie pokazują kolejne iteracje procesu erozji przy założeniu, że element strukturalny ma rozmiary  $3 \times 3$  piksele lub wyniki erozji z coraz większym elementem strukturalnym. Na rysunku 5.4. pokazano w istocie negatywy odpowiednich obrazów, ponieważ oryginalne obrazy (przedstawiające białe plamy odpowiadające obszarom z sygnałem „1” na czarnym tle utożsamianym z wartością „0”) są mniej wygodne do oglądania i do interpretacji. Nie należy się jednak dziwić, oglądając potem kolejne obrazy poddawane erozji, że na pokazanym niżej rysunku erozji towarzyszy zmniejszanie się ciemnych obszarów, natomiast na obrazach rzeczywistych obraz w trakcie erozji będzie ciemniał. Jest to wynik wspomnianego wyżej, użytego wyłącznie do prezentacji efektów erozji na sztucznym obrazie, zabiegu polegającego na demonstrowaniu negatywu, a nie samego obrazu.

<sup>27</sup> RGB - ang. (*red, green, blue*) - rozkład koloru na składowe: (czerwony, zielony, niebieski).

<sup>28</sup> HSL - ang. (*hue, saturation, luminosity*) - rozkład koloru na składowe: (czerwono-niebieski, nasycenie, jasność).



Rys. 5.4. Przebieg erozji prostego obrazu (pokazany w negatywie).

Erozja posiada następujące cechy, mające duże znaczenie praktyczne:

- Jest addytywna, co oznacza, że erozję o założonej wielkości można interpretować jako złożenie odpowiedniej ilości erozji o wielkości jednostkowej<sup>29</sup>. Cecha ta ma wielkie znaczenie przy praktycznej realizacji przekształcania.
- Erozja złożonym elementem strukturalnym jest równoważna złożeniu erozji poszczególnymi elementami tego elementu strukturalnego.
- Położenie punktu centralnego elementu strukturalnego nie ma większego znaczenia. Zmiana położenia punktu centralnego o dany wektor powoduje przesunięcie (translację) obrazu wynikowego o taki sam wektor.
- Erozja ma zdolność do eliminacji drobnych szczegółów i wyglądzania brzegu figury.
- Erozja elementami strukturalnymi o podłużnym kształcie pozwala uwypuklić fragmenty obrazu zorientowane liniowo w tym samym kierunku, co element strukturalny.
- Erozja dokonuje generalizacji obrazu. Odizolowane, drobne wyróżnione obszary zostają usunięte. Brzegi wyróżnionych obszarów zostają wygładzone, ich długość zostaje zdecydowanie zmniejszona. Zmniejszone zostają także ich powierzchnie. Często większe wyróżnione obszary podzielone zostają na mniejsze<sup>30</sup>. Spada “nasycenie” obrazu.

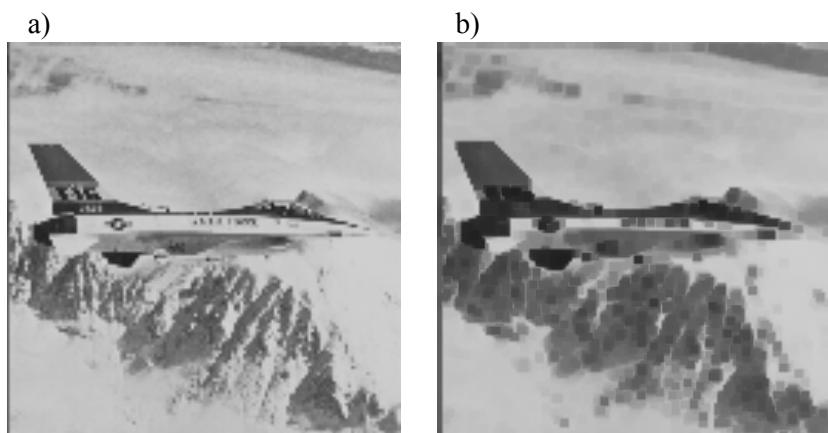
Efekty erozji pokazane wyżej na sztucznych i dość wyidealizowanych obrazach ziaren nie ujawniły wszystkich aspektów tego procesu, dlatego niżej pokazano skutek działania operatora erozji dla obrazów rzeczywistych. Skorzystano przy tym ze znanych już z wcześniejszych rozdziałów obrazów „Lena” i „Jet”. Na rysunku 5.5 pokazano obraz „Lena” przed i po erozji, najpierw z elementem strukturalnym o rozmiarach  $3 \times 3$  piksele, a potem z elementem strukturalnym o rozmiarze  $5 \times 5$  pikseli. Na rysunku 5.6 pokazano w analogiczny sposób产生的 obraz „Jet” (tylko obraz oryginalny i erozja  $5 \times 5$ ).

<sup>29</sup> Stwierdzenie to nie jest do końca prawdziwe. Przy bardzo małych elementach strukturalnych i warunkach wynikających z przyjętej siatki ich kształt znacznie odbiega od teoretycznego elementu strukturalnego (np. koła). W wyniku tego niekorzystnego zjawiska złożenie wielu erozji jednostkowych może odbiegać od spodziewanego wyniku.

<sup>30</sup> Zjawisko to jest wykorzystane przy rozdzielaniu fragmentów obrazu (na przykład celem ich policzenia) ponieważ erozja wykazuje zdolność do podziału niektórych obiektów (posiadających wyraźne przewężenia) na kilka odrębnych obszarów, co wykorzystywane jest w niektórych algorytmach podziału sklejonych obiektów przed ich segmentacją.



Rys. 5.5. Erozja obrazu „Lena”; a) obraz oryginalny, b) erozja 3x3, c) erozja 5x5.



Rys. 5.6. Erozja obrazu „Jet”; a) obraz oryginalny, b) erozja 5x5.

### 5.3.2 Dylatacja

Dylatacja jest przekształceniem odwrotnym do erozji. Aby zdefiniować operację dylatacji zakłada się, że istnieje nieregularny obszar (figura) na obrazie X i koło B o promieniu r, które będzie elementem strukturalnym. Wówczas dylatację figury X elementem B można zdefiniować na trzy różne sposoby:

- figura po dylatacji jest zbiorem środków wszystkich kół B, dla których choć jeden punkt pokrywa się z jakimkolwiek punktem figury wyjściowej.
- koło B przetacza się po zewnętrznej stronie brzegu figury. Kolejne położenia środka koła B wyznaczają brzeg figury po dylatacji.
- analogicznie, jak w przypadku erozji, dylatację można zdefiniować jako **filtr maksymalny**.

W uzupełnieniu tych określeń można wprowadzić jeszcze jedno:

- Dylatację można także zdefiniować jako **negatyw erozji negatywu obrazu** - co w pełni oddaje fakt, że dylatacja jest przekształceniem odwrotnym do erozji.

Operację dylatacji można wykonać posługując się elementem przedstawionym na rys. 5.7.

x	x	x
x	0	x
x	x	x

Rys. 5.7. Element strukturalny dla dylatacji.

Jeżeli punkt centralny i otoczenie elementu strukturalnego (rys. 5.7) zgadza się z lokalną konfiguracją punktów obrazu, to nowa wartość punktu centralnego obrazu przyjmuje wartość 1, ale pod warunkiem, że nie wszystkie wartości otoczenia obrazu są równe 0. W przeciwnym wypadku wartość punktu centralnego obrazu nie ulega zmianie.

Formalnie dylatację można zdefiniować na następujące sposoby:

### **Definicja I**

Zakładamy, że obraz wyjściowy zawiera obszar X wyróżniający się pewną charakterystyczną cechą (np. jasnością). Figura przekształcona przez dylatację to zbiór punktów centralnych wszystkich elementów strukturalnych, których którykolwiek punkt mieści się we wnętrzu obszaru X. Miarą dylatacji jest wielkość elementu strukturalnego.

### **Definicja II**

Dylatację można traktować jako filtr maksymalny. Rozpatrywane otoczenie lokalne punktu jest odpowiednikiem elementu strukturalnego. Dzięki temu pojęcie dylatacji można rozszerzyć na obrazy posiadające wiele stopni szarości, a nawet kolorowe. W przypadku obrazu posiadającego wiele poziomów jasności operację dylatacji wygodnie zapisać jest jako:

$$L'(m,n) = \max_{m_i, n_i \in B(m,n)} (L(m_i, n_i))$$

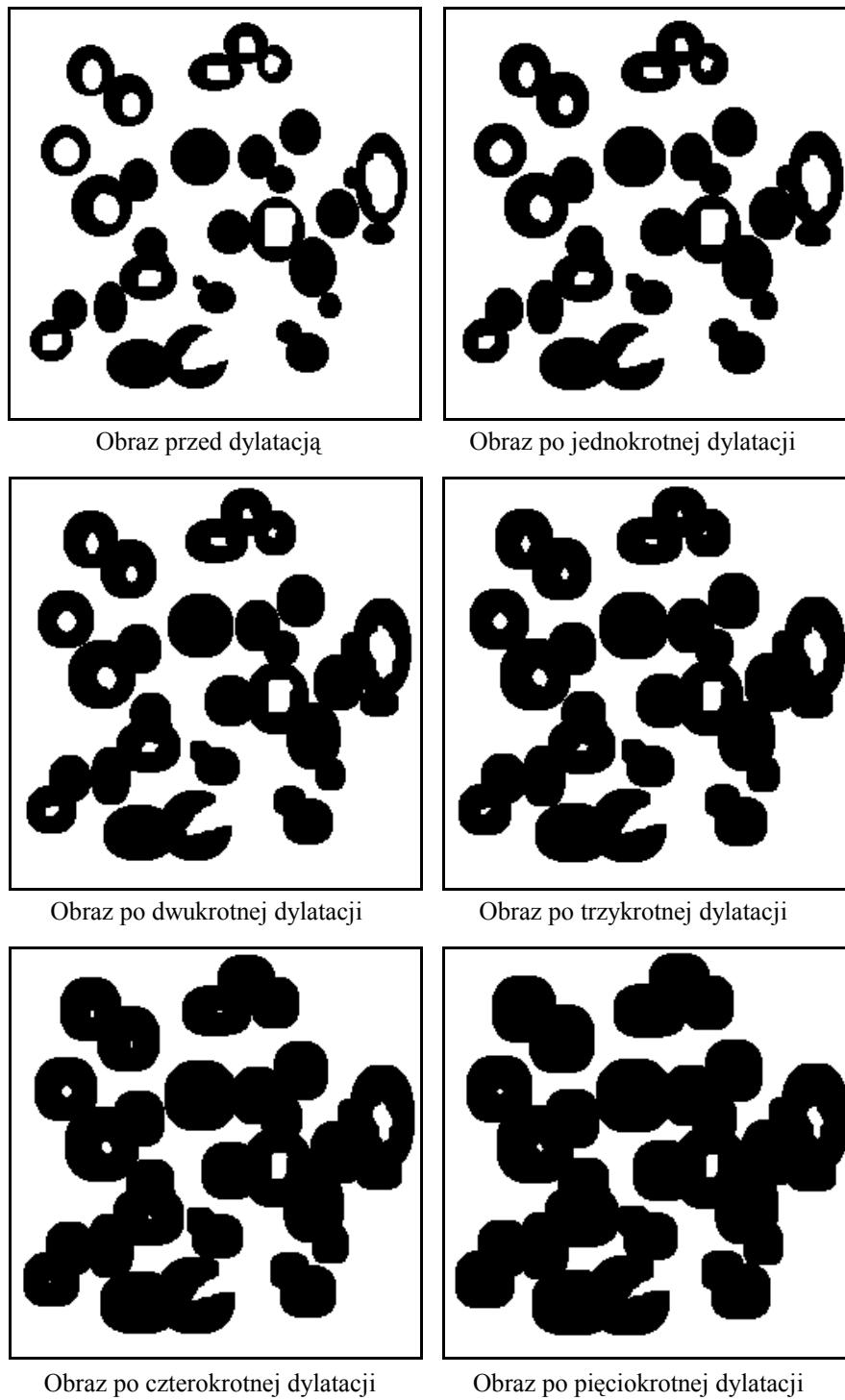
gdzie:  $L(m,n)$  - jasność punktu o współrzędnych  $(m,n)$ ;

$B(m,n)$  - element strukturalny z punktem centralnym o współrzędnych  $(m,n)$ .

Dylatację obrazu kolorowego można zdefiniować jako złożenie niezależnych dylatacji przeprowadzonych na jego poszczególnych składowych.

Podstawowe własności dylatacji są następujące:

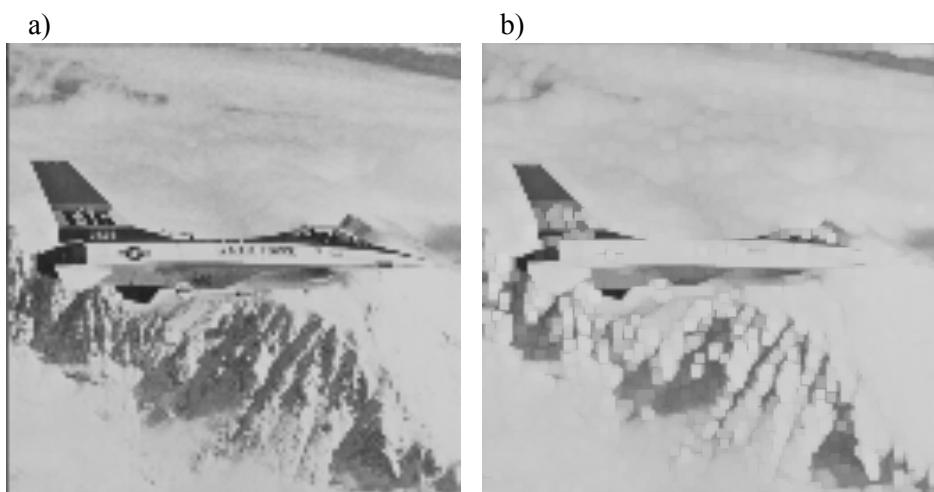
- zamknięcie małych otworów i wąskich „zatok” w konturach obiektów na obrazie;
- zdolność do łączenia obiektów, które położone są blisko siebie;



Rys. 5.8. Przebieg dylatacji prostego obrazu (pokazany w negatywie).



Rys. 5.9. Dylatacja obrazu „Lena”; a) obraz oryginalny, b) dylatacja  $3 \times 3$ , c) dylatacja  $5 \times 5$ .



Rys. 5.10. Dylatacja obrazu „Jet”; a) obraz oryginalny, b) dylatacja  $5 \times 5$ .

Podsumowując można stwierdzić, że dylatacja również (podobnie jak erozja, chociaż w innym sensie) dokonuje generalizacji obrazu. Drobne wklęsłości w wyróżnionych obszarach zostają usunięte. Brzegi wyróżnionych obszarów zostają wygładzone, ich długość zostaje zdecydowanie zmniejszona. Zwiększone natomiast zostają ich powierzchnie. Często dylatacja prowadzi do zrastania się obiektów: mniejsze, blisko położone wyróżnione obszary połączone zostają w większe<sup>31</sup>. Wzrasta także “nasycenie” obrazu.

W związku z podobnym zdefiniowaniem większość cech erozji i dylatacji jest wspólna. Na przykład wspólna jest addytywność przekształcenia oraz możliwość wpływu na wynik przekształcenia poprzez dobór odpowiednich elementów strukturalnych.

<sup>31</sup> Zjawisko to jest wykorzystane przy łączeniu fragmentów obrazu - na przykład celem pomiaru ich powierzchni.

Przebieg procesu dylatacji dla prostego obrazu ziaren prześledzić można na rysunku 5.8. Kolejne kadry na tym obrazie pokazują kolejne iteracje procesu dylatacji przy założeniu, że element strukturalny ma rozmiary  $3 \times 3$  piksele lub wyniki dylatacji z coraz większym elementem strukturalnym. Oczywiście z powodów omówionych w poprzednim podrozdziale rysunek 5.8. przedstawia odpowiednie obiekty i tło w negatywie. Obrazy 5.9 i 5.10 pokazują z kolei skutki procesu dylatacji na rzeczywistych obrazach (już bez brania negatywu).

### 5.3.3 Otwarcie i zamknięcie

Opisane wcześniej przekształcenia obrazu (dylatacja i erozja) posiadają niestety istotną wadę. Zmieniają one w wyraźny sposób pole powierzchni przekształcanych obszarów. Erozja zmniejsza je, a dylatacja zwiększa. Aby wyeliminować tę wadę wprowadzono dwa przekształcenia będące złożeniem poprzednich. Są to **otwarcie** i **zamknięcie**. Zdefiniować je można w sposób następujący:

$$\text{otwarcie} = \text{erozja} + \text{dylatacja}$$

$$\text{zamknięcie} = \text{dylatacja} + \text{erozja}$$

**Otwarcie** polega na przetaczaniu koła B po wewnętrznej stronie brzegu figury i odrzuceniu wszystkich tych punktów, które nie mogą być osiągnięte przez koło. **Zamknięcie** polega natomiast na przetaczaniu koła B po zewnętrznej stronie brzegu figury i dodaniu do niej wszystkich tych punktów, które nie mogą być osiągnięte przez koło. W odróżnieniu od erozji i dylatacji w omawianych tu przekształceniach rozpatrujemy całe koło B, a nie tylko jego środek.

Przekształcenia te posiadają następujące własności:

- Są to przekształcenia rosnące, tzn. jeżeli obszar A zawiera się w obszarze B, to po przekształceniu obszar A zawiera się w obszarze B. Reguła ta funkcjonuje zarówno w przypadku otwarcia jak i zamknięcia, nawiąsem mówiąc tą samą własność posiada także erozja i dylatacja.
- Jeżeli dla pewnej figury  $f$  wykonamy omawiane wyżej przekształcenia i otrzymamy następujące nowe figury:

$$E(f) - \text{erozja figury } f,$$

$$O(f) - \text{otwarcie figury } f,$$

$$C(f) - \text{zamknięcie figury } f,$$

$$D(f) - \text{dylatacja figury } f,$$

to dla figur tych (będących skutkiem przekształceń) zachodzi następująca nierówność:

$$\#E(f) \leq \#O(f) \leq \#C(f) \leq \#D(f),$$

gdzie:  $\#$  - oznacza wielkość (liczbę pikseli) należących do określonej figury.

- Przekształcenia otwarcia i zamknięcia są niezmienne względem siebie. Własność tą można zapisać następującym wzorem:

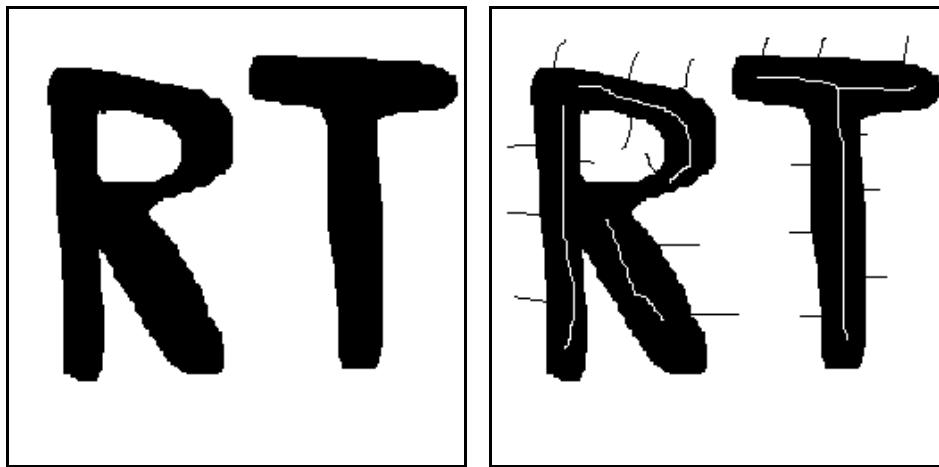
$$O(O(f)) = O(f) \text{ oraz } C(C(f)) = C(f)$$

Własność ta odróżnia otwarcie i zamknięcie od erozji i dylatacji, które są addytywne.

Własności „użytkowe”, operacji otwarcia i zamknięcia dla przypadku obrazów binarnych są następujące:

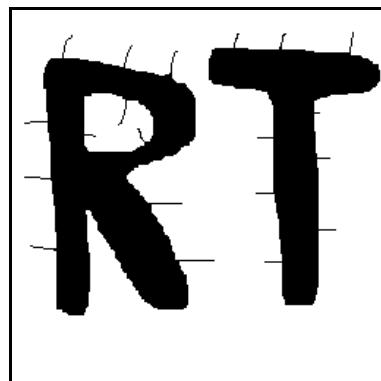
- otwarcie usuwa drobne obiekty i drobne szczegóły, jak półwyspy i wypustki, nie zmieniając wielkości zasadniczej części figury, może też rozłączyć niektóre obiekty z przewężeniami;
- zamknięcie wypełnia wąskie wcięcia i zatoki oraz drobne otwory wewnętrz obiektu, nie zmieniając wielkości jego zasadniczej części, może też połączyć leżące blisko siebie obiekty;
- obydwie operacje nie zmieniają kształtu ani wymiarów dużych obiektów o wyrównanym, gładkim brzegu.

Operacje otwarcia i domknięcia dają możliwość usuwania z obrazów pewnych szczególnie uciążliwych zakłóceń. Na rysunku 5.11. pokazano prosty obraz binarny i jego zakłóconą wersję. Widać, że zakłócenia (czarne i białe linie) są na tyle kłopotliwe, że ich usunięcie technikami „klasycznej” filtracji nasuwałoby sporo trudności.



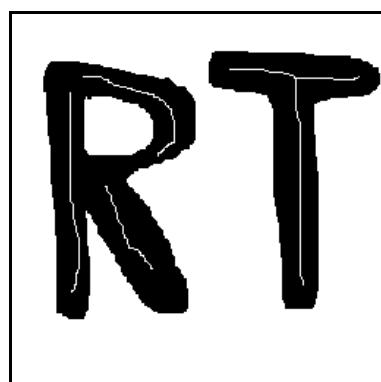
Rys. 5.11. Obraz binarny i jego zakłócona wersja.

Na rysunku 5.12 pokazano jednak wynik operacji domknięcia (z elementem strukturalnym o rozmiarach 3 x 3 piksele) wykonanej na zakłóconym obrazie. Widać, że niektóre zakłócenia (białe pasy) zostały usunięte.

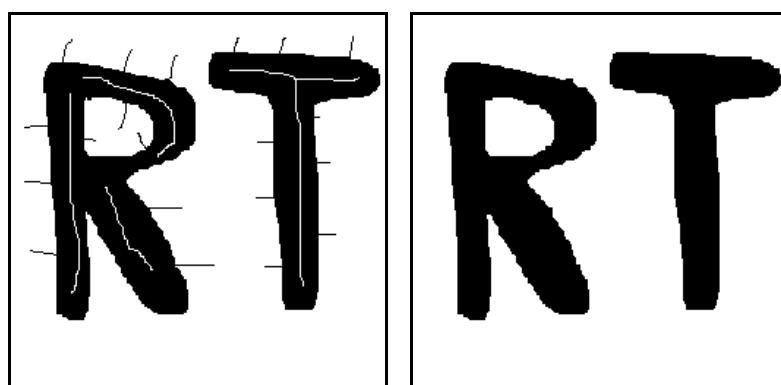


Rys. 5.12. Zakłócony obraz po operacji domknięcia jest wolny od części zakłóceń.

Z kolei na rysunku 5.13. pokazano wynik wykonania na zakłóconym obrazie operacji otwarcia. Tym razem usunięte zostały zbyteczne czarne linie.



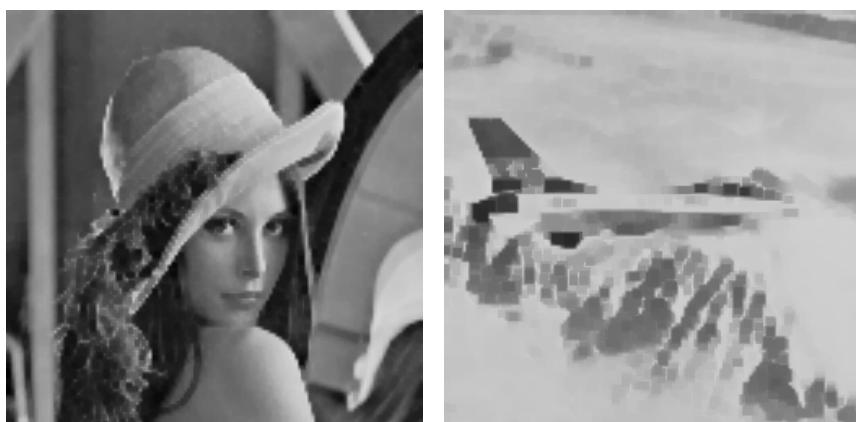
Rys. 5.13. Zakłócony obraz po operacji otwarcia jest wolny od innego typu zakłóceń.



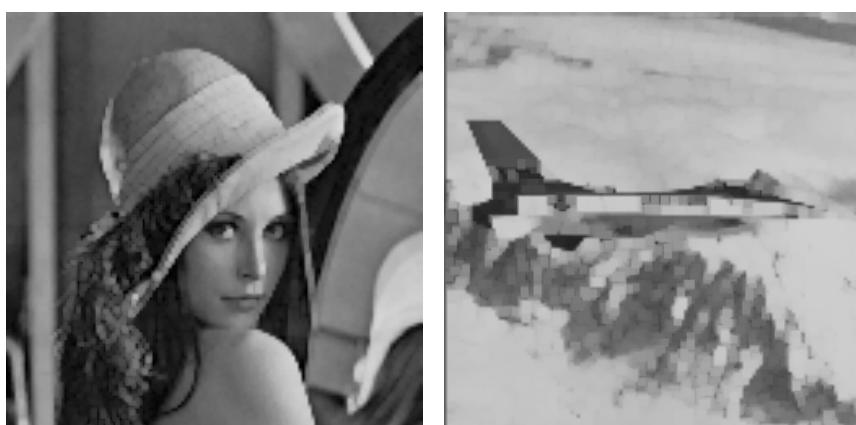
Rys. 5.14. Zakłócony obraz po operacji domknięcia i otwarcia jest wolny od wszystkich zakłóceń.

Skoro otwarcie usuwa jedne, a domknięcie drugie zakłócenia - jest w pełni logiczne, że pod wykonaniu sekwencji operacji najpierw domknięcia, a potem otwarcia - znikają wszystkie zakłócenia (rys. 5.14).

Operacje domknięcia i otwarcia - obok tego, że usuwają zakłócenia - niestety trochę znieksztalcają obraz. Najłatwiej to zauważać na obrazach rzeczywistych. Na rysunku 5.15. pokazano wynik operacji domknięcia z elementem strukturalnym o rozmiarach  $3 \times 3$  na obrazie „Lena” i wynik operacji domknięcia z elementem strukturalnym o rozmiarach  $5 \times 5$  na obrazie „Jet”. Natomiast na rysunku 5.16 pokazano efekt działania operacji otwarcia (odpowiednio także z elementem strukturalnym o rozmiarach  $3 \times 3$  i  $5 \times 5$ ) na tych samych obrazach.



Rys. 5.15. Wynik operacji domknięcia (z elementem strukturalnym o rozmiarach  $3 \times 3$ ) na obrazie „Lena” i (z elementem strukturalnym o rozmiarach  $5 \times 5$ ) na obrazie „Jet”.



Rys. 5.16. Wynik operacji otwarcia (z elementem strukturalnym o rozmiarach  $3 \times 3$ ) na obrazie „Lena” i (z elementem strukturalnym o rozmiarach  $5 \times 5$ ) na obrazie „Jet”.

### 5.3.4 Otwarcie i zamknięcie właściwe, automediana

Kolejne przekształcenia morfologiczne mogą być budowane jako coraz bardziej złożone transformacje obrazu - w oparciu o zdefiniowane wcześniej otwarcie i zamknięcie. Serię interesujących przekształceń figury  $f$  można na przykład zrealizować przez porównanie złożen otwarcia i zamknięcia - z obrazem wyjściowym figury  $f$  zanim wykonano na niej jakiekolwiek przekształcenia. Powstać mogą w ten sposób następujące przekształcenia:

- Otwarcie właściwe:  $\Theta(f) = \min(f, C(O(C(f))))$
- Zamknięcie właściwe:  $\Theta'(f) = \max(f, O(C(O(f))))$
- Automediana:  $A(f) = \max(O(C(O(f)))), \min(f, C(O(C(f))))$

Przekształcenia te prowadzić mogą do regularyzacji konturów obiektów (zwłaszcza na obrazach binarnych) i do swoistej „filtracji kształtów” obiektów (w odróżnieniu od dyskutowanej w rozdziale 3 filtracji poszczególnych punktów). Zagadnienie to nie będzie tu dokładniej dyskutowane.

### 5.3.5 Detekcja ekstremów<sup>32</sup>

Aby wyodrębnić z obrazu lokalne ekstrema (minima i maksima) można wykorzystać zdefiniowane wcześniej przekształcenia otwarcia i zamknięcia. I tak aby wyszukać lokalne maksima należy od wyniku otwarcia danego obrazu odjąć obraz wyjściowy, a następnie dokonać binaryzacji z dolnym progiem otrzymanej różnicicy:

$$M(f) = B(O(f) - f),$$

gdzie:  $B(f)$  - binaryzacja z dolnym progiem obrazu  $f$ .



Rys. 5.17. Wynik działania operacji *top-hat* w wersji angażującej operację otwarcia.

---

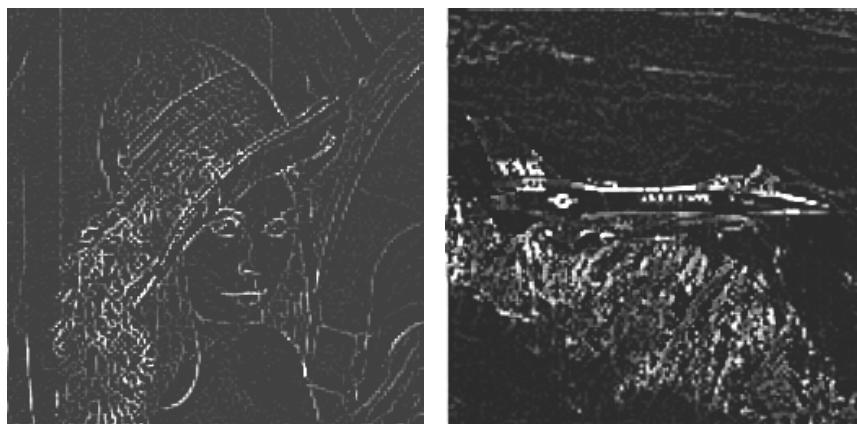
<sup>32</sup> ang. *top-hat* (czubek kapelusza) - pod taką nazwą zwykle odnaleźć można przekształcenie w menu różnych programów przetwarzających obrazy.

Efekt działania tej bardzo użytecznej operacji na rzeczywistych obrazach pokazuje rysunek 5.17.

Analogicznie, aby wyodrębnić lokalne minima obrazu, należy dokonać podobnej operacji, z tym, że pierwszą operacją będzie zamknięcie:

$$m(f) = B(C(f) - f)$$

Efekt tych operacji zależy w dużym stopniu od przyjętych parametrów: wielkości otwarcia (lub odpowiednio zamknięcia) i progu binaryzacji. Efekt działania tej bardzo użytecznej operacji na rzeczywistych obrazach pokazuje rysunek 5.18.



Rys. 5.18. Wynik działania operacji *top-hat* w wersji angażującej operację zamknięcia.

Łatwo zauważać, że pod pewnymi względami operacja *top-hat* daje podobny wynik, jak operatory wydzielające krawędzie (na przykład Laplasjan).

### 5.3.6 Ścienianie

Ścienianie jest wspólną nazwą dla pewnego podzbioru przekształceń morfologicznych przeprowadzanych w następujący sposób. Ścienianie obiektu X przy użyciu elementu strukturalnego B polega na przyłożeniu tego elementu do każdego punktu obrazu w ten sposób, że punkt centralny pokrywa się z analizowanym punktem i podjęciu jednej z dwóch decyzji:

- nie zmieniać punktu, gdy element nie pokrywa się z jego sąsiedztwem
- zamienić wartość punktu na 0, jeżeli element strukturalny pasuje do sąsiedztwa analizowanego punktu

Ścienianie może być powtarzane wielokrotnie, aż do momentu gdy następny krok nie wprowadza żadnych zmian w obrazie.

Operację ścieniania można wykonać posługując się elementem przedstawionym na rys. 5.19.

<table border="1"><tr><td>x</td><td>1</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	x	1	1	1	1	1	1	1	1	<table border="1"><tr><td>1</td><td>1</td><td>x</td></tr><tr><td>1</td><td>1</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	1	1	x	1	1	1	1	1	1	<table border="1"><tr><td>1</td><td>1</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr><tr><td>1</td><td>1</td><td>x</td></tr></table>	1	1	1	1	1	1	1	1	x	<table border="1"><tr><td>1</td><td>1</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr><tr><td>x</td><td>1</td><td>1</td></tr></table>	1	1	1	1	1	1	x	1	1
x	1	1																																					
1	1	1																																					
1	1	1																																					
1	1	x																																					
1	1	1																																					
1	1	1																																					
1	1	1																																					
1	1	1																																					
1	1	x																																					
1	1	1																																					
1	1	1																																					
x	1	1																																					

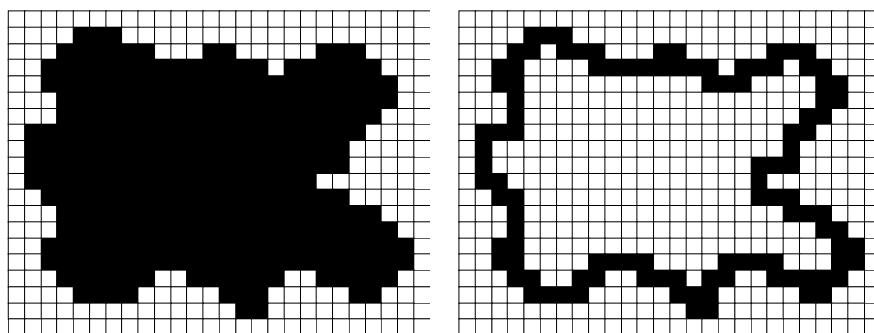
Rys. 5.19. Przykłady elementów strukturalnych wykorzystywanych przy ścienianiu.

Najczęściej element strukturalny jest rotowany (przekręcany o kąt  $90^\circ$  w przypadku siatki kwadratowej) pomiędzy kolejnymi operacjami. Wynikiem ścieniania jest zawsze obraz binarny. W większości przypadków obrazem wejściowym dla ścieniania jest również obraz binarny, chociaż to nie jest konieczne.

Dla przykładu, przy zastosowaniu następującego elementu strukturalnego pokazanego na rysunku 5.20 uzyskuje się wyodrębnienie brzegu analizowanej figury (rys. 5.21).

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & X & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

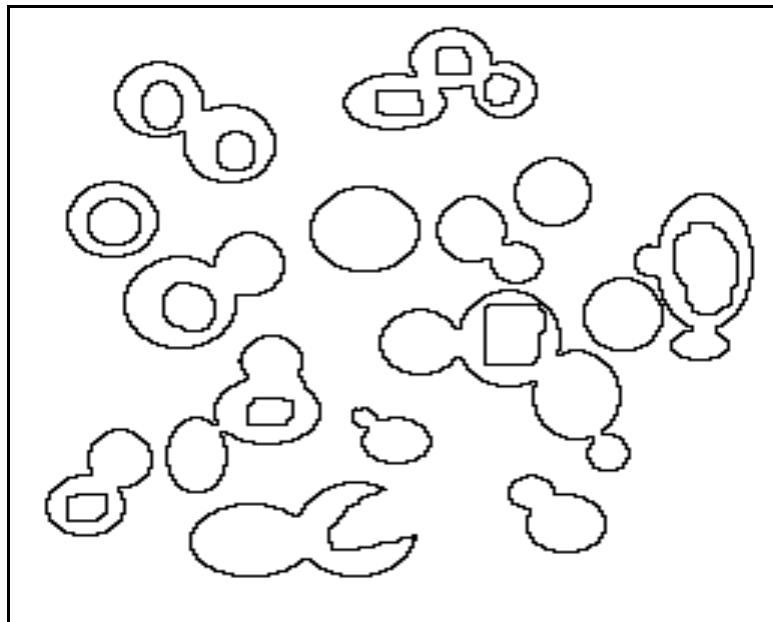
Rys. 5.20. Element strukturalny wykorzystywany przy ścienianiu.



Rys. 5.21. Figura przed i po ścienaniu elementem strukturalnym z rysunku 5.20.

Efekt wydzielania krawędzi jako wyniku ścieniania daje się jeszcze lepiej przesledzić na rysunku 5.22.

Cechą charakterystyczną ścieniania jest to, że figura po ścienaniu zawiera się w figurze wyjściowej. W dalszej części, w kolejnych podrozdziałach przedstawione zostaną przykłady przekształceń ukierunkowanych na uzyskanie bardzo konkretnych celów opartych o operację ścieniania wraz z pewnymi uzupełnieniami.



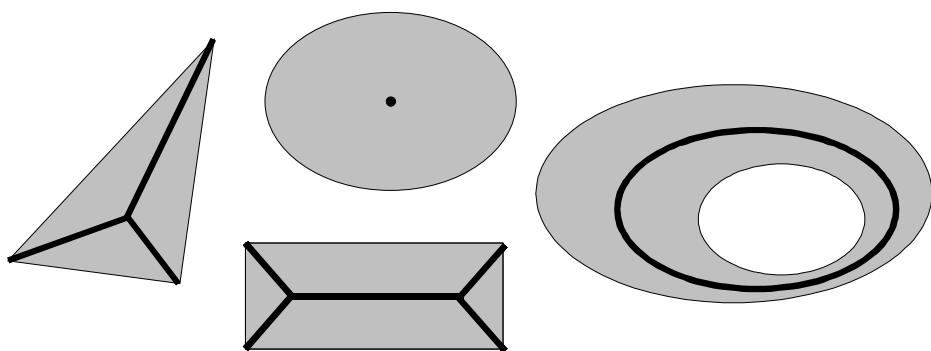
Rys. 5.22. Bardziej złożony obraz przed i po ścienianiu elementem strukturalnym z rysunku 5.20.

#### 5.3.6.1 Szkieletyzacja

Szkieletyzacja jest operacją pozwalającą wyodrębnić osiowe punkty (szkielety) figur w analizowanym obszarze.

##### Definicja

Szkielet figury jest zbiorem wszystkich punktów, które są równoodległe od co najmniej dwóch punktów należących do brzegu. Przykłady niektórych figur geometrycznych i ich szkieletów przedstawia rysunek 5.23.



Rys. 5.23. Wybrane figury i ich szkielety.

Szkielet figury jest znacznie mniejszy od niej, a w pełni odzwierciedla jej podstawowe topologiczne własności. Analiza szkieletów figur umożliwia przeprowadzenie następujących analiz obrazu:

- klasyfikacja częstek na podstawie ich kształtu;
- określenie orientacji podłużnych obiektów;
- rozdzielenie “posklejanych” obiektów;
- wyznaczenie linii środkowej szerszych linii;
- symulacja procesów rozrostu częstek oraz tworzenia struktury ziarnistej.

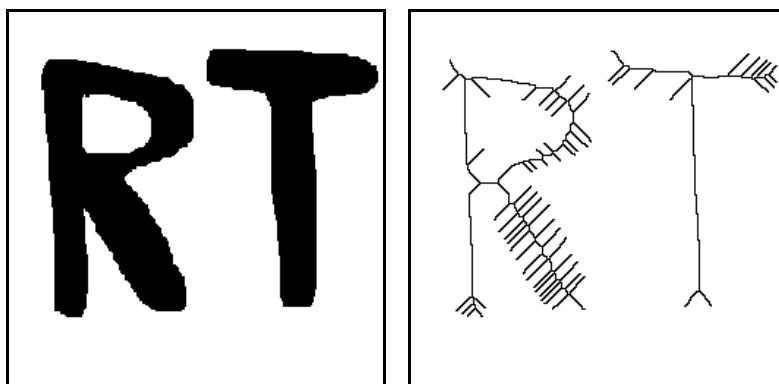
Szkieletyzacja może być realizowana jako ścienianie z elementem strukturalnym pokazanym na rysunku 5.24.

$$\begin{bmatrix} X & 0 & X \\ X & 1 & X \\ 1 & 1 & 1 \end{bmatrix}$$

Rys. 5.24. Element strukturalny używany do ścieniania przy szkieletyzacji.

W praktyce do ścieniania często stosowane są różne elementy strukturalne na przemiennie, na przykład

$$\begin{bmatrix} 1 & 1 & 1 \\ X & 1 & X \\ 0 & 0 & 0 \end{bmatrix} \text{ i } \begin{bmatrix} 1 & X & 1 \\ X & 1 & X \\ 0 & 0 & 0 \end{bmatrix}^{33} \text{ lub } \begin{bmatrix} 1 & 1 & X \\ 1 & 1 & 0 \\ 1 & 1 & X \end{bmatrix}^{34}.$$



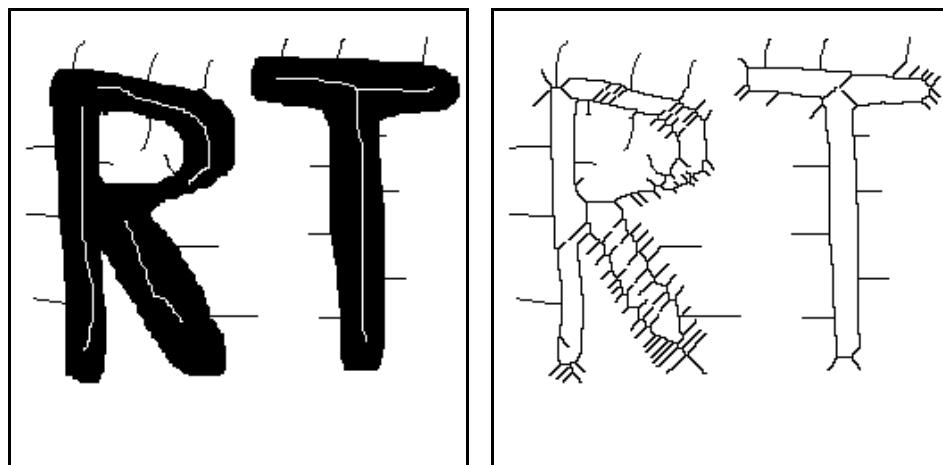
Rys. 5.25. Efekt szkieletyzacji prostego obrazu binarnego (po lewej obraz przed szkieletyzacją, po prawej po szkieletyzacji).

<sup>33</sup> tzw. element “L” alfabetu Golay'a

<sup>34</sup> tzw. element “M” alfabetu Golay'a

Wynik szkieletyzacji przeprowadzonej opisanyem wyżej sposobem na prostym obrazie binarnym pokazano na rysunku 5.25.

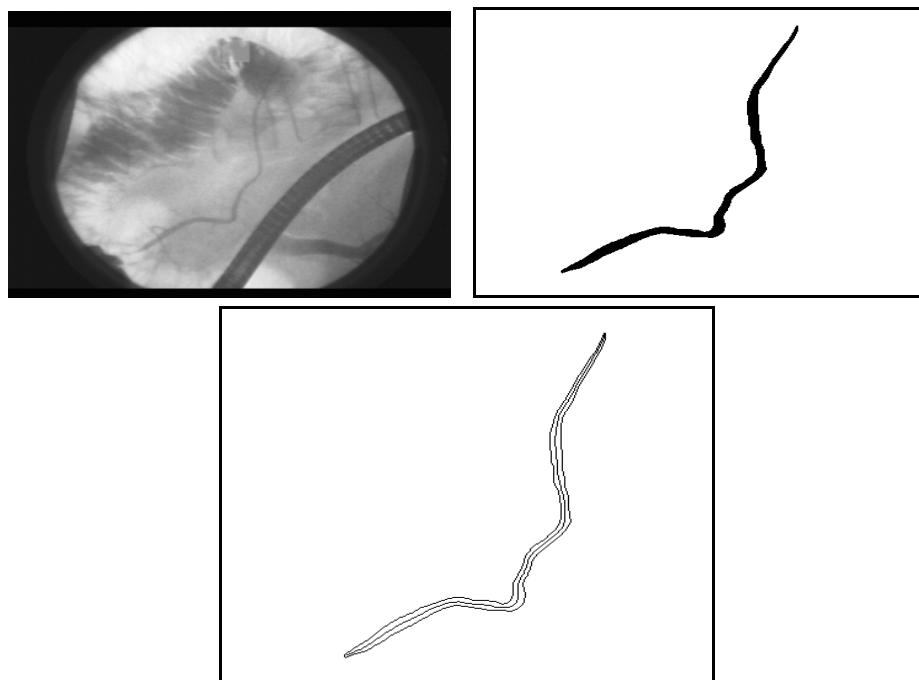
Niestety, proces szkieletyzacji może wprowadzać do obrazu pewne „artefakty” w postaci bocznego „gałżkowania” linii szkieletu. Efekt ten staje się szczególnie kłopotliwy jeśli oryginalna figura posiada zakłócenia, które mogą mieć przemożny wpływ na kształt szkieletu (rys. 5.26).



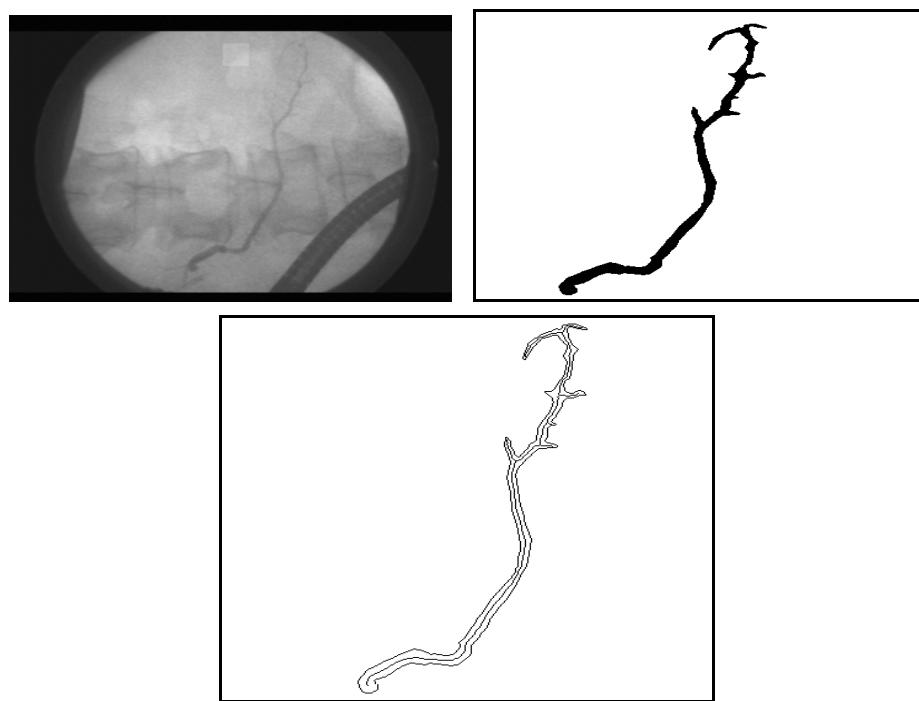
Rys. 5.26. Efekt szkieletyzacji zakłóconego obrazu binarnego (po lewej obraz przed szkieletyzacją, po prawej po szkieletyzacji).

Pokazywane wyżej szkielety prostych binarnych obiektów graficznych mogły być traktowane po trosze jako „sztuka dla sztuki”. Szkieletyzacja jest jednak w rzeczywistości techniką przetwarzania obrazu o bardzo dużym znaczeniu aplikacyjnym. Przykładowym zadaniem praktycznym, w którym ważną rolę odgrywa szkieletyzacja, jest kilkakrotnie już przywoływane w tej książce zadanie analizy przebiegu głównego przewodu trzustkowego na obrazach ERCP. Na rysunku 5.27 pokazano przykładowy przebieg prawidłowego przewodu trzustkowego - w postaci pełnego obrazu źródłowego, jego zbinaryzowanego fragmentu odpowiadającego interesującemu przewodowi oraz wykrytego metodami morfologicznymi szkieletu. Widać, że szkielet zawiera w praktyce tę samą informację, co obraz źródłowy - zajmując przy tym zdecydowanie mniej miejsca w pamięci komputera.

Szkielet może dobrze odtwarzać także bardziej skomplikowane kształty. Na rysunku 5.28 pokazano - analogicznie jak na rysunku 5.27 - przebieg procesu szkieletyzacji przewodu trzustkowego pacjenta cierpiącego na przewlekłe zapalenie trzustki. Widać wyraźnie, że zdecydowanie bardziej złożony kształt analizowanego obiektu także i w tym przypadku znalazł swój poprawny odpowiednik w postaci poprawnie wyznaczonego szkieletu.

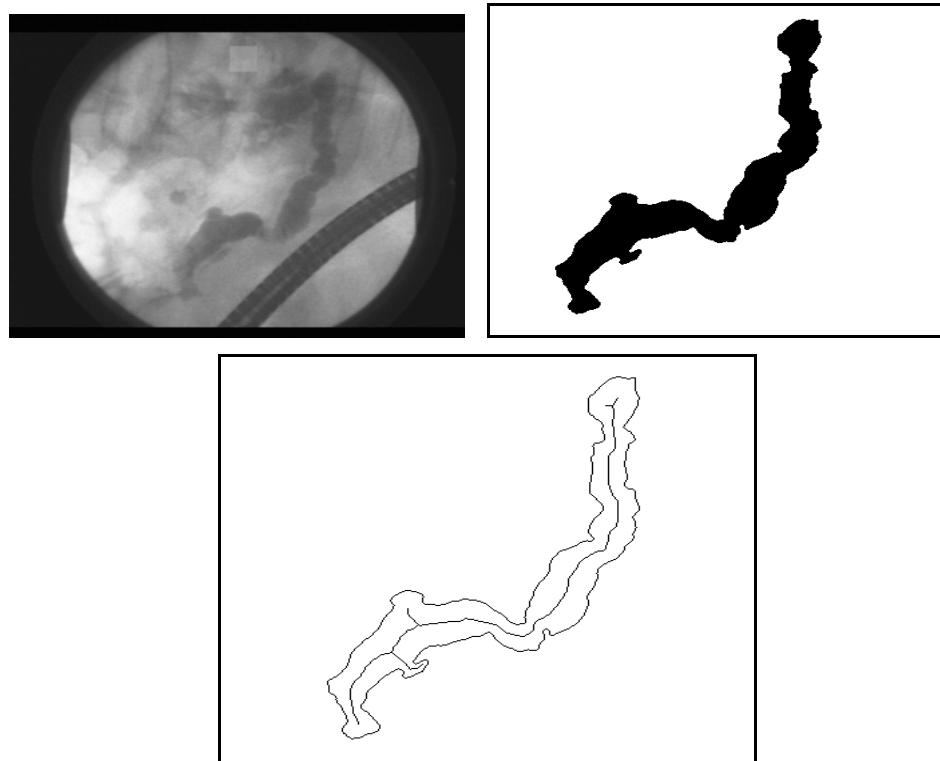


Rys. 5.27. Obraz ERCP - oryginalny, zbinaryzowany i po szkieletyzacji.



Rys. 5.28. Inny przykład poprawnej szkieletyzacji obrazu ERCP.

Nie zawsze jednak sytuacja jest tak komfortowa, jak na przytoczonym wyżej rysunku. Niekiedy operacja szkieletyzacji wnosi pewne zniekształcenia (artefakty), które powodują, że szkielet nie reprezentuje dostatecznie dobrze podlegającego szkieletyzacji obiektu. Na rysunku 5.29. pokazano kolejny obraz ERCP, pokazujący przewód trzustkowy pacjenta, u którego wykryto raka trzustki. Niestety, jak łatwo zauważać algorytm szkieletyzacji wprowadził w tym przypadku swój „twórczy wkład” do kształtu obrazu - pojawiły się bowiem w szkielecie fałszywe odgałęzienia boczne, których przyczyną były nierównomierności brzegu obiektu (cysty występujące w dotknietym choroba przewodzie trzustkowym). Takie zafałszowanie rzeczywistego kształtu obiektu przez algorytm wydzielania szkieletu (zwane popularnie „gaławkowaniem”) jest jednym z problemów, z którymi musi się uporać badacz stosujący metody morfologiczne. Techniki, jakie są w związku z tym stosowane, zostaną omówione w kolejnym podrozdziale.



Rys. 5.29. Przykład niepoprawnej szkieletyzacji obrazu ERCP.

### 5.3.6.2 Obcinanie gałęzi

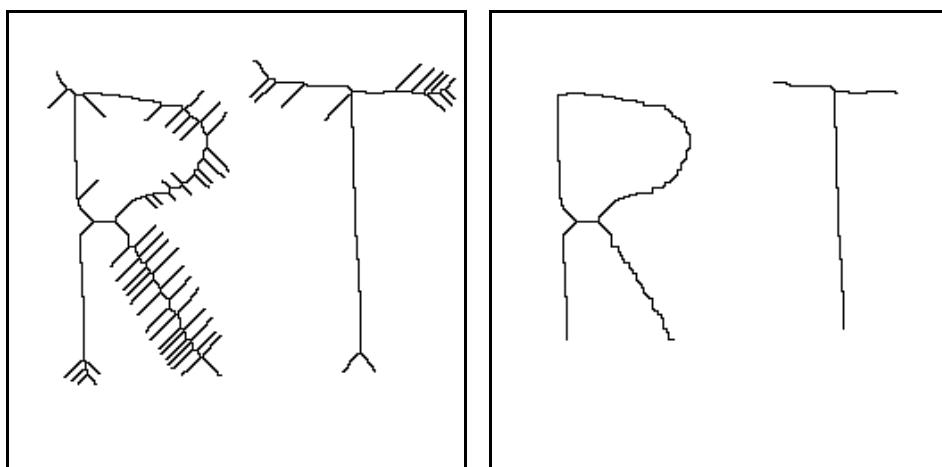
W praktycznym zastosowaniu szkielet figury w dużym stopniu zależy od regularności brzegu figury. Każda, nawet niewielka, nieregularność powoduje powstanie

dodatkowego, niepotrzebnego odgałęzienia wynikowego szkieletu. Takie zbyteczne odnogi mogą powstać również w dość regularnych figurach, które nachylone są jednak pod niewłaściwym kątem do siatki.

Aby usunąć tego typu zniekształcenia można zastosować algorytm “obcinania gałęzi”. Polega on na stopniowym redukowaniu odcinków posiadających wolne zakończenie. W ostateczności pozostają jedynie zamknięte pętle i odcinki przecinające brzeg obrazu. Zazwyczaj jednak obcinanie gałęzi przeprowadza się w ograniczonym stopniu, zależnym od ilości przeprowadzonych iteracji. Algorytm obcinania gałęzi może być realizowany jako ścienianie przy elemencie strukturalnym pokazanym na rysunku 5.30.

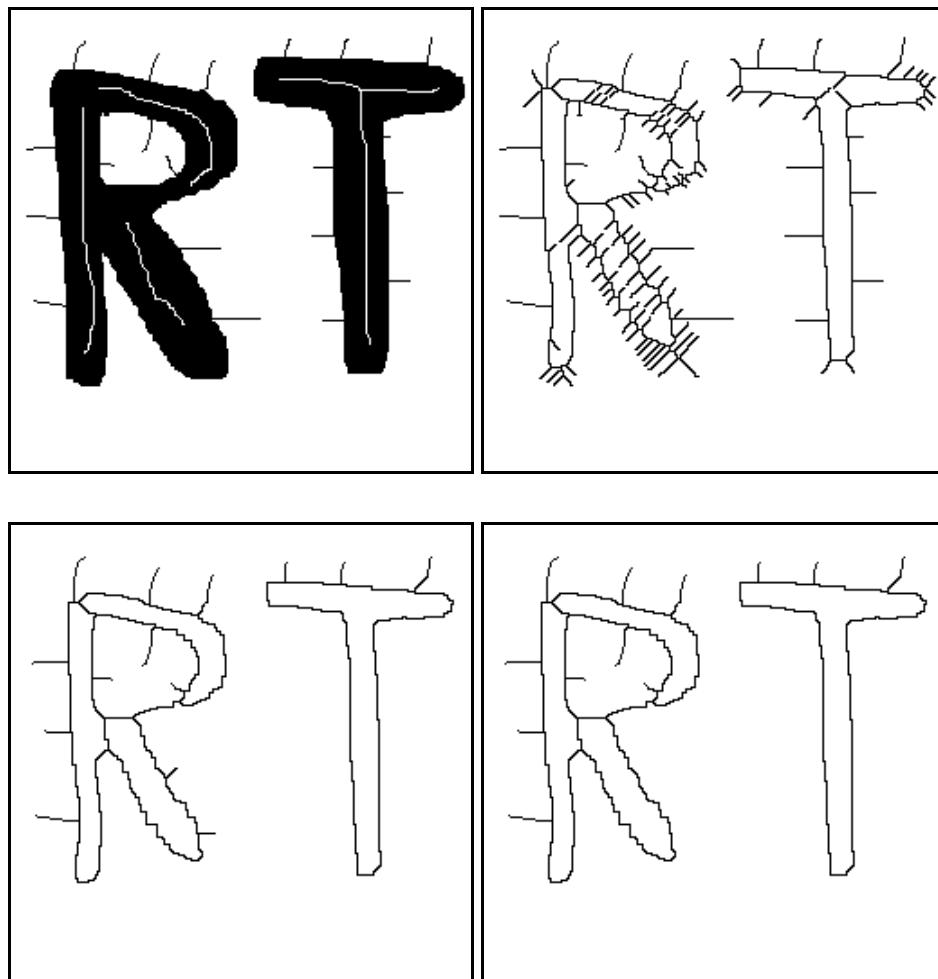
$$\begin{bmatrix} 0 & X & X \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Rys. 5.30. Element strukturalny, którego użycie pozwala na uzyskanie efektu obcinania gałęzi.



Rys. 5.31. Efekt działania algorytmu obcinania gałęzi na szkielecie prostego obrazu binarnego.

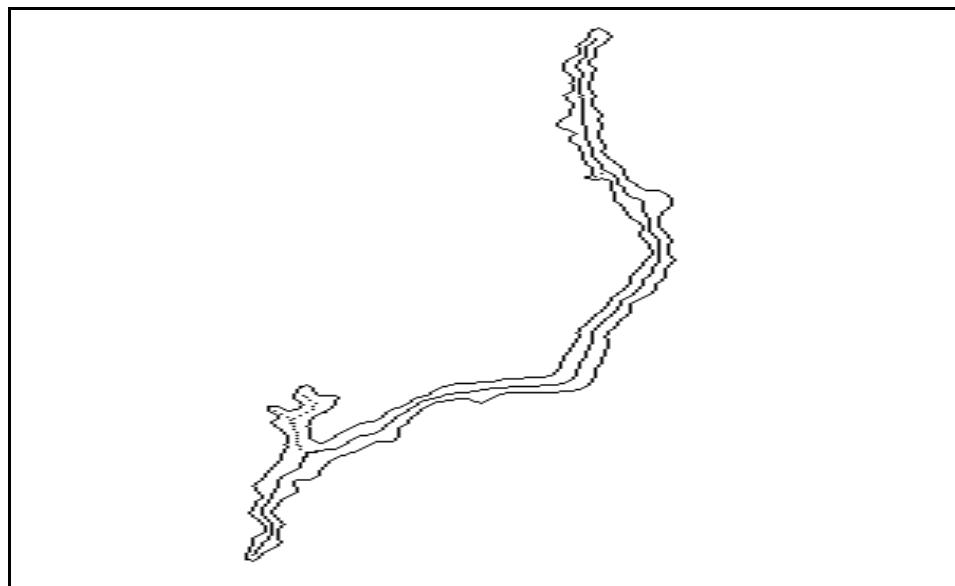
Efekty algorytmu obcinania gałęzi można obejrzeć na rysunku 5.31 (dla prostszego obrazu, gdzie wystarczyło 5 iteracji z użyciem elementu strukturalnego 5.30) i na rysunku 5.32 gdzie na obrazie bardziej skomplikowanym (na skutek działania zakłóceń) pokazano efekt obcinania odgałęzień po 5 i po 10 iteracjach. Jak widać w tym drugim przypadku nawet wydłużenie liczby iteracji nie jest w stanie doprowadzić do usunięcia wszystkich artefaktów związanych z procesem szkieletyzacji.



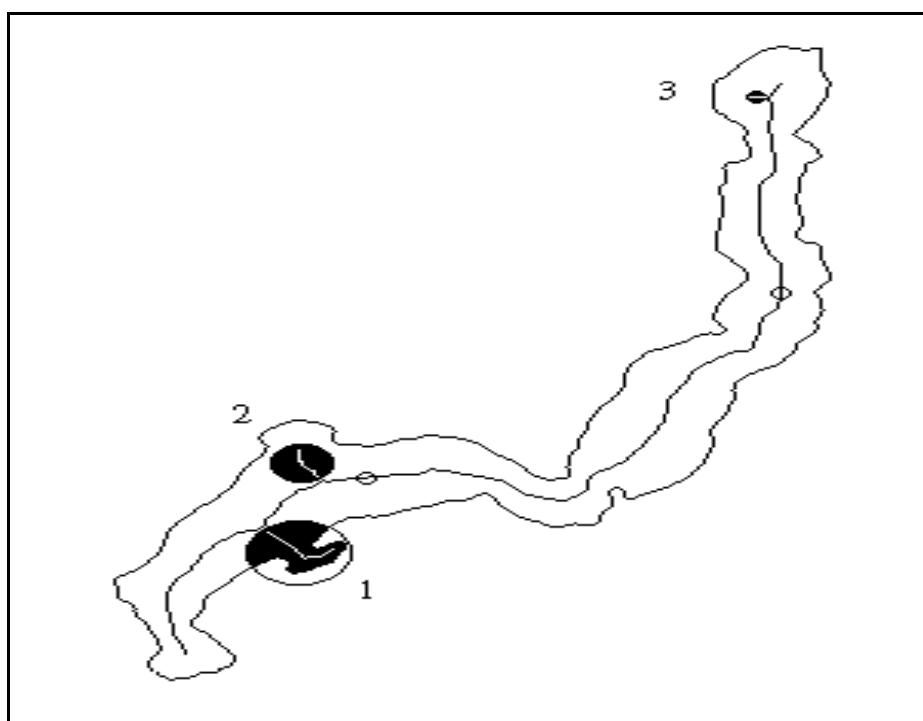
Rys. 5.32. Efekt działania algorytmu obcinania gałęzi na szkielecie zakłóconego obrazu binarnego.

Warto też obejrzeć skutki działania algorytmu usuwania odgałęzień bocznych na szkieletach obrazów rzeczywistych. W tym celu znowu odwołamy się do obrazu ERCP - szkielet na rysunku 5.33 pokazuje, że możliwe jest skuteczne usunięcie artefaktów na obrazie szkieletu przewodu trzustkowego u pacjenta cierpiącego na raka wątroby.

Do obcinania gałęzi bocznych na szkieletach figur o charakterze ciągłych linii (na przykład wspomniane wyżej obrazy ERCP, ale także przetwarzane cyfrowo wykresy z przyrządów rejestrujących, rozpoznawanie pisma, itp.) wykorzystać można algorytm eliminacji odgałęzień opracowany przez dr Marka Ogięle. Działanie tego algorytmu pokazuje rysunek 5.34.



Rys. 5.33. Efekt działania algorytmu obcinania gałęzi na szkielecie obrazu ERCP.

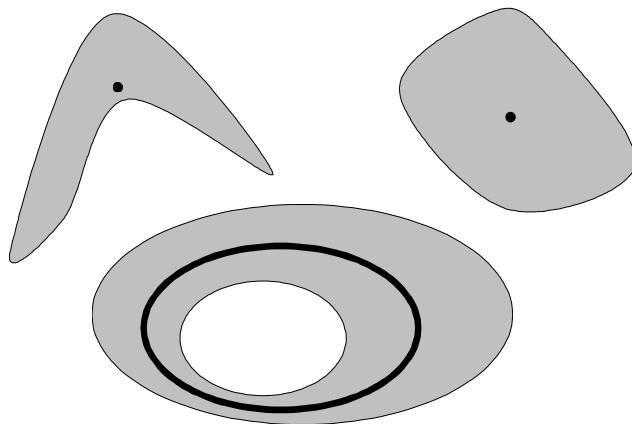


Rys. 5.34. Wynik działania algorytmu dr Ogieli.

## 5.4 Inne przekształcenia morfologiczne

### 5.4.1 Wyznaczanie centroidów

Centroid jest szkieletem figury z szczególnie mocno obciętymi gałęziami. Operacja wyznaczania centroidu sprowadza figurę do punktu w przypadku, gdy figura nie posiada “otworów” lub do pętli, gdy takowe posiada. Na rysunku 5.35. pokazano centroidy wyznaczone dla kilku przykładowych figur.



Rys. 5.35. Figury i ich centroidy.

Wyznaczenie centroidu realizowane jest jako ścienianie przy użyciu dwóch elementów strukturalnych:

$$\begin{bmatrix} 0 & X & X \\ 0 & 1 & 1 \\ 0 & 0 & X \end{bmatrix}, \text{ a następnie } \begin{bmatrix} X & 0 & X \\ 0 & 1 & 1 \\ X & 0 & X \end{bmatrix}.$$

Kolejne iteracje wykonywane są dopóki wprowadzane są jakieś zmiany w obrazie.

### 5.4.2 Pogrubianie

Pogrubianie obiektu  $X$  przy użyciu elementu strukturalnego  $B$  polega na przyłożeniu tego elementu do każdego punktu obrazu w ten sposób, że punkt centralny pokrywa się z analizowanym punktem i podjęciu jednej z dwóch decyzji:

- nie zmieniać punktu, gdy element nie pokrywa się z jego sąsiedztwem
- zamienić wartość punktu na 1, jeżeli element strukturalny pasuje do sąsiedztwa analizowanego punktu obrazu

Operacja ta jest przeprowadzana wielokrotnie, aż do braku zmian wprowadzanych przez operację. Najczęściej element strukturalny jest przekręcany (o kąt  $90^\circ$  w przypadku siatki kwadratowej) pomiędzy kolejnymi operacjami.

Pogrubianie jest przekształceniem odwrotnym do ścieniania. Dokładniej **ścienianie zbioru X elementem B jest równoważne dopełnieniu zbioru powstalego w wyniku pogrubiania dopełnienia zbioru X elementem dopełniającym B**.

Operację pogrubiania dla izolowanych punktów można wykonać posługując się elementem przedstawionym na rys. 5.36.

<table border="1"><tr><td>x</td><td>1</td><td>x</td></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>0</td><td>0</td></tr></table>	x	1	x	0	0	0	0	0	0	<table border="1"><tr><td>0</td><td>0</td><td>x</td></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>0</td><td>x</td></tr></table>	0	0	x	0	0	1	0	0	x	<table border="1"><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>x</td><td>1</td><td>x</td></tr></table>	0	0	0	0	0	0	x	1	x	<table border="1"><tr><td>x</td><td>0</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>x</td><td>0</td><td>0</td></tr></table>	x	0	0	1	0	0	x	0	0
x	1	x																																					
0	0	0																																					
0	0	0																																					
0	0	x																																					
0	0	1																																					
0	0	x																																					
0	0	0																																					
0	0	0																																					
x	1	x																																					
x	0	0																																					
1	0	0																																					
x	0	0																																					

Rys. 5.36. Element strukturalny wykorzystywany przy pogrubianiu.

Jeżeli punkt centralny i otoczenie rotującego elementu strukturalnego (rys. 5.36) zgadza się z lokalną konfiguracją punktów obrazu to nowa wartość punktu centralnego obrazu przyjmuje wartość 1. W przeciwnym przypadku punkt centralny nie ulega zmianie.

#### 5.4.3 Dylatacja bez stykania obszarów (SKIZ)

Jednym z podstawowych problemów stojących przed komputerową analizą obrazu jest wydzielenie poszczególnych, regularnych obszarów obrazu. Zadanie to jest szczególnie trudne, gdy poszczególne obiekty stykają się lub nawet częściowo zachodzą na siebie. Jako pierwszy krok wykonać można wtedy erozję, która rozdzieli sklejone obszary. Powstałe jednak w ten sposób obszary mają dużo mniejszą powierzchnię, niż wynikowe. Aby powrócić do wyjściowej powierzchni nie można zastosować normalnej dylatacji, gdyż powiększone obszary połączą się ponownie. Aby temu zapobiec, trzeba zastosować specjalne przekształcenie, które powiększy powierzchnię obszarów zachowując pewien odstęp pomiędzy nimi. Przekształcenie to nazywane jest dylatacją bez stykania obszarów i może być realizowane jako pogrubianie z następującym elementem strukturalnym:

$$\begin{bmatrix} X & X & X \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Gdy opisaną wyżej metodę prowadzi się cyklicznie, aż do braku zmian w analizowanym obrazie uzyskuje się przekształcenie zwane SKIZ<sup>35</sup>. Skrót ten oznacza

<sup>35</sup> ang. *skeleton by influence zone*

“szkielet strefy wpływów”. Strefa wpływów danego punktu definiowana jest jako zbiór wszystkich punktów obrazu, dla których odległość do danego punktu jest mniejsza, niż do pozostałych. Operację tę można także wykonać stosując rotujący element strukturalny z rys. 5.37.

<b>x</b>	<b>1</b>	<b>x</b>
<b>x</b>	<b>0</b>	<b>x</b>
<b>0</b>	<b>0</b>	<b>0</b>

<b>0</b>	<b>x</b>	<b>x</b>
<b>0</b>	<b>0</b>	<b>1</b>
<b>0</b>	<b>x</b>	<b>x</b>

<b>0</b>	<b>0</b>	<b>0</b>
<b>x</b>	<b>0</b>	<b>x</b>
<b>x</b>	<b>1</b>	<b>x</b>

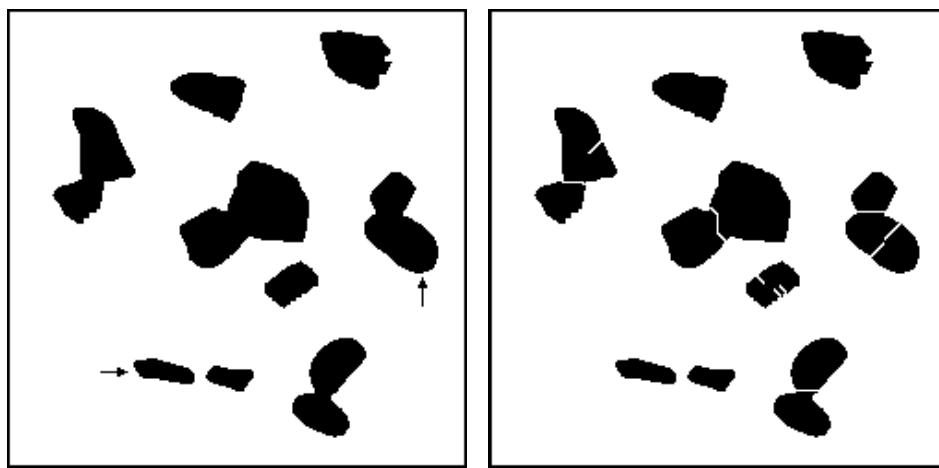
<b>x</b>	<b>x</b>	<b>0</b>
<b>1</b>	<b>0</b>	<b>0</b>
<b>x</b>	<b>x</b>	<b>0</b>

Rys. 5.37. Rotujący element strukturalny.

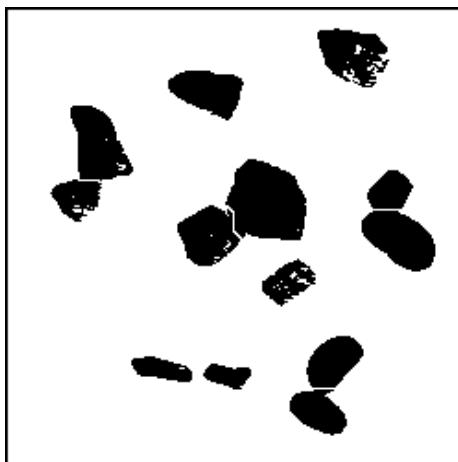
Jeżeli punkt centralny i otoczenie jednego z elementów strukturalnych (rys. 5.37) zgadza się z lokalną konfiguracją punktów obrazu, to nowa wartość punktu centralnego obrazu przyjmuje wartość 1. W przeciwnym przypadku wartość punktu centralnego obrazu nie ulega zmianie. Efekt działania tego przekształcenia obejrzeć można na rysunku 5.38. Generalnie jest ono przydatne i użyteczne - chociaż nie można nie zauważać pewnych artefaktów wytwarzanych przez operację SKIZ. Przekształcenie to posiada bowiem pewną ogólnie znaną wadę. Powstałe w jego wyniku figury mają bardzo nieregularny brzeg z licznymi wąskimi i głębokimi wklęściami. Narzucającym się rozwiążaniem jest zastosowanie następującej sekwencji przekształceń:

1. Negatyw
2. Obcięcie gałęzi
3. Negatyw

Wynik działania takiej korekty przedstawia rysunek 3.39.



Rys. 5.38. Efekt działania operacji SKIZ. Po lewej obraz przed operacją, po prawej - po operacji.

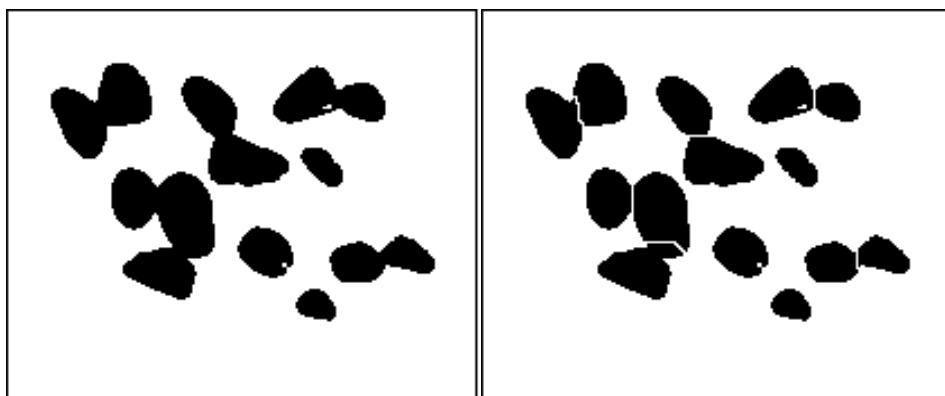


Rys. 5.39. Efekt działania operacji SKIZ po dodatkowej korekcie.

Opisane wyżej przekształcenie nie przekształca jednak pojedynczych punktów. Dlatego pierwszym krokiem w przekształceniu SKIZ jest jednostkowe pogrubienie elementem strukturalnym (przy każdej jego orientacji):

$$\begin{bmatrix} 0 & 0 & X \\ 0 & X & 1 \\ 0 & 0 & X \end{bmatrix}$$

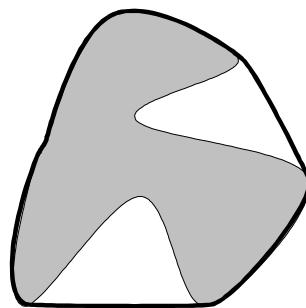
Przekształcenie SKIZ ma wiele praktycznych zastosowań. Oprócz znanego już efektu separacji obiektów stykających się na obrazach binarnych (por. rys. 5.40) stosowane jest ono także na przykład w symulacji procesów zachodzących w strukturach ziarnistych. Procesy te obejmują rozrost ziaren, krzepnięcie, pękanie, itp.



Rys. 5.40. Separacja obiektów stykających się za pomocą algorytmu SKIZ.

#### 5.4.4 Wypukłe otoczenie

Wypukłe otoczenie figury można zdefiniować jako najmniejszą figurę wypukłą zawierającą daną figurę. Dla figur wypukłych, ich wypukłe otoczenie jest zatem z nimi tożsame.



Rys. 5.41. Figura i jej wypukłe otoczenie.

Wyznaczenie wypukłego otoczenia może być realizowane jako pogrubiane kolejno dwoma następującymi elementami strukturalnymi:

$$\begin{bmatrix} 1 & X & X \\ 1 & X & 1 \\ 1 & 1 & X \end{bmatrix}, \text{ a następnie } \begin{bmatrix} X & 1 & X \\ 1 & X & 0 \\ X & 1 & X \end{bmatrix}.$$

#### 5.4.5 Przekształcenie trafi - nie trafi

Jednym z najprostszych przekształceń morfologicznych jest przekształcenie “trafi - nie trafi”<sup>36</sup>. Definicja tego przekształcenia jest następująca:

##### Definicja

Do każdego punktu analizowanego obrazu przykładowany jest punkt centralny danego elementu strukturalnego. Jeżeli lokalne otoczenie analizowanego punktu zgodne jest z elementem strukturalnym - odpowiedni punkt obrazu wynikowego uzyskuje wartość 1. W przeciwnym wypadku - wartość 0. Tak jak we wcześniej opisanych przekształceniach operacja ta jest przeprowadzana wielokrotnie, aż do braku zmian wprowadzanych przez operację.

Przekształcenie to jest na tyle elementarne, że można na jego podstawie zdefiniować wszystkie inne przekształcenia morfologiczne. I tak opisana wcześniej erozja może być zdefiniowana jako przekształcenie trafi - nie trafi z następującym elementem strukturalnym:

---

<sup>36</sup> ang. *hit or miss*

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Przekształcenie trafi - nie trafi jest wygodne przy detekcji pewnych charakterystycznych elementów obrazu. Ich detekcję należy przeprowadzić na obrazie poddanym wcześniej szkieletyzacji:

1. **Pojedyncze, odizolowane punkty.** Do detekcji pojedynczych, odizolowanych punktów w obrazie można wykorzystać przekształcenie trafi - nie trafi z następującym elementem strukturalnym:

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

1. **Punkty końcowe.** Do ich detekcji można wykorzystać następujący element strukturalny:

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ X & X & X \end{bmatrix}$$

1. **Punkty węzlowe.** Punkty węzlowe są punktami leżącymi na skrzyżowaniu linii obrazu. Można je wykryć przy użyciu następujących elementów strukturalnych:

$$\begin{bmatrix} X & 1 & X \\ X & 1 & X \\ 1 & X & 1 \end{bmatrix}, \text{ lub } \begin{bmatrix} 1 & X & X \\ X & 1 & X \\ 1 & X & 1 \end{bmatrix}$$

## 5.5 Alfabet Golay'a

Jak dotąd nie opracowano usystematyzowanej teorii doboru elementu strukturalnego realizującego założone przekształcenie morfologiczne. Pozostaje metoda prób i błędów. Pewną wskazówką jest fakt, że elementy strukturalne dokonujące detekcji pewnych charakterystycznych elementów obrazu są zazwyczaj obrazem tychże elementów.

Aby w pewien sposób usystematyzować nieskończoną mnogość elementów strukturalnych wprowadzono tzw. alfabet Golay'a. Wiąże on litery alfabetu rzymskiego z pewnymi klasami elementów strukturalnych.

W tabeli 5.1 ujęto symbole alfabetu Golay'a, odpowiadające im przykładowe elementy strukturalne dla siatki prostokątnej oraz ich zastosowania w przekształceniach ścieniania, pogrubiania i trafi - nie trafi. Zastosowane w niej oznaczenie  $B^C$  oznacza element strukturalny dopełniający do B.

Tabela 5.1. Elementy alfabetu Golay'a

Symbol	Element strukturalny	Ścienianie	Pogrubianie	Trafi - nie trafi
<b>C</b>	$\begin{bmatrix} 1 & X & X \\ 1 & 0 & X \\ 1 & 1 & X \end{bmatrix} i^{37} \begin{bmatrix} X & 1 & X \\ 1 & 0 & X \\ X & 1 & X \end{bmatrix}$		wypukłe otoczenie	
<b>D</b>	$\begin{bmatrix} 0 & X & X \\ 0 & 1 & 1 \\ 0 & 0 & X \end{bmatrix} i \begin{bmatrix} X & 0 & X \\ 0 & 1 & 1 \\ X & 0 & X \end{bmatrix}$	marker lub centroid	przybliżenie wypukłego otoczenia ( $D^C$ )	
<b>E</b>	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & X & X \end{bmatrix} i \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ X & 0 & X \end{bmatrix}$	obcinanie gałęzi	SKIZ ( $L^C$ , a następnie $E^C$ )	detekcja punktów końcowych
<b>F</b>	$\begin{bmatrix} X & 1 & X \\ X & 1 & X \\ 1 & X & 1 \end{bmatrix}, \begin{bmatrix} 1 & X & X \\ X & 1 & X \\ 1 & X & 1 \end{bmatrix}, \dots$			detekcja punktów potrójnych
<b>H</b>	$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	detekcja brzegu		erozja
<b>I</b>	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$			detekcja odizolowanych punktów
<b>L</b>	$\begin{bmatrix} 1 & 1 & 1 \\ X & 1 & X \\ 0 & 0 & 0 \end{bmatrix} i \begin{bmatrix} 1 & X & 1 \\ X & 1 & X \\ 0 & 0 & 0 \end{bmatrix}$	szkielet	SKIZ ( $L^C$ , a następnie $E^C$ )	
<b>M</b>	$\begin{bmatrix} 1 & 1 & X \\ 1 & 1 & 0 \\ 1 & 1 & X \end{bmatrix}$	szkielet	pogrubianie izolowanych punktów ( $M^C$ )	
<b>R</b>	$\begin{bmatrix} X & X & X \\ X & 1 & 0 \\ X & X & X \end{bmatrix}$	erozja liniowa	dylatacja liniowa ( $R^C$ )	

<sup>37</sup> "i" oznacza wykonywanie przekształcenia morfologicznego przy wykorzystaniu naprzemiennie podanych elementów strukturalnych.

## 5.6 Złożone przekształcenia morfologiczne

Opisane w tym i następnych punktach przekształcenia morfologiczne są złożeniem opisanych wcześniej metod. Niektóre z tych złożen mają tak duże znaczenie w praktyce, że nadano im własne nazwy.

### 5.6.1 Rekonstrukcja

Rekonstrukcja jest pewnym przekształceniem pomocniczym, często wykorzystywanym w innych, bardziej złożonych przekształceniach. Obrazem wyjściowym dla rekonstrukcji jest tak zwany obraz markerów (znaczników) będący pewnym podzbiorem obrazu wyjściowego całego, złożonego przekształcenia. Rekonstrukcja polega na cyklicznym dokonywaniu dylatacji obrazu i wyznaczeniu części wspólnej z obrazu uzyskanego po dylatacji i obrazu wyjściowego całego przekształcenia (wykonuje się operację logiczną AND z obrazem wyjściowym a więc usuwa się te fragmenty, które po dylatacji „wyszły” poza odtwarzaną figurę). Cykl ten powtarza się, aż do uzyskania zbieżności, to znaczy braku zmian w obrazie. Praktyczne działanie rekonstrukcji przedstawione zostanie na przykładzie wykorzystującego ją przekształcenia czyszczenia brzegu.

### 5.6.2 Czyszczenie brzegu

Czyszczenie brzegu ma na celu wyeliminowanie z obrazu wszystkich obszarów przecinających brzeg obrazu. Przekształcenie to często poprzedza dokonanie wnikliwszej analizy obrazu. Jest ono przydatne np. przed wykonaniem pomiarów w przypadku, gdy chcemy oceniać jedynie całkowicie widoczne obiekty. Jest to ważne i potrzebne właściwie we wszystkich dających się rozważać zastosowaniach techniki *computer vision*. Przykładowo jeśli zadanie polega na ocenie parametrów (na przykład powierzchni i obwodu) widocznych na obrazie obiektów - to obiekty widoczne nie całkowicie nie mogą być brane pod uwagę, bo algorytm wyznaczający ich powierzchnię lub obwód z pewnością dostarczy wyników błędnych, co może znacząco wpływać na wyniki dalszych analiz i na wnioski z nich wypływające (na przykład, jeśli dalsze wnioskowanie uwzględnia wartość średniej powierzchni obiektu - to będzie ona zaniedzona). Podobnie w przypadku, kiedy zadaniem systemu wizyjnego jest podanie informacji wspomagającej proces sterowania ramienia robota, dopuszczenie do analizy obiektów znajdujących się częściowo poza polem widzenia kamery może prowadzić do niepotrzebnych i niebezpiecznych pomyłek - na przykład planując trajektorię robota tak, by jego chwytek pochwycił widziany przez kamerę obiekt możemy - przy niecałkowitej widoczności - źle ocenić kształt obiektu (w rezultacie robot użyje - na przykład do montażu - nieodpowiedniej części) albo możemy źle wyznaczyć środek ciężkości obiektu (robot uchwyci obiekt w niewłaściwym miejscu, w wyniku czego obiekt wypadnie z szczeć chwytaka i uszkodzi się sam, albo spowoduje uszkodzenia innych obiektów i maszyn roboczych).

Jak wspomniano wcześniej w przekształceniu tym wykorzystuje się rekonstrukcję. Obrazem markerów dla rekonstrukcji jest część wspólna obrazu wyjściowego i jego brzegu. Operacja przebiega w trzech etapach:

- tworzenie markerów - wspólnej części obrazu i jego brzegu
- rekonstrukcja obiektów przeciętych przez brzeg obrazu (por. 5.6.1)
- generacja różnicy obrazu wejściowego i obrazu z obiektem po rekonstrukcji

Wynikiem przekształcenia jest różnica obrazu wyjściowego i wyniku rekonstrukcji. Na rysunku 5.42 przedstawiono przykład działania algorytmu czyszczenia brzegu.



Rys. 5.42 Przykład działania algorytmu czyszczenia brzegu.

### 5.6.3 Zalewanie otworów

Często spotykanym problemem przy analizie obrazu jest wypełnienie zamkniętych otworów w wyróżnionych obszarach. Może to być potrzebne z trzech powodów:

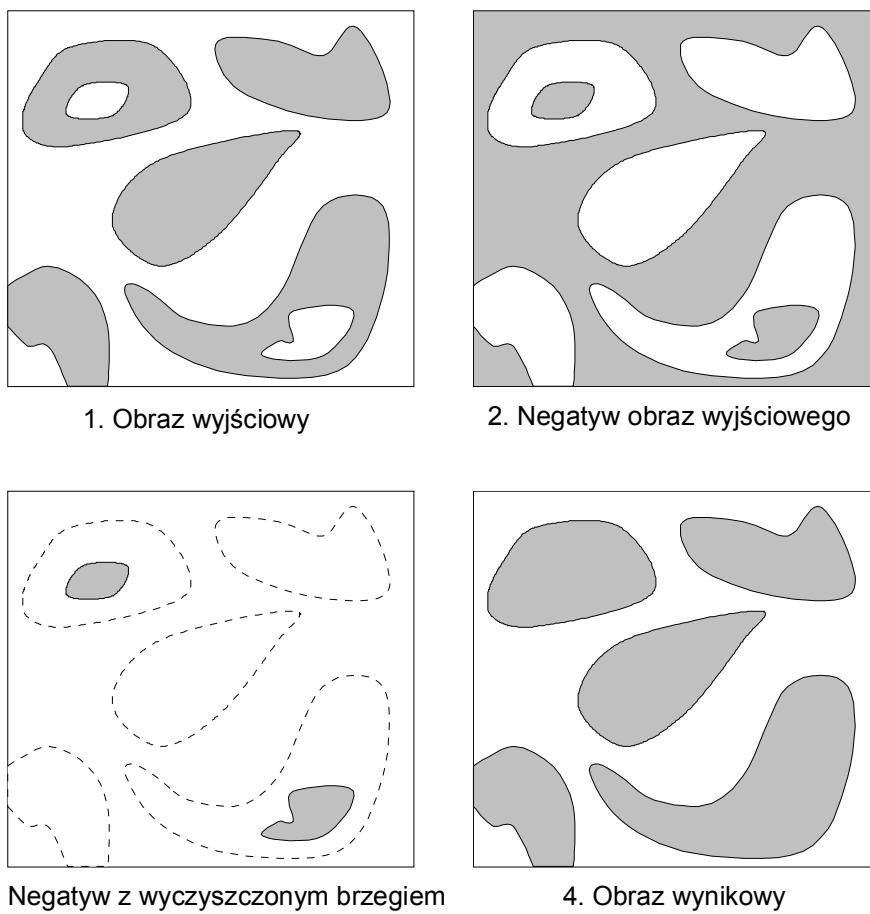
- bywa niekiedy, że otwory na obrazie nie odpowiadają rzeczywistym otworom w obiekcie, lecz powstały sztucznie na przykład jako skutek odblasku światła reflektorów oświetlających scenę, zatem przed oceną obiektu (na przykład w celu wyznaczenia położenia jego środka ciężkości) trzeba te „otwory” zlikwidować;
- potrzebne jest wyznaczenie parametrów obiektu bez uwzględniania otworów (na przykład trzeba wyznaczyć współczynniki kształtu (por. rozdz. 6) zewnętrznego konturu obiektu bez uwzględniania otworów;
- trzeba skupić uwagę wyłącznie na otworach (na przykład w celu policzenia, czy jest ich właściwa liczba i czy są poprawnie rozmieszczone - w zadaniach wykorzystania systemu wizyjnego jako elementu śródoperacyjnej kontroli jakości wyrobów). W takim przypadku właściwa droga postępowania może polegać na załaniu otworów (bo to można zrobić stosunkowo prosto podanym niżej

algorytmem), a potem na wyznaczeniu obrazu różnicowego - obrazu oryginalnego i obrazu z zalonymi otworami. Na uzyskanym obrazie wynikowym widoczne będą tylko otwory (traktowane w tym przypadku jako obiekty).

Algorytm realizujący funkcję zalewania otworów może sprowadzać się do trzech kroków:

1. Wyznaczenie negatywu z obrazu wyjściowego.
2. Wyczyszczenie brzegu uzyskanego negatywu (w wyniku tego pozostają na obrazie same otwory).
3. Wyznaczenie sumy logicznej (operacji OR) obrazu wyjściowego i wyniku czyszczenia brzegu.

Na rysunku 5.43. przedstawiono przykład działania algorytmu zalewania otworów.



Rys. 5.43. Przykład działania algorytmu zalewania otworów.

Zagadnienie wypełniania otworów pojawia się z całą ostrością przy stosowaniu techniki komputerowego przetwarzania obrazów do automatycznego rozpoznawania odcisków palców. Biały obiekt (dziura) wewnątrz śladu listewki skórnej to jeden z najbardziej niepożądanych efektów ubocznych powstających w procesie binaryzacji obrazu odcisku palca. Pozostawienie w obrazie binarnym dziur powoduje, że procedura szkieletyzacji pozostawi w miejscu dziury pętlę, wprowadzając tym samym nieprawdziwą cechę charakterystyczną odcisku palca. Tego typu postępowanie może doprowadzić do całkowicie błędnej interpretacji odcisku, silnie uzależniając wyniki od jakości wprowadzanych obrazów daktyloskopijnych. W celu wyeliminowania ujemnego wpływu dziur na późniejszy proces szkieletyzacji zbudowano algorytm ich wypełniania.

Procedura wypełniania otworów w tym przypadku, jest to rekurencyjna procedura znajdująca ograniczone czarnymi liniami białe obiekty. Parametrem istotnym tej procedury jest *rozmiar pola powierzchni* do jakiej białe obiekty należy traktować jako zakłócenie i wypełniać kolorem czarnym. To już oczywiście zależy od konkretnego zastosowania. Na przykład w zadaniu automatycznego rozpoznawania odcisków palców badania poszczególnych obiektów obrazu daktyloskopijnego, w tym także dziur dowiodły, że maksymalne pole powierzchni jakie może być wypełnione nie powinno przekraczać 16 punktów. Jest to spowodowane faktem, że najmniejszą spotkaną minucją zamkniętą (oczko, delta) w daktyloskopijnym obrazie binarnym była minuca o powierzchni 17 punktów (dotyczy białego obiektu - wnętrza).

Kryterium powierzchniowe wypełniania dziur nie jest wystarczające. Dość łatwo można sobie wyobrazić sytuację, w której pojedyncza listewka skóra o średniej grubości np. 7 pikseli posiada dziurę optyczną o powierzchni przekraczającej 17 punktów i kształcie cienkiej, długiej linii. Oczywiście taki obiekt jest zakłóceniem i należy go wypełnić. Tak więc wypełnieniu kolorem czarnym podlegać powinny te dziury optyczne, których powierzchnie nie przekraczają 17 pikseli, z wyjątkiem dziur w kształcie linii o maksymalnej grubości miejscowej nie większej niż 4 piksele (minimalna grubość listewki skórnej wynosi 6 punktów).

#### 5.6.4 Funkcja odległości

Funkcja odległości jest przekształceniem morfologicznym zdefiniowanym jedynie dla obrazów binarnych. Jest ona bardzo pomocna w pewnych bardziej złożonych przekształceniach. Funkcja odległości zdefiniowana jest w następujący sposób:

##### **Definicja**

Do każdego punktu analizowanego obrazu o wartości 1, odpowiadający punkt w obrazie wynikowym przyjmuje wartość równą minimalnej odległości tego punktu od brzegu figury. Pozostałe punkty uzyskują wartość 0.

W praktyce funkcja odległości może być realizowana jako suma kolejnych erozji obrazu wejściowego.

### 5.6.5 Erozja warunkowa

Po wykonaniu odpowiednio dużej liczby kroków zwykła erozja doprowadza do usunięcia wszystkich cząstek. Erozja warunkowa natomiast eroduje obiekty w taki sposób, aby z każdego obiektu pozostał obraz znacznika (tj. punkt lub cząstka, która w następnym kroku erozji znika). Tak więc warunkiem w erozji warunkowej jest pozostawienie z każdego obiektu pewnego minimalnego zbioru punktów. Operacja ta jest bardzo ważnym krokiem przy rozdzielaniu sklejonych cząstek. Obraz powstały w wyniku erozji warunkowej stanowi zbiór znaczników, wykorzystywanych do rekonstrukcji granic między cząstkami. Tym samym erozja warunkowa pozwala na łatwe wyznaczenie liczby posklejanych cząstek.

Algorytm erozji warunkowej można wykonać posługując się operacją erozji i rekonstrukcji. W pierwszym etapie na obrazie wejściowym wykonuje się operację erozji a następnie rekonstrukcji. Otrzymany obraz po rekonstrukcji jest odejmowany logicznie od obrazu wejściowego. Utworzona różnica obrazów przechowuje w ten sposób obiekty, które nie zostały zrekonstruowane ponieważ znikły po pierwszej erozji. W następnym etapie obraz po pierwszej erozji staje się obrazem wejściowym i poddaje się go kolejnej erozji i rekonstrukcji. Otrzymany obraz po rekonstrukcji jest odejmowany logicznie od obrazu po pierwszej erozji. Utworzona różnica obrazów jest następnie dodawana logicznie do różnicy obrazów z poprzedniego etapu. W kolejnym etapie obraz po drugiej erozji warunkowej staje się obrazem wejściowym i poddaje się go trzeciej z kolei erozji i rekonstrukcji. Otrzymany obraz po rekonstrukcji jest odejmowany logicznie od obrazu po drugiej erozji. Utworzona różnica obrazów jest następnie dodawana logicznie do różnicy obrazów z poprzedniego etapu. Kolejne etapy algorytmu są wykonywane aż do momentu, gdy kolejna operacja erozji doprowadzi do zniknięcia wszystkich obiektów. Różnice obrazów sumowane z poszczególnych etapów w efekcie dają obraz znaczników.

Erozja warunkowa jest więc pewną odmianą normalnej erozji. Przeprowadzana jest ona dla każdego obszaru aż do momentu, gdy kolejny krok usunąłby zupełnie pozostałość po obszarze. Pozostawia ona zatem pewien minimalny zbiór punktów z każdej figury. Dla przykładu erozja warunkowa pozostawia z prostokąta jedynie odcinek, a z koła - punkt.

## 6 Analiza obrazu

### 6.1 Uwagi wstępne

Analiza obrazu jest procesem polegającym na wyodrębnieniu z całkowitej informacji docierającej do obserwatora bądź detektora tej części, która jest istotna z punktu widzenia użytkownika lub procesu. Z reguły towarzyszy temu bardzo radykalna redukcja ilości informacji. Z kilkuset tysięcy (lub nawet kilku milionów) bajtów reprezentujących w typowych warunkach każdy obraz pozostaje zaledwie kilkadziesiąt lub kilkaset bajtów reprezentujących wartości wydobytych w trakcie analizy parametrów obiektów składających się na obraz.

Analiza obrazu może być prowadzona przy wykorzystaniu narzędzi programowych i sprzętowych. Narzędzia programowe wyposażone są z reguły w znacznie bogatszy zestaw przekształceń, są więc bardziej uniwersalne. Wadą ich jest stosunkowo długi czas obróbki obrazu, w praktyce mogący uniemożliwić pracę w czasie rzeczywistym. Wady tej pozbawione są specjalizowane procesory realizujące sprzętowo pewne przekształcenie lub ich grupę. Są one jednak znacznie mniej uniwersalne od narzędzi programowych. Procesory takie mogą pracować w systemach czasu rzeczywistego. Jednak nawet w tych systemach specjalizowane procesory służące do analizy obrazu występują znacznie rzadziej, niż omawiane wcześniej specjalizowane procesory przystosowane do przetwarzania obrazu (a zwłaszcza do jego filtracji - patrz rozdz. 3).

Niżej omówione zostaną podstawowe elementy procesu komputerowej analizy obrazów. Omówienie to będzie z konieczności fragmentaryczne i niepełne, ponieważ w typowych warunkach analiza obrazu (w odróżnieniu od jego filtracji) jest bardzo ściśle związana z konkretnym celem, dla którego zbudowano rozważany system wizyjny. Dlatego w tej książce, z założenia operującej informacjami ogólnymi, nie będzie możliwe przedstawienie metod analizy stosujących się specyficznie do obrazów - przykładowo - ultrasonograficznych. Ogólnych metod analizy jest natomiast relatywnie (w stosunku do liczby technik przetwarzania obrazu) raczej niewiele.

### 6.2 Techniki segmentacji

#### 6.2.1 Podział technik segmentacji

Czynnością spinającą poziom wstępnego przetwarzania obrazu z programami analizy poszczególnych obiektów jest **proces segmentacji obrazu**. Polega on na podziale obrazu (uprzednio przefiltrowanego) na fragmenty odpowiadające po-

szczególnym widocznym na obrazie obiektem. Towarzyszy temu zwykle opisana dalej **indeksacja obiektów** (*labelling*), czyli przypisanie do wszystkich pikseli tych obiektów identyfikatorów (etykiet), wskazujących, do którego obiektu który piksel można przypisać.

Segmentacja jest więc techniką przetwarzania obrazu, umożliwiającą wydzielenie obszarów obrazu spełniających pewne kryteria jednorodności. Takimi kryteriami mogą być np. kolor obszaru, poziom jasności, faktura. Obszary uzyskane przez zastosowanie takich czy innych technik segmentacji bardzo często mają swoje odpowiedniki (płaszczyzny, powierzchnie) w rzeczywistej scenie, której obraz poddano obróbce. Najczęściej segmentacja polega bowiem na oddzieleniu poszczególnych obiektów wchodzących w skład obrazu i na wyodrębnieniu ich od tła, na którym występują. Dzięki tej własności segmentacja jest bardzo istotnym elementem procesu **analizy logicznej scen**. Wykorzystuje się ją między innymi jako etap przygotowawczy do rozpoznawania obiektów, określania ich relacji przestrzennych, detekcji cech obiektów i rozwiązywania problemu korespondencji zgodnych ze sobą punktów w stereowizji.

Ogólnie można wyróżnić dwie **techniki segmentacji**:

- segmentacja przez podział obszaru (*region - splitting*)
- segmentacja przez rozrost obszaru (*region - growing*)

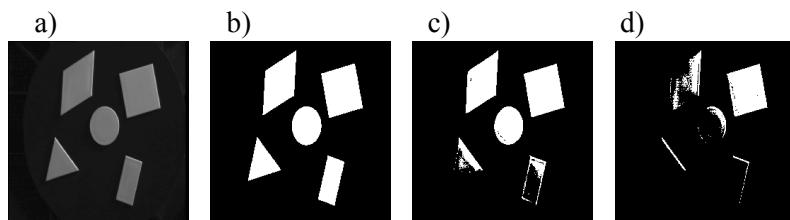
Segmentacja przez podział obszaru, zwana też iteracyjną (rekursywną), polega na stopniowym podziale dużych obszarów obrazu na mniejsze, w których piksele mają odpowiednią własność (kolor, jasność), znacznie różniącą się od własności pikseli w innych obszarach. W segmentacji przez rozrost obszaru piksele są testowane na to, czy mają określony stopień podobieństwa i jeżeli spełniają warunki, są dołączane do obszaru. Z kolei obszary sprawdza się według uśrednionych własności i relacji przestrzennych w celu połączenia ich w większy obszar. Segmentacja obrazu dostarcza zawsze krawędzi zamkniętych (granice obszarów). Z tego też powodu rezultaty segmentacji chętnie są wykorzystywane przez procesy wizji wysokiego poziomu, dokonujące np. rozpoznawania obiektów. Ciągłość krawędzi jest z pewnością atutem segmentacji w porównaniu z detektorami krawędzi, które nie zapewniają ciągłości wykrytych krawędzi.

### 6.2.2 Segmentacja przez podział obszaru

W metodzie tej wartość jasności każdego elementu obrazu jest porównywana z wartością progową, po czym element jest przydzielony do jednej z dwóch kategorii: o wartości progowej przekroczonej lub nie przekroczonej. Wyboru wartości progowej dokonuje się przeważnie na podstawie histogramu. Przy praktycznym stosowaniu tej metody do segmentacji obrazu wyłaniają się dwa problemy. Pierwszy polega na wyborze progu dyskryminacji. W sytuacji, gdy krzywa rzeczywistego histogramu bywa wielogarbna, a wierzchołki, pomiędzy którymi należy prowadzić separacje, bywają trudne do wyróżnienia - powstaje

problem z ustaleniem progu. Drugi problem polega na tym, że mechaniczne stosowanie metody progowania bardzo łatwo może doprowadzić do pojawienia się całego szeregu „fałszywych obiektów” - izolowanych punktów mających wartość 1, podczas gdy dookoła są wyłącznie punkty tła. Tworzenia izolowanych pseudoobiektów można uniknąć poprzez tzw. „określenie obszaru jednolitego”. Odbywa się to poprzez kolejne dołączanie pikseli spełniających warunki progowania i będących sąsiadami jednego lub więcej pikseli należących już do obszaru. Bardziej uogólnioną procedurą tworzenia obszaru jest dołączanie pikseli, których atrybuty mieszą się w określonym przedziale.

Progowanie jest odpowiednią techniką dla scen zawierających obiekty o w miarę jednorodnych powierzchniach widzianych na kontrastowym tle. Metoda segmentacji oparta na progowaniu pozwala dokonywać segmentacji nawet złożonych scen. Na podstawie obrazu sceny, wyliczane są najpierw histogramy związane z wieloma atrybutami (odcienie, nasycenie, jasność i inne). Spośród powyższych histogramów jest wybierany ten, który posiada najbardziej wyrazisty wierzchołek. On też określa, według jakiego atrybutu i z jakim przedziałem progowania będzie przeprowadzana segmentacja. Proces ten jest powtarzany dla wszystkich wyselekcjonowanych już obszarów i dla pozostały części obrazu aż do momentu, gdy nie można już wyizolować kolejnego, nowego obszaru. Dzięki temu, że pewne różnice pomiędzy pikselami uwidaczniają się po wyizolowaniu ich z większego obszaru, osiągalna jest tą drogą dość szczegółowa segmentacja (rysunek 6.1).



Rys. 6.1. Poprawna i niepoprawna segmentacja obrazu: a) widok obiektów podlegających segmentacji, b) obraz po poprawnej segmentacji, c,d) obraz po nieprawidłowej segmentacji.

Przedstawiona technika zwana jest **segmentacją rekursywną**. Algorytm tej segmentacji jest kosztowny obliczeniowo. Udoskonaleniem może tu być zastosowanie zredukowanej rozdzielczości. Segmentację wstępную przeprowadza się na zredukowanej rozdzielczości, a następnie - segmentację dokładną przy pełnej rozdzielczości obrazu.

### 6.2.3 Segmentacja metodą wykrywania krawędzi

Przy tej metodzie są wyszukiwane krawędzie między obszarami. W tym celu jest stosowany operator gradientu, a następnie na gradiencie jest wykonywana opera-

cja progowania. W kolejnym etapie elementy obrazu, które zostały zidentyfikowane jako krawędzie, muszą być połączone dla uformowania krzywej zamkniętej otaczającej obszary. Techniką powszechnie stosowaną jest rozpatrywanie różnic między dwoma grupami elementów, podobnie jak w filtrach liniowych górnoprzepustowych. Aby można było uwzględnić różnice w ukierunkowaniu krawędzi, trzeba zastosować więcej niż jeden filtr. Ponieważ filtry górnoprzepustowe mają tendencję do uwydatnienia zakłóceń, wartość tego typu metod w przypadku obrazów z zakłóceniami jest ograniczona.

Metody wykrywania krawędzi nie nadają się do tych samych obrazów, do których można stosować progowanie. Proste detektory krawędzi są odpowiednie jedynie dla obrazów z dużym kontrastem i niskim poziomem zakłóceń o wysokich częstotliwościach. Jednak nie dla wszystkich tego typu obrazów można dokonać właściwej segmentacji metodą progowania. Zakłócenia o małych częstotliwościach, które nie oddziałują przy wykrywaniu krawędzi, mogą spowodować, że progowanie nie da wyników.

Obraz, dla którego występuje gradient natężenia oświetlenia, jest najprostszym, typowym przykładem. Wzrok ludzki jest wrażliwy na kontrast i może nie zauważać płynnych zmian natężenia oświetlenia. Jednak to, co wygląda jak „ciemne” w jednym końcu obrazu ma tę samą wartość jak to, co wygląda jak „jasne” w drugim końcu. Stosując binaryzację ze zmiennym progiem można określić próg oddzielający miejsca „ciemne” od „jasnych”. Z drugiej strony, prosty detektor krawędzi może także z powodzeniem wykonać to zadanie. Powyższe wnioski mogą być słuszne również w przypadkach, gdy zakłócenia o niskich częstotliwościach nie są tak silne, ponieważ łatwiejszy może być wybór progu decydującego o istnieniu krawędzi niż progu do segmentacji całego obrazu.

#### 6.2.4 Segmentacja przez rozrost obszaru

Metody wykrywania krawędzi i progowania zajmują się różnicami między wartościami elementów obrazu, natomiast w metodzie rozrostu obszaru poszukuje się grup elementów o zbliżonej jasności. Najprostsza postać tej metody to rozpoczęcie od jednego elementu i sprawdzanie czy elementy sąsiednie mają podobną jasność. Jeśli tak, są one grupowane w obszar.

W ten sposób powstają obszary, które rozrastają się z pojedynczych elementów obrazu. W bardziej złożonej postaci tej metody działanie rozpoczyna się nie od pojedynczych elementów, ale od podziału obrazu na zbiór małych obszarów. Następnie dla każdego obszaru stosuje się test jednolitości i w razie negatywnego wyniku obszar jest ponownie dzielony na mniejsze elementy. Proces ten jest powtarzany tak długo, aż wszystkie obszary są jednolite. W ten sposób obszary są tworzone z mniejszych obszarów, a nie z elementów. Sposób realizacji metody zależy w dużym stopniu od struktur danych używanych do reprezentacji obrazu. Główną zaletą użycia małych obszarów zamiast elementów obrazu jest zmniejszenie wrażliwości na zakłócenia.

Jedno z kryteriów jednolitości jest oparte na porównaniu maksymalnej różnicy między wartością elementu obrazu  $L(m,n)$  i wartością średnią dla obszaru. Dla obszaru  $R$  o wielkości<sup>38</sup>  $\#R$  oblicza się średnią  $S$ :

$$S = \frac{1}{\#R} \sum_{m,n \in R} L(m,n)$$

a następnie sprawdza się warunek:

$$\max_{m,n \in R} |L(m,n) - S| < T$$

Obszar jest jednolity, gdy powyższy warunek jest spełniony dla ustalonego progu  $T$ . Wybór tego progu oprzeć można na znany fakcie: prawdopodobieństwo, że jasność elementu  $L(m,n)$  różni się od jego średniej  $S$  o więcej niż pewną wielkość  $x$  jest dane następującą całką:

$$I_n = \frac{2}{\sqrt{2\pi}\sigma_x} \int_0^\infty e^{-\frac{z^2}{2\sigma_x^2}} dz$$

gdzie  $\sigma$  jest odchyleniem standardowym zakłóceń. Ustalając próg  $T$  ustala się więc także prawdopodobieństwo tego, że do wydzielonego obszaru włączone będą obiekty nie należące do niego. Jeśli obszar obiektu jest jednolity, to wartość średnia  $S$  jest optymalnym oszacowaniem wartości jasności tego obszaru. W takim przypadku odchylenie wartości elementu od średniej  $S$  będzie zależało jedynie od zakłóceń.

Inną procedurą segmentacji przez rozrost jest procedura, która scalą najpierw regiony atomowe badając wartość średnią określonego **atrybutu** w stosunku do określonego zakresu progowania. Zastosowany w tym etapie próg  $T$  może być inny, mniej szczegółowo wyznaczany niż przy ekstrakcji regionów atomowych. Następne scalenia są oparte na względnych własnościach dwóch sąsiednich obszarów. Jeśli jeden obszar w dużym stopniu otacza inny, to są one scalane.

Wyniki segmentacji przez rozrost silnie zależą od wyboru progu  $T$ . O ile segmentacja taka daje dobre wyniki dla nieskomplikowanych scen, o tyle dla scen złożonych ma ona tendencje do tworzenia dużej liczby małych obszarów.

Jedyną z możliwości zredukowania wpływu wyboru progu  $T$  na rezultaty segmentacji jest przechowywanie całego drzewa (ogólniej grafu) obszarów otrzymanych na różnych poziomach rozrostu obszarów. Korzeniem tego drzewa jest cały obraz, liśmi zaś - obszary atomowe. Pośrednie poziomy zawierają obszary utworzone przez scalenie obszarów niższego poziomu. Mając takie drzewo procedura segmentacji może badać dodatkowe zależności pomiędzy obszarami, tworząc - zależnie od potrzeb - różne warianty scalenia obszarów niższego poziomu.

---

<sup>38</sup> Wielkość obszaru  $\#R$  można rozumieć na przykład jako liczbę pikseli należących do obszaru.

Drzewo opisujące relacje pomiędzy regionami na różnych poziomach zwane jest czasami **drzewem obrazu** (*picture tree*). Jeśli natomiast drzewo ma tę własność, że obszar poziomu wyższego powstaje z czterech obszarów niższego poziomu, jest określone jako *quad tree*.

### 6.2.5 Segmentacja oparta na statystyce

Dość specyficznym problemem jest segmentacja obrazów, dla których jednorodność obszaru nie jest mierzona jednorodnością takich atrybutów, jak kolor czy jasność. Obszary uważa się w tym przypadku za jednorodne, jeśli są wypełnione pewnym stałym wzorcem (teksturą). Typowym przykładem może być obraz przedstawiający próbę geologiczną. W próbce tej obserwowana skała zawiera ziarna różnych minerałów, z których każdy ma charakterystyczny dla siebie obraz tekstury. W tych przypadkach nie można zastosować segmentacji opartych na uprzednio opisanych technikach, gdyż pojedyncze piksele znacznie różniąc się od siebie, tworzą pomimo to jednorodny obszar. Dlatego też w tym przypadku określa się pewne statystyczne własności obszarów i przeprowadza segmentację o nie opartą.

### 6.2.6 Segmentacja obiektów stykających się

Dla celów praktycznej segmentacji obrazów zawierających wiele stykających się obiektów używa się zwykle bardziej złożonego algorytmu, składającego się z następujących etapów:

1. Usunięcie elementów przeciętych przez brzeg obrazu.
2. Wykonanie operacji zamknięcia na obrazie pierwotnym, a następnie - operacji wypełniania otworów (powstanie obrazu wejściowego).
3. Erozja warunkowa, w wyniku której powstaje obraz znaczników z każdego erodowanego obiektu.
4. Wykonanie jednej operacji dylatacji dla każdego zerodowanego obiektu.
5. Wykonanie operacji SKIZ aż do osiągnięcia stanu ustalonego, przy czym po każdej operacji dylatacji obraz jest mnożony logicznie z obrazem wejściowym.
6. Obcięcie gałęzi i przemnożenie logiczne obrazu po obcięciu z obrazem oryginalnym.

Do algorytmu tego dodać warto następujące uwagi uzupełniające:

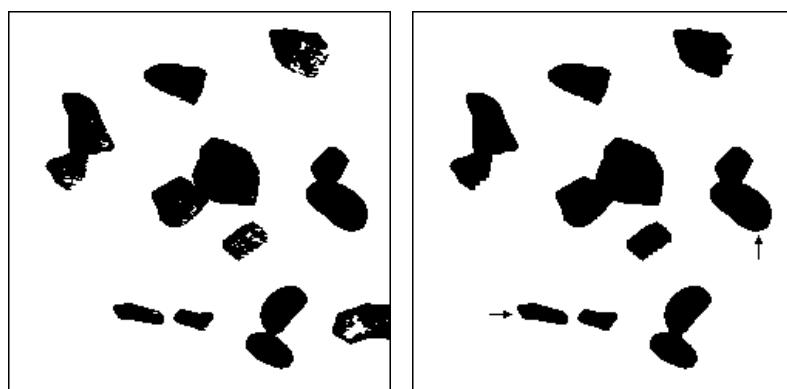
- Ad.1. Metoda czyszczenia brzegów obrazu działa w ten sposób, że najpierw dokonuje się indeksacji obrazu. Potem wystarczy już tylko sprawdzić, czy dany piksel stanowi krawędź obrazu czy nie. Jeśli tak, to usuwane są wszystkie piksele o indeksie takim samym jak indeks wykrytego piksela znajdującego się na krawędzi obrazu, czyli praktycznie usuwany jest cały obiekt przecięty przez brzeg obrazu.

- Ad.2. Operacja zamknięcia pozwala na wygładzenie postrzępionych brzegów obiektów oraz wypełnianie małych otworów, które mogą być przyczyną powstawania dodatkowych znaczników, a w efekcie końcowym - dodatkowych podziałów obiektów. Zamknięcie jest złożeniem operacji dylatacji i erozji i dokładniej zostało omówione w rozdziale 5. Operacja wypełnienia większych otworów złożona jest z czterech kroków. Najpierw wykonywana jest negacja obrazu. Wartość każdego piksela obiektu jest zamieniana z wartością 255 na zero, a wartość pikseli tła obrazu i dziur w obiektych jest zamieniana z wartością zero na 255 (tło obrazu to wszystkie te piksele obrazu, które nie należą do pikseli tworzących obiekty obrazu i dziury w tych obiektach). Potem następuje indeksacja obiektów, podczas której tło obrazu otrzymuje indeks jeden. Dziury natomiast otrzymują indeks większy od jednego. W następnym etapie wszystkie obiekty o indeksie większym od jednego, czyli dziury, otrzymują wartość zero. W ostatniej fazie dokonuje się powtórnej negacji obrazu.
- Ad.3. Erozja warunkowa stosowana w tym algorytmie charakteryzuje się tym, że dla każdego erodowanego obiektu jest pamiętana liczba erozji. Po każdej erozji elementy obrazu są na nowo indeksowane. W tablicy pomocniczej zapisywany jest identyfikator obiektu pod elementem tablicy wyznaczonym przez jego indeks. Potem sprawdzane są poszczególne wartości tej tablicy z zakresu od 1 do wartości maksymalnej indeksu obiektów obrazu. Jeśli któraś z wartości tablicy pomocniczej jest równa zero, to oznacza to, że obiekt o identyfikatorze równym indeksowi tablicy dla tej wartości nie istnieje, zniknął po ostatniej erozji. Ponieważ algorytm przechowuje kopię obrazu z poprzedniego kroku, to można w tym momencie rozpoznać, który element zniknął po ostatniej erozji i przepisać go do tablicy przechowującej obraz znaczników z indeksem równym liczbie erozji. Dzięki takiej realizacji w algorytmie są stosowane tylko dwie operacje: erozja i indeksacja. Jeżeli punkt centralny i otoczenie rotującego elementu strukturalnego zgadza się z lokalną konfiguracją punktów obrazu, to nowa wartość punktu centralnego obrazu przyjmuje wartość 1. W przeciwnym przypadku punkt centralny przyjmuje wartość 0. Każda z czterech postaci rotującego elementu strukturalnego dokonuje kolejno analizy obrazu, a wynik za każdym razem jest przepisywany do matrycy pomocniczej.
- Ad.4. Po zerodowaniu wszystkich obiektów powstaje obraz znaczników. W wyniku operacji erozji, może dojść do sytuacji, że z jednego obiektu (najczęściej o wydłużonym kształcie) mogą powstać dwa znaczniki i wówczas prowadzi to do dodatkowego podziału obiektu. W tej sytuacji wystarczy (choć nie we wszystkich przypadkach) wykonać jedną operację dylatację. Dylatacja doprowadzi do połączenia tych dwóch znaczników, zapobiegając w ten sposób przed dodatkowym podziałem obiektu.

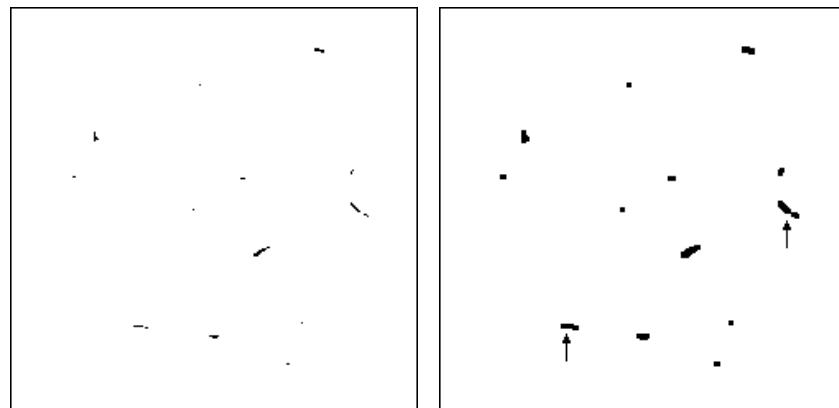
Ad.5. Następną operacją algorytmu segmentacji jest SKIZ. Operacja ta została także opisana w rozdziale 5. W pierwszej fazie wykonywana jest dylatacja bez dotykania o liczbie kroków równej maksymalnej liczbie erozji (tj. liczbie, która charakteryzuje ostatni zerodowany obiekt). Najpierw dylatacji bez dotykania poddawany jest obiekt o największej liczbie erozji, a później, w zależności od kolejnych liczb wykonanych poprzednio erozji, operacja jest wykonywana dla następnych obiektów. Tak więc każdy z elementów poddawanych dylatacji jest rekonstruowany w odpowiednich proporcjach w stosunku do wszystkich pozostałych obiektów. W drugiej fazie wszystkie obiekty podlegają dylatacji bez dotykania równocześnie, aż do osiągnięcia stanu ustalonego. Po każdej dylatacji wykonywana jest operacja logicznego mnożenia z obrazem oryginalnym, który wcześniej poddany został operacji zamknięcia, a później - wypełnienia otworów. Mnożenie przez tak uformowany obraz wejściowy ma na celu wyznaczenie dokładnych granic pomiędzy obiektami w obszarze, w którym elementy mogą jeszcze powiększać swoje rozmiary.

Ad.6. Ostatnimi operacjami algorytmu są: obcięcie gałęzi i przemnożenie uzyskanego obrazu przez obraz oryginalny. W wyniku wielokrotnej operacji dylatacji bez dotykania powstają gałęzie, które można zlikwidować techniką opisaną w rozdziale 5. Jeżeli punkt centralny i otoczenie rotującego elementu strukturalnego zgadza się z lokalną konfiguracją punktów obrazu, to nowa wartość punktu centralnego obrazu przyjmuje wartość 0. W przeciwnym przypadku wartość punktu centralnego nie ulega zmianie. W celu wyszukania wszystkich elementów końcowych element strukturalny musi być obracany tak, aby przyjął wszystkie możliwe dla danej siatki pozycje. Obcinanie gałęzi to w rzeczywistości powtarzany cykl wyszukiwania i usuwania punktów końcowych.

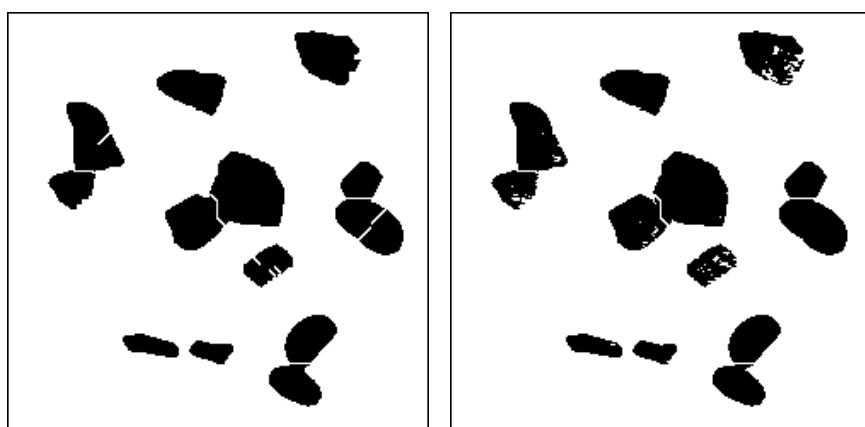
Przebieg opisanej wyżej segmentacji pokazuje seria rysunków 6.2 - 6.4 podanych niżej:



Rys. 6.2. Etap usuwania obiektów stykających się z krawędzią obrazu.

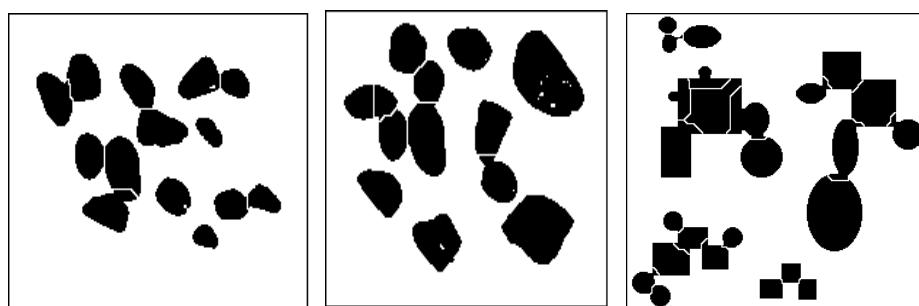


Rys. 6.3. Etap erozji warunkowej i wydobywania markerów.



Rys. 6.4. Etap po operacji SKIZ (przed i po usunięciu „fiordów”).

Proces segmentacji nie zawsze daje nienagannie poprawne wyniki. Przykłady wadliwej segmentacji kilku obrazów przedstawiono na rys. 6.5.

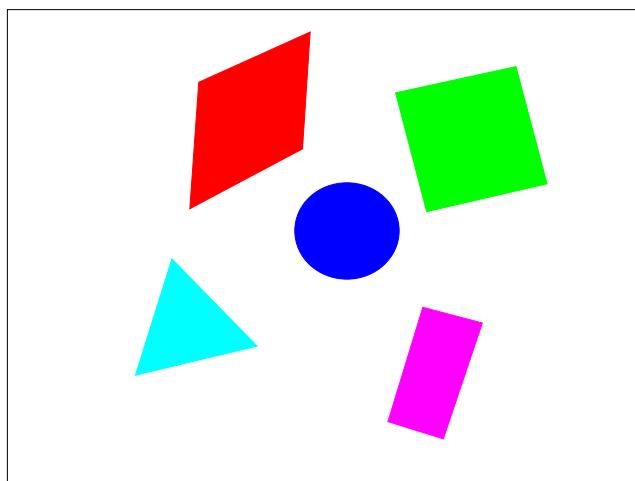


Rys. 6.5. Przykłady wadliwej segmentacji obrazów.

### 6.3 Technika indeksacji obrazu

Celem zabiegów opisanych w tym podrozdziale jest **indeksacja obiektów** (*labeling*), czyli przypisanie do wszystkich pikseli tych obiektów identyfikatorów (etykiet), wskazujących, do którego obiektu który piksel można przypisać. Najczęściej wykonuje się to poprzez znaczenie tych pikseli specjalnymi charakterystycznymi wartościami, co odpowiada wypełnianiu obszarów odpowiadających rozróżnianym obiektem zróżnicowanymi, sztucznie wprowadzonymi „poziomami szarości”. Nie prowadzi to zwykle do nieporozumień, ponieważ oryginalny obraz poddawany indeksacji jest już najczęściej (w wyniku wcześniej przeprowadzonej segmentacji) doprowadzony do postaci binarnej.

Dzięki indeksacji możliwe są wszystkie inne pomiary na indywidualnych obiektach, gdyż komputer ma wyznaczone „granice” dla każdego analizowanego obiektu. Obraz powstający w wyniku indeksacji pokazano na rysunku 6.6.



Rys. 6.6. Wynik etykietowania obiektów na prostym obrazie.

Proces indeksacji może być wykonywany na kilka niezależnych sposobów, przy czym najpopularniejsza technika polega na przeglądaniu zbinaryzowanego obrazu linia po linii aż do napotkania punktu o wartości 1 (czyli punktu należącego do obiektu). Po napotkaniu takiego punktu (piksela), oznaczonego dalej jako **X**, nadaje mu się etykietę (wartość koloru) analizując wartości pikseli A, B, C, D, stanowiących tę część jego otoczenia, która była już wcześniej analizowana (rys. 6.7).

A	B	C
D	X	

Rys. 6.7. Zasada etykietowania punktów na podstawie ich otoczenia.

Dalszy ciąg postępowania zależy od tego, jakie wartości mają piksele A, B, C, D. Jeśli

$$A = B = C = D = 0$$

wówczas przyjmuje się (rys. 6.8)

$$X = L + 1$$

gdzie L jest maksymalnym numerem (indeksem) użyтыm wcześniej do etykietowania wcześniej napotkanych obiektów (na początku procesu indeksacji zwykle przyjmuje się  $L=0$ , dzięki czemu pierwszy napotkany obiekt otrzymuje - zgodnie z podaną wyżej regułą - indeks  $L=1$ , a kolejne następne uzyskują kolejne dalsze numery).

0	0	0
0	<b>L+1</b>	

Rys. 6.8. Pierwszy przypadek etykietowania punktów na podstawie ich otoczenia.

W przeciwnym przypadku, to znaczy gdy pewien z pikseli A, B, C, D ma już wartość różną od zera (czyli wartość będącą indeksem jakiegoś obiektu) trzeba rozróżnić dwie możliwe sytuacje. Pierwsza polega na tym, że wszystkie piksele mające niezerowe indeksy wskazują na ten sam obiekt L (rys. 6.9). Wówczas oczywiście nowy (aktualnie badany) piksel otrzymuje ten sam indeks L.

0	0	0
L	<b>L</b>	

L	L	L
L	<b>L</b>	

0	L	L
0	<b>L</b>	

0	0	L
0	<b>L</b>	

0	L	L
L	<b>L</b>	

L	0	L
0	<b>L</b>	

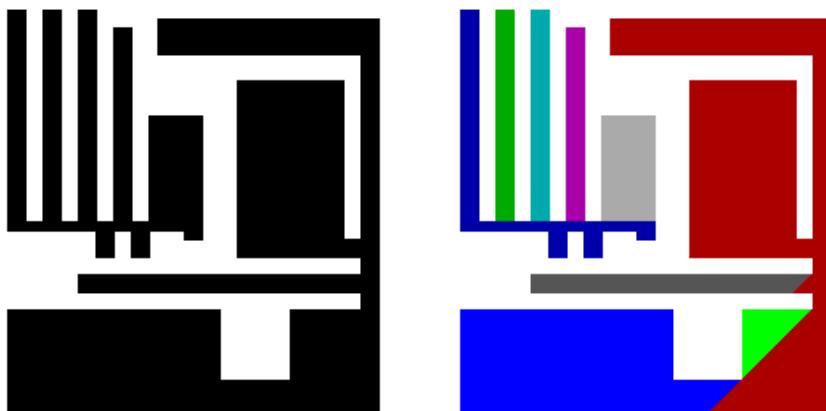
Rys. 6.9. Kilka przykładów zgodnego etykietowania punktów na podstawie ich otoczenia.

Bardziej skomplikowana sytuacja powstaje, gdy w otoczeniu badanego punktu X pojawią się punkty należące do dwóch różnych obiektów o numerach (indeksach) L1 i L2 (rys. 6.10). W takim przypadku przyjmowana jest zasada, że punktowi X przypisuje się *mniejszy* z wchodzących w grę indeksów (na rysunku 6.10 założono, że L1 < L2).

L1	0	L2
L1	<b>L1</b>	
L1	0	L2
0	<b>L1</b>	

Rys. 6.10. Przykłady kolizji podczas etykietowania punktów na podstawie ich otoczenia.

Przedstawiona zasada działania może prowadzić podczas etykietowania do powstania w obrębie jednego obiektu kilku obszarów o różnych etykietach. Ilustruje to rysunek 6.11, na którym pokazano wynik stosowania omówionego algorytmu w przypadku szczególnie „złośliwie” dobranych obiektów.



Rys. 6.11. Przykład etykietowania obrazów, w wyniku którego doszło do błędnego podziału obiektów.

W celu usunięcia powstałych błędów konieczne jest użycie w trakcie etykietowania dodatkowego mechanizmu, nazywanego tablicą klejów. Tablica taka ma tyle kolumn, ile maksymalnie etykiet chcemy wpisać na obrazie (najczęściej ze względów technicznych wybiera się 255) i dwa wiersze. Budowę tabeli przed rozpoczęciem procesu etykietowania obrazu pokazano na rysunku 6.12. W górnym wierszu tablicy podane są po prostu kolejne numery (nie będą się one więcej zmieniały, więc ich rolę mogą też pełnić odpowiednio przeskalowane adresy poszczególnych bajtów obszaru pamięci RAM, w którym ulokowano

tablicę). W dolnym wierszu początkowo wstawione są zera, co oznacza, że żaden obiekt nie został jeszcze zaetykietowany.

1	2	3	4	5	...	252	253	254	255
0	0	0	0	0	...	0	0	0	0

Rys. 6.12. Zawartość tablicy sklejeń przed znalezieniem jakiegokolwiek obiektu.

Po znalezieniu piksela **nowego** obiektu (por. rys. 6.8) następuje odnalezienie pierwszej (od lewej strony) kolumny tabeli sklejeń, w której dolnym wierszu znajduje się 0, a następnie znalezionemu pikselowi przypisuje się numer (etykietę) odpowiadającą zawartości górnego wiersza znalezionej kolumny, a równocześnie ten sam numer (etykietę) wpisuje się do dolnego wiersza. Ilustrują to rysunki 6.13 i 6.14.

1	2	3	4	5	...	252	253	254	255
1	0	0	0	0	...	0	0	0	0

Rys. 6.13. Zawartość tablicy sklejeń po znalezieniu pierwszego obiektu.

1	2	3	4	5	...	252	253	254	255
1	2	3	0	0	...	0	0	0	0

Rys. 6.14. Zawartość tablicy sklejeń po znalezieniu trzeciego obiektu.

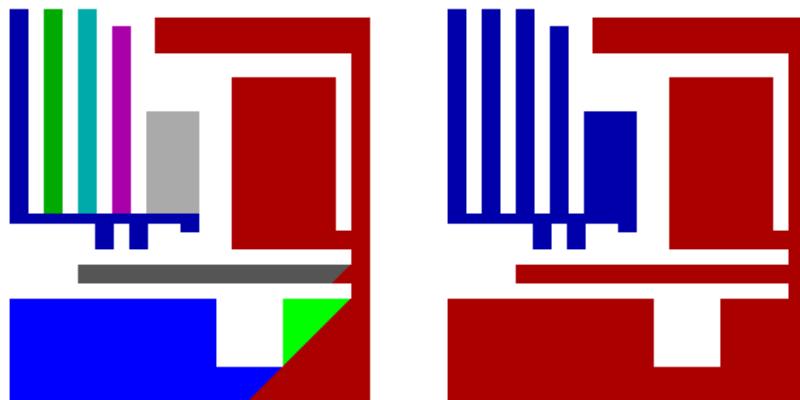
W przypadkach zgodnego etykietowania, opisanych na rysunku 6.9 w tablicy sklejeń nic się nie zmienia. Natomiast w przypadku wystąpienia kolizji podczas etykietowania (por. rys. 6.10) następuje skorygowanie kolumny w tablicy sklejeń odpowiadającej większemu z kolidujących indeksów (L2) i w dolnym wierszu tej kolumny zostaje wpisany indeks L1 zamiast L2 (patrz rys. 6.15).

1	2	3	4	5	...	252	253	254	255
1	2	1	4	0	...	0	0	0	0

Rys. 6.15. Zawartość tablicy sklejeń po wykryciu faktu sklejenia pierwszego i trzeciego obiektu.

Jeśli opisaną procedurę stosuje się konsekwentnie i systematycznie, to po przejściu całego obrazu i zaetykietowaniu wszystkich obiektów postanie tablica sklejeń, która może następnie zostać w prosty sposób użyta do przeindeksowania

błędnie zaindeksowanych części obiektów. W najprostszym przypadku tablicę sklejeń można wykorzystać jako podstawę do wykonania na obrazie operacji LUT (por. rozdz. 2.3). Wynik przeindeksowania dla obrazu testowego pokazano na rysunku 6.16. Widać, że po przeindeksowaniu pierwotne błędy indeksacji zostają w praktyce całkowicie wyeliminowane.



Rys. 6.16. Korekta indeksacji obiektu z rysunku 6.11 po zastosowaniu tablicy sklejeń.

## 6.4 Pomiary

### 6.4.1 Rola i znaczenie pomiarów parametrów obiektów na obrazach

Pomiary są często jednym z końcowych etapów komputerowej analizy obrazu. Analiza obrazu przebiega najczęściej w następujący sposób:

1. Wstępne przetwarzanie obrazu. Ma ono na celu wyeliminowanie z obrazu nieistotnych lub zakłócających elementów z punktu widzenia zamierzonych celów analizy, jak również uwypuklenie interesujących cech analizowanego obrazu. Przetwarzanie to prowadzi do określonych zmian w obrazie, a jego wynikiem jest również obraz. Przykłady takiego przetwarzania opisane zostały w poprzedniej części niniejszej pracy.
2. Dokonanie pomiarów poszukiwanych cech analizowanego obrazu. Pomiary pozwalają uzyskać wielkości ilościowe opisujące określone cechy analizowanego obrazu. Przeprowadzanie pomiarów będzie tematem niniejszego rozdziału.
3. Interpretacja otrzymanych wyników ilościowych i podjęcie na ich podstawie odpowiednich decyzji. Etap ten jest całkowicie zależny od zamierzonego, ogólnego celu przetwarzania obrazu. Interpretacja ta zależy od konkretnej dziedziny nauki lub techniki, w której stosowana jest analiza obrazu i wykracza całkowicie poza ramy niniejszej pracy.

Jak wspomniano powyżej, wynikiem pomiarów są pewne wielkości ilościowe, opisujące wybrane, charakterystyczne cechy badanego obrazu. Pozwalają one **zredukować** obraz do zbioru wielkości go opisujących, w pewnym, interesującym nas kontekście. Taki zbiór zajmuje wielokrotnie mniej pamięci, niż analizowany obraz, a na jego podstawie można bezpośrednio wyciągać wnioski, co do rozpatrywanego obrazu.

Generalnie, pomiarom podlegać mogą dwie kategorie wielkości:

- **Parametry lokalne**, które opisują pewien uśredniony element obrazu. Przykładem jest średnia powierzchnia lub średnica obiektu, średnia krzywizna brzegu, średni moment bezwładności, itp.
- **Parametry globalne**, opisujące cechy grup obiektów lub innych elementów obrazu. Są one zawsze wielkościami względnymi, na przykład ilość obiektów na jednostkę powierzchni długości. Przykładami takich parametrów mogą być: liczba obiektów na jednostkę objętości lub powierzchni, udział powierzchniowy wybranych elementów obrazu, długość linii na jednostkę pola powierzchni obrazu, itp.

W przypadku komputerowej analizy obrazu zarówno parametry lokalne, jak i globalne są wyznaczane pośrednio na podstawie pomiarów przeprowadzonych dla poszczególnych obrazów. Na przykład - na obrazie wyznacza się liczbę elementów a następnie, znając rzeczywistą wielkość obszaru widocznego na obrazie, oblicza się liczbę elementów przypadającą na jednostkę powierzchni.

Po wykonaniu pomiarów na obrazie otrzymuje się wielkości opisujące dane obiekty, które trzeba poddać dalszemu przetwarzaniu aby określić ich charakterystyczne cechy. Podstawowym zadaniem analizy kształtów jest wyznaczenie takich cech obiektów, które dobrze charakteryzują ich kształty, przyjmując jednakowe wartości dla obiektów mających jednakowy kształt (na przykład dla wszystkich kwadratów), a równocześnie dobrze różnicujących obiekty o różnych kształtach (na przykład koła od kwadratów).

Ponieważ z procesem wprowadzania obrazu wiążą się pewne deformacje obiektów, które najczęściej w procesie analizy należy ignorować, dlatego interesują nas takie współczynniki, które są niezmienne względem typowych przekształceń obrazów, jakimi są obroty, przesunięcia i zmiany skali. Parametry otrzymany wane drogą pomiarów bezpośrednio na obrazie, takie jak współrzędne obiektów, pole, obwód, itd., nie wykazują oczekiwanej niezmienności względem powyższych przekształceń. Większość pomiarów przeprowadzana jest na obrazie binarnym, chociaż zdarzają się wyjątki od tej reguły.

#### 6.4.2 Liczebność elementów

Warunkiem wyjściowym do liczenia częstek jest wyznaczenie ramki (prostokątnego wycinka obrazu), w której przeprowadzony zostanie pomiar. Obiekty

widoczne na obrazie stanowią zwykle jedynie niewielki fragment większej całości i pewne części są przecinane przez brzeg. Zatem, gdyby analizować po kolejnych przylegających do siebie obszary o rozmiarach analizowanego obrazu i zliczać wszystkie widoczne obiekty, to obiekty przecięte przez brzeg byłyby zliczane dwukrotnie, a części znajdujące się w narożnikach obrazu - nawet czterokrotnie. Aby poprawnie zliczać obiekty można zastosować jeden z trzech sposobów:

1. Usunięcie wszystkich elementów przeciętych przez brzeg obrazu i zliczanie elementów wewnętrznych (unika się wielokrotnego zliczania części obiektów, ale otrzymany wynik jest zaniedbany).
2. Przyjęcie odpowiedniej poprawki, która eliminuje wielokrotne zliczanie. I tak obiekty przecięte przez brzeg liczymy z wagą 0,5, natomiast obiekty znajdujące się w narożnikach obrazu - z wagą 0,25.
3. Wykorzystanie idei ramki bezpieczeństwa, biorąc pod uwagę całe obiekty, a nie ich części. Na obrazie zaznacza się ramkę i do analizy bierze się obiekty zawarte wewnątrz ramki oraz przecięte przez dwa jej brzegi, np. prawy i dolny. Obiekty przecięte przez pozostałe dwa brzegi ramki pomija się. Przy statystycznie przypadkowym wyborze ramki prawdopodobieństwo przecięcia obiektu przez jedną z par brzegów: prawy - dolny i lewy - górny jest takie samo. Zatem sposób ten daje zarówno poprawne oszacowanie liczby obiektów, jak również umożliwia pełną ich charakterystykę.

Samo policzenie ilości części realizowane jest przez algorytm etykietowania<sup>39</sup> obrazu. Komputer analizuje punkt po punkcie zapamiętany obraz. W momencie, gdy natrafi na punkt należący do obiektu, wyszukuje te punkty, które są z nim spójne. Wszystkim tym punktom nadaje się pewną nową wartość, czyli indeks. Polega to na iteracji następującej procedury, prowadzonej poprzez wszystkie punkty obrazu. Na początku żaden punkt obrazu nie posiada etykiety. Punkty należące do tła są pomijane. Pierwszemu napotkanemu punktowi należącemu do wydzielonego na etapie segmentacji obiektu nadawana jest etykieta ①. Ta sama etykieta nadawana jest wszystkim punktom z nim spójnym. Następnemu punktowi należącemu do obiektu, a nie posiadającemu jeszcze etykiety, nadawana jest etykieta o numerze o jeden większym. I tak dalej, aż do przeanalizowania wszystkich punktów obrazu. Przy dalszych poszukiwaniach pomija się te punkty, którym już nadano indeks, a każdy następny indeks ma numer o jeden większy. W ten sposób wszystkie części zostają ponumerowane. Po indeksacji nie ma już w gruncie rzeczy potrzeby zliczania obiektów. Częstek jest bowiem tyle, ile wynosi największy numer indeksu. Dzięki indeksacji możliwe są wszystkie inne pomiary na indywidualnych obiektach, gdyż komputer ma wyznaczone „granice” dla każdego analizowanego obiektu.

---

<sup>39</sup> ang. *lubelling*

Algorytm etykietowania obrazu jest również etapem wstępny wielu innych pomiarów, gdyż w prosty sposób wyodrębnia z obrazu poszczególne obiekty, które można poddać niezależnym pomiaram.

#### 6.4.3 Pole powierzchni

Pole powierzchni jest parametrem wyjątkowo prostym do wyznaczenia i można go wyznaczyć z dużą dokładnością. Pomiar pola powierzchni sprowadza się do zliczenia punktów obrazu należących do interesującego obszaru. Dodatkowo po wycechowaniu obrazu, czyli podaniu, jakiej rzeczywistej odległości odpowiada odległość wyróżnionych punktów obrazu, można otrzymać wynik pomiaru w takich jednostkach, jakie są wymagane.

Na wynik pomiaru powierzchni ma duży wpływ sposób przeprowadzenia binaryzacji, poprzedzającej pomiar. Nawet niewielka zmiana progu binaryzacji może mieć znaczący wpływ na wynikowe pole powierzchni.

#### 6.4.4 Długości krawędzi

Pomiar długości krawędzi obszaru jest sprawą trudną. Największą trudność sprawia przybliżenie ciągłej krzywej poprzez dyskretną kombinację punktów obrazu. W praktyce wykorzystuje się następujące sposoby pomiaru długości brzegu figury:

- **Zliczenie punktów brzegowych figury.** Jest to najprostsza metoda, ale daje duże zafalzowanie, w szczególności dla małych figur.
- **Zliczenie punktów brzegowych figury z uwzględnieniem ich położenia.** Punkty zlicza się z odpowiednią wagą. Dla punktów brzegowych siedzących w pionie lub poziomie stosuje się wagę wielkości 1, a dla punktów siedzących po przekątnej - wagę  $\sqrt{2}$ . Jest to metoda daleko doskonalsza od poprzedniej.
- **Wyliczenie średniej z długości zliczonych po zewnętrznej i wewnętrznej stronie brzegu figury.** Metoda ta daje dokładniejszy wynik, niż poprzednie.
- **Przybliżenie figury odpowiednim wielokątem.** Jest to metoda teoretycznie najlepsza, ale w praktyce brak jest jednoznacznego zdefiniowania sposobu wyboru odpowiedniego wielokąta przybliżającego figury.
- **Przybliżanie długości brzegu liniami krzywymi** (metoda ta może wprowadzać znaczne błędy, szczególnie dla obiektów o prostolinowych krawędziach)
- **Wykorzystanie tzw. formuły Croftona.** Metoda ta polega na korzystaniu z zasady Cauchy'ego:

$$L = \int_0^{\pi} D(\alpha) d\alpha$$

gdzie:  $L$  - obwód,  $\alpha$  - kąt rzutu,  $D(\alpha)$  – długość rzutu.

Z formuły Cauchy'ego po przekształceniach otrzymujemy formułę Croftona dla siatki kwadratowej. Dla obrazu dyskretnego przy siatce prostokątnej ograniczamy się do sumowania rzutów w czterech podstawowych kierunkach (z uwzględnieniem różnic występujących w odległościach pomiędzy punktami dla kątów  $45^\circ$  i  $135^\circ$ ):

$$L = \frac{\pi}{4} \cdot [a \cdot (N_0 + N_{90}) + \frac{a}{\sqrt{2}} \cdot (N_{45} + N_{135})]$$

gdzie:  $N_0, N_{90}, N_{45}, N_{135}$  - rzuty figury dla wybranych kierunków rzutowania

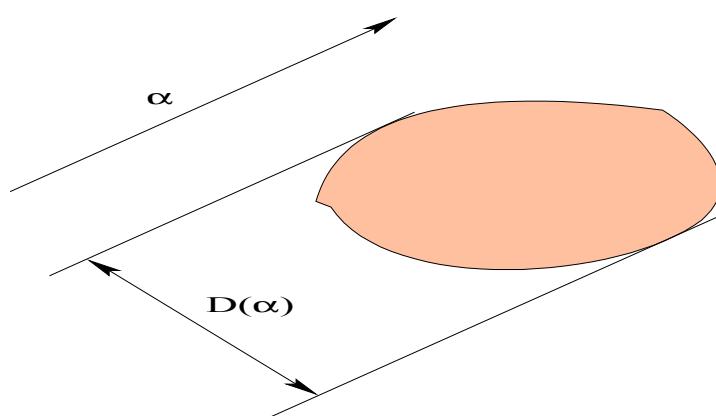
$a$  - odległość punktów siatki

Z powodu ograniczenia jedynie do czterech kierunków opisany sposób daje mało dokładne wyniki, ale jest bardzo szybki i prosty w praktycznej realizacji.

Generalną zasadą przy pomiarze długości na obrazie cyfrowym jest fakt, że im większą dokładność pomiaru się przyjmie, tym większy otrzymuje się wynik pomiaru. Ta, z pozoru irracjonalna, własność jest jedną z podstaw teorii fraktali.

#### 6.4.5 Długości rzutów

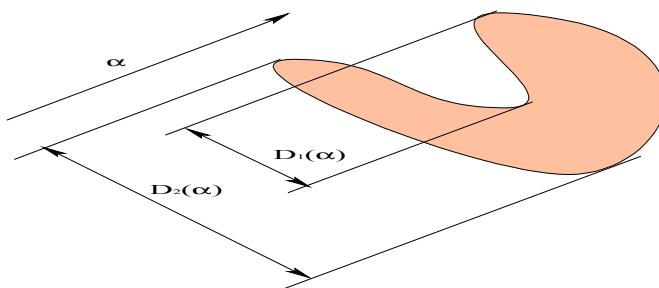
Rzutem  $D(\vec{\alpha})$  figury w kierunku wektora rzutowania  $\vec{\alpha}$  nazywamy największą odległość pomiędzy wszystkimi prostymi, równoległymi do wektora  $\vec{\alpha}$ , mający mi część wspólną z analizowaną figurą. Inaczej można to też wyrazić w sposób następujący: rzutem  $D(\alpha)$  figury w kierunku  $\alpha$  będziemy nazywali długość odcinka prostopadłego do kierunku rzutowania, przecinającego wszystkie proste o kierunku  $\alpha$ , które trafiają w analizowaną figurę.



Rys. 6.17. Rzut figury.

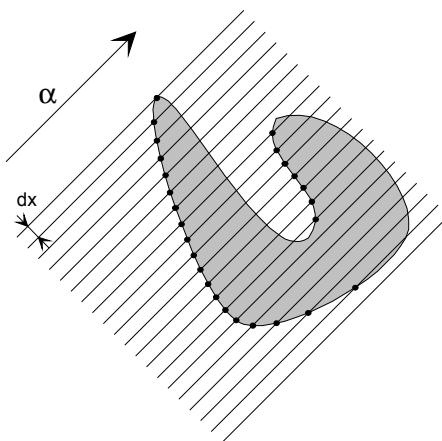
W przypadku figur wypukłych, tak jak na rys. 6.17, mamy dwie linie styczne do figury, zaś każda inna linia, która trafia w figurę, ma dwa punkty wspólne z brzegiem figury. Sprawa komplikuje się, gdy analizujemy figurę wklęsłą. Na rys. 6.18 można zauważać, że w rejonie wklęsłości proste przecinające figurę mają więcej niż dwa punkty wspólne z brzegiem figury. W tej sytuacji można wyznaczyć dwa rzuty cząstkowe, zaznaczone odpowiednio jako  $D_1$  i  $D_2$ . Sumując je otrzymamy rzut rozwinięty, będący uogólnieniem normalnego rzutu prostopadłego:

$$D(\alpha) = D_1 + D_2$$



Rys. 6.18. Rzut rozwinięty.

Rzut rozwinięty można łatwo wyznaczyć na obrazie komputerowym. Dla danej cząstki należy wyszukać wszystkie punkty, których lokalne otoczenie odpowiada „wchodzeniu” siecznej do cząstki, zliczyć te punkty i pomnożyć otrzymany wynik przez odległość między kolejnymi siecznymi  $dx$ . Oczywiście obraz komputerowy jest obrazem dyskretnym i gęstość linii rzutujących zależy od gęstości rastra tworzącego obraz.



Rys. 6.19. Wyznaczanie rzutu rozwiniętego figury.

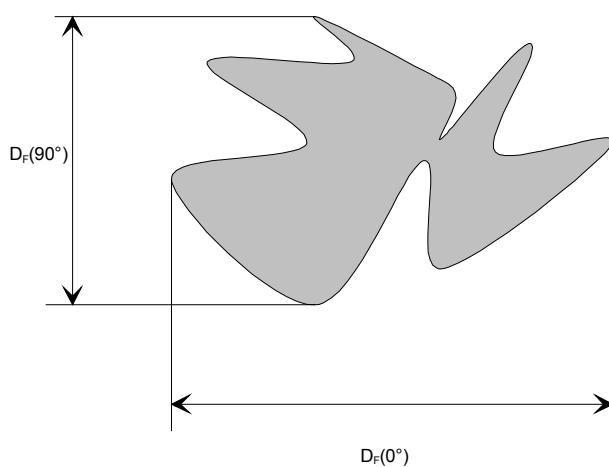
Największą dokładność uzyskuje się w kierunkach zgodnych z największym upakowaniem punktów w siatce obrazu. Dla siatki prostokątnej są to następujące kierunki:  $0^\circ$ ,  $45^\circ$ ,  $90^\circ$  i  $135^\circ$ . W tabeli 6.1. zebrano przykładowe elementy strukturalne do wyznaczenia długości rzutów dla podanych kątów.

Tabela 6.1. Przykładowe elementy strukturalne do wyznaczenia długości rzutów figury

kąt	otoczenie
$0^\circ$	$\begin{bmatrix} X & X & X \\ X & 0 & 1 \\ X & X & X \end{bmatrix}$
$45^\circ$	$\begin{bmatrix} X & X & 1 \\ X & 0 & X \\ X & X & X \end{bmatrix}$
$90^\circ$	$\begin{bmatrix} X & 1 & X \\ X & 0 & X \\ X & X & X \end{bmatrix}$
$135^\circ$	$\begin{bmatrix} 1 & X & X \\ X & 0 & X \\ X & X & X \end{bmatrix}$

#### 6.4.6 Średnice Fereta

Często wykorzystywanymi parametrami opisującymi liniowe rozmiary cząstek są tzw. *średnice Fereta*. Są one miarą rozciągłości cząstki w pionie i poziomie (rys. 6.20).



Rys. 6.20. Średnice Fereta.

Wyznaczenie ich jest bardzo proste. Dla przykładu, dla poziomej średnicy Fereta, wystarczy wyliczyć różnicę pomiędzy maksymalną i minimalną poziomą współrzędną wszystkich punktów należących do danej figury.

#### 6.4.7 Momenty bezwładności

Prostymi do wyznaczenia są także momenty bezwładności figur. Momenty bezwładności pierwszego rzędu ( $M_{1X}$ ,  $M_{1Y}$ ) określają położenie środka ciężkości figury, natomiast momenty drugiego rzędu są miarą bezwładności danej figury.

Wyznaczenie poszczególnych momentów bezwładności sprowadza się do prostego sumowania po wszystkich punktach figury:

$$\begin{aligned} M_{1X} &= \frac{1}{A(X)} \sum_{\text{X}} x_i \\ M_{1Y} &= \frac{1}{A(X)} \sum_{\text{X}} y_i \\ M_{2X} &= \frac{1}{A(X)} \sum_{\text{X}} (x_i - M_{1X})^2 \\ M_{2Y} &= \frac{1}{A(X)} \sum_{\text{X}} (y_i - M_{1Y})^2 \\ M_{2XY} &= \frac{1}{A(X)} \sum_{\text{X}} (x_i - M_{1X})(y_i - M_{1Y}) \end{aligned}$$

gdzie:  $X$  - analizowana figura,  
 $A(X)$  - pole powierzchni figury  $X$ ,  
 $(x_i, y_i)$  - współrzędne poszczególnych punktów figury.

Wyznaczenie momentów bezwładności da się w naturalny sposób rozszerzyć na obrazy wieloodcieniowe. Wystarczy każdy analizowany punkt zsumować z wagą będącą stopniem jego jasności.

### 6.5 Współczynniki kształtu

#### 6.5.1 Definicje współczynników kształtu

Ogólnie współczynniki kształtu można podzielić na dwie grupy. Pierwsza to współczynniki, które charakteryzują się głównie możliwością ich szybkiego obliczania, co jest bardzo ważne w systemie, który musi dostarczać wyników przetwarzania obrazu w czasie rzeczywistym. Można tu wymienić następujące współczynniki:

1° Bezwymiarowy współczynnik kształtu do ilościowego charakteryzowania kształtu cząstek:

$$R_S = \frac{L^2}{4\pi \cdot S}$$

gdzie:  
 L - obwód obiektu  
 S - pole powierzchni obiektu

2° Współczynnik Fereta (charakteryzuje wydłużenie cząstki)

$$R_F = \frac{L_h}{L_v}$$

gdzie:  
 L<sub>h</sub> - maksymalna średnica obiektu w poziomie  
 L<sub>v</sub> - maksymalna średnica obiektu w pionie

3° Współczynniki cyrkularności:

$$R_{C1} = 2 \cdot \sqrt{\frac{S}{\pi}} \quad R_{C2} = \frac{L}{\pi}$$

gdzie:  
 L - obwód obiektu  
 S - pole powierzchni obiektu

4° Współczynnik Malinowskiej:

$$R_M = \frac{L}{2 \cdot \sqrt{\pi \cdot S}} - 1$$

gdzie:  
 L - obwód obiektu  
 S - pole powierzchni obiektu

Drugą grupę natomiast tworzą współczynniki, które wykorzystywane są w przypadkach, kiedy szybkość działania nie jest najważniejsza, a istotna jest dokładność opisu obiektów za pośrednictwem ich cech. Do tej grupy należą następujące współczynniki:

1° Współczynnik Blaira-Blissa:

$$R_B = \frac{S}{\sqrt{2\pi \cdot \sum_i r_i^2}}$$

gdzie:  
 S - pole powierzchni obiektu  
 r<sub>i</sub> - odległość piksela obiektu od środka ciężkości obiektu  
 i - numer piksela obiektu

2° Współczynnik Danielssona:

$$R_D = \frac{S^3}{\left( \sum_i l_i \right)^2}$$

gdzie:  $S$  - pole powierzchni obiektu

$l_i$  - minimalna odległość piksela obiektu od konturu obiektu

$i$  - numer piksela obiektu

3° Współczynnik Haralicka:

$$R_H = \sqrt{\frac{(\sum_i d_i)^2}{n \cdot \sum_i d_i^2 - 1}}$$

gdzie:  $d_i$  - odległość pikseli konturu obiektu od jego środka ciężkości

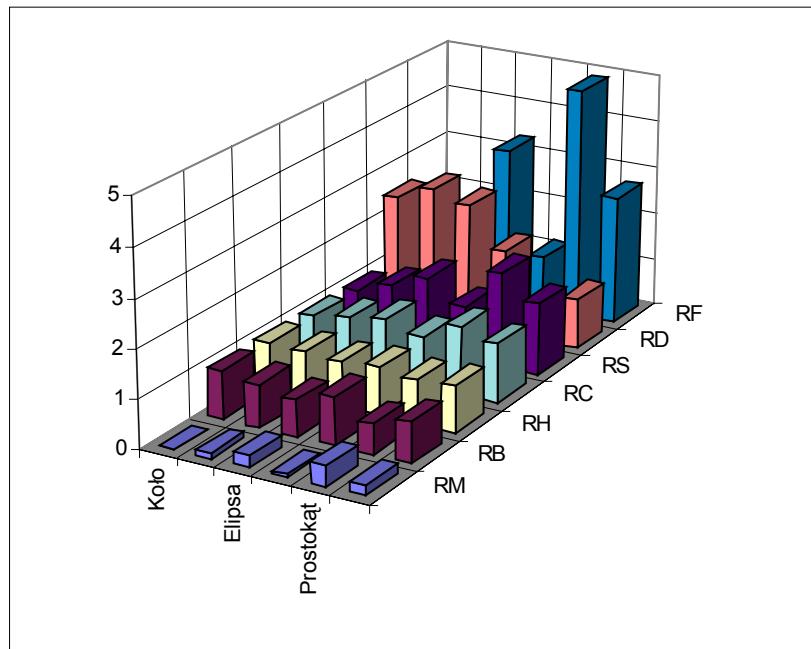
$n$  - liczba pikseli konturu

$i$  - numer piksela obiektu

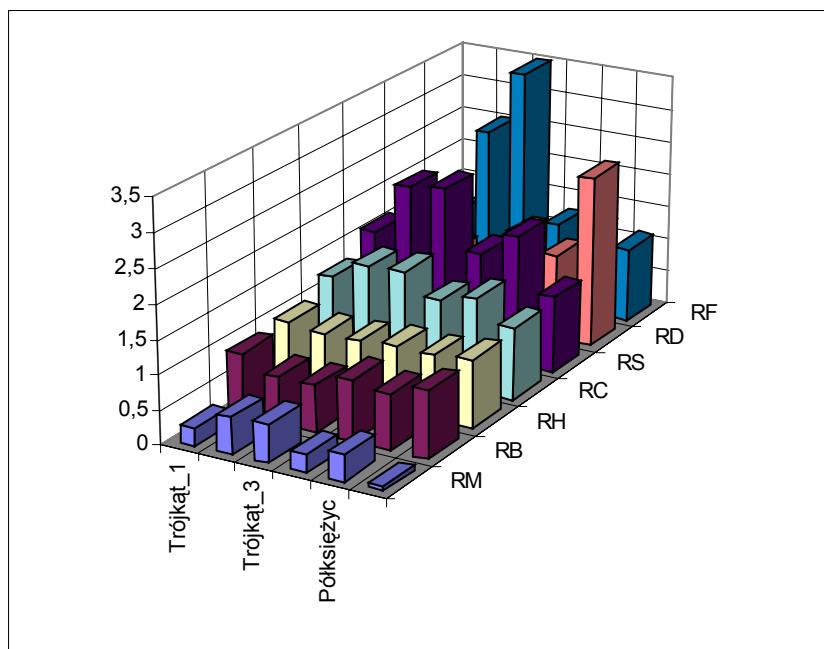
Przedstawione dwie grupy cech można wykorzystywać łącznie lub można wybrać tylko niektóre współczynniki, decydując się albo na dokładniejsze odwzorowanie kształtu obiektów, albo na szybsze działanie algorytmu.

### 6.5.2 Własności współczynników kształtu

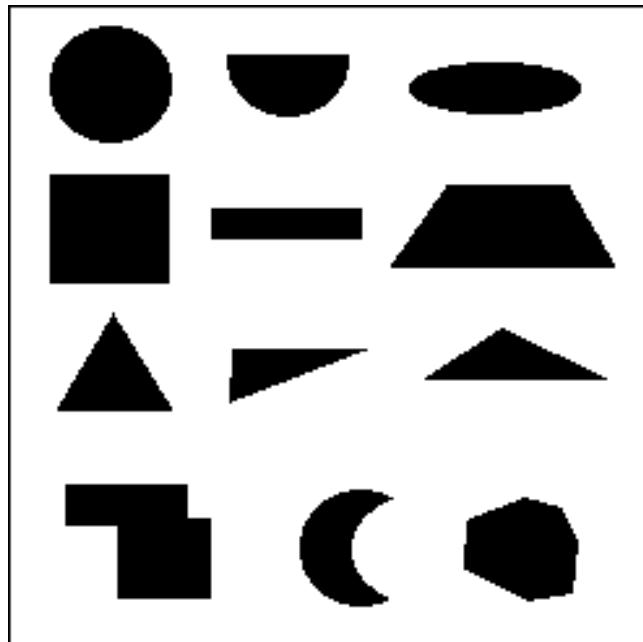
Współczynniki kształtu powinny przede wszystkim dobrze różnicować figury o różnych kształtach. W tym zakresie właściwości współczynników opisanych w poprzednim podrozdziale ilustrują rysunki 6.21 i 6.22. Pokazane na tych rysunkach wartości współczynników kształtów wyznaczane były dla obrazu figur geometrycznych, przedstawionego na rysunku 6.23. Na rysunkach widać, że największą zmiennością cechuje się współczynnik Fereta. Dla obiektów bardziej wydłużonych ma on większą wartość, niż dla obiektów o zwartym kształcie jak koło, kwadrat, trójkąt czy siedmiokąt. Dużą zmiennością charakteryzuje się także współczynnik Danielssona, który w odróżnieniu od współczynnika Fereta przyjmuje małą wartość dla obiektów wydłużonych. Mniejszy zakres zmienności wykazuje natomiast współczynnik Malinowskiej i jego odmiana - współczynnik  $R_S$ , ale tak samo jak w poprzednim przypadku dla elementów o wydłużonym kształcie wartości są większe. Współczynnik  $R_C$  jest stosunkiem dwóch współczynników cyrkularności, które określają odpowiednio  $R_{C1}$  - średnicę koła o obwodzie równym obwodowi analizowanego obiektu, a  $R_{C2}$  - średnicę koła, którego pole jest równe polu danego obiektu. Współczynnik ten cechuje podobny zakres zmienności jak Blaira-Blissa, ale współczynnik  $R_C$  przyjmuje duże wartości dla obiektów o wydłużonym kształcie. Najmniejszy przedział zmienności wykazuje współczynnik Haralicka.



Rys. 6.21. Wrażliwość współczynników kształtu dla różnych prostych figur geometrycznych.



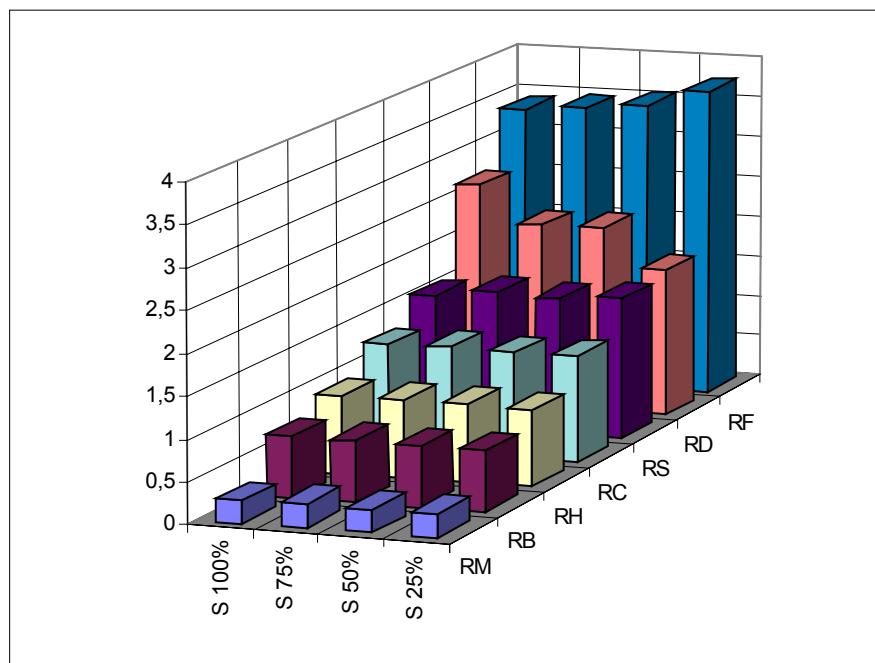
Rys. 6.22. Wrażliwość współczynników kształtu dla różnych bardziej złożonych figur geometrycznych.



6.23. Cyfrowe obrazy figur geometrycznych wykorzystane do ilustrowania własności współczynników kształtu.

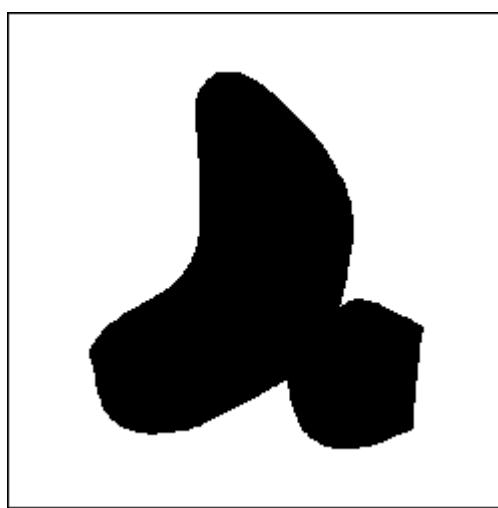
Współczynniki kształtu **powinny** być wrażliwe na zmienność kształtu figury, powinny jednak zachować niewrażliwość (niezmienność) jeśli idzie o zmianę sposobu przedstawienia figury. Na przykład na rysunku 6.24 pokazano zestaw wartości współczynników kształtu dla tej samej figury (elipsy) przy różnych (wyrażonych w procentach) stopniach jej powiększenia. Analizując wykresy na rysunku 6.25 można zaobserwować, że największą wrażliwością na zmiany skali cechuje się współczynnik Fereta. W rozważanym przypadku elipsy występują dość duże zmiany współczynnika Danielssona, przy czym wraz ze zmniejszaniem się skali współczynnik ten zmniejsza swoją wartość, a współczynnik Fereta - zwiększa. W przypadku innych figur (na przykład prostokąta) współczynnik Danielssona wykazuje o wiele mniejsze zmiany podobnie jak pozostałe współczynniki.

Biorąc pod uwagę złożoność obliczeniową poszczególnych współczynników, można stwierdzić, że największą odporność na zmianę skali wykazuje współczynnik Haralicka, ale jednocześnie charakteryzuje się on małą zmiennością. Współczynnik Fereta jest najłatwiejszy do wyliczenia ale przez to jest też bardzo podatny na zmiany skali. Natomiast jego zaletą jest o wiele większy zakres zmienności niż w przypadku współczynnika Haralicka. Najbardziej czasochłonnym obliczeniowo jest współczynnik Danielssona, który w dodatku wykazuje dużą zmienność przy zmianie skali dla niektórych figur (np. elipsy), jednakże cechuje go duży przedział zmienności dla różnych figur.

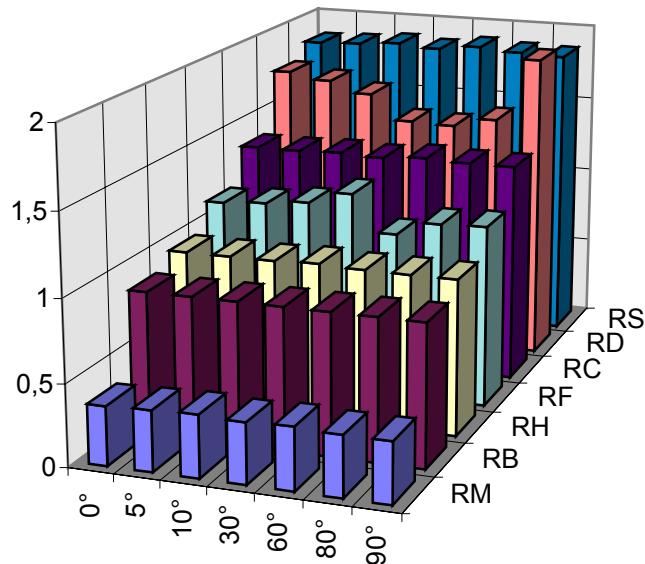


Rys. 6.24. Wartości różnych współczynników kształtu dla tej samej figury przy różnych powiększeniach.

Współczynniki kształtu powinny też wykazywać dużą niezależność od obrotów. Na rysunku 6.26. zebrano wartości wszystkich rozważanych współczynników kształtów dla obiektu pokazanego na rysunku 6.25, który poddano obrotowi pod siedmioma kątami:  $0^\circ, 5^\circ, 10^\circ, 30^\circ, 60^\circ, 80^\circ$  i  $90^\circ$ .



Rys. 6.25. Obraz z obiektem, który został poddany rotacji.

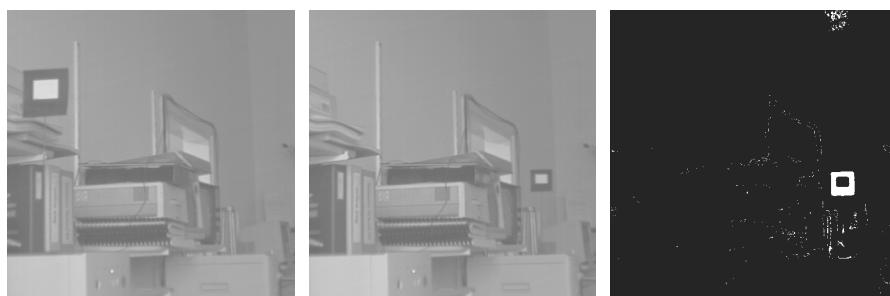


Rys. 6.26. Wartości współczynników kształtu uzyskane dla obracanego obiektu z rys. 6.25.

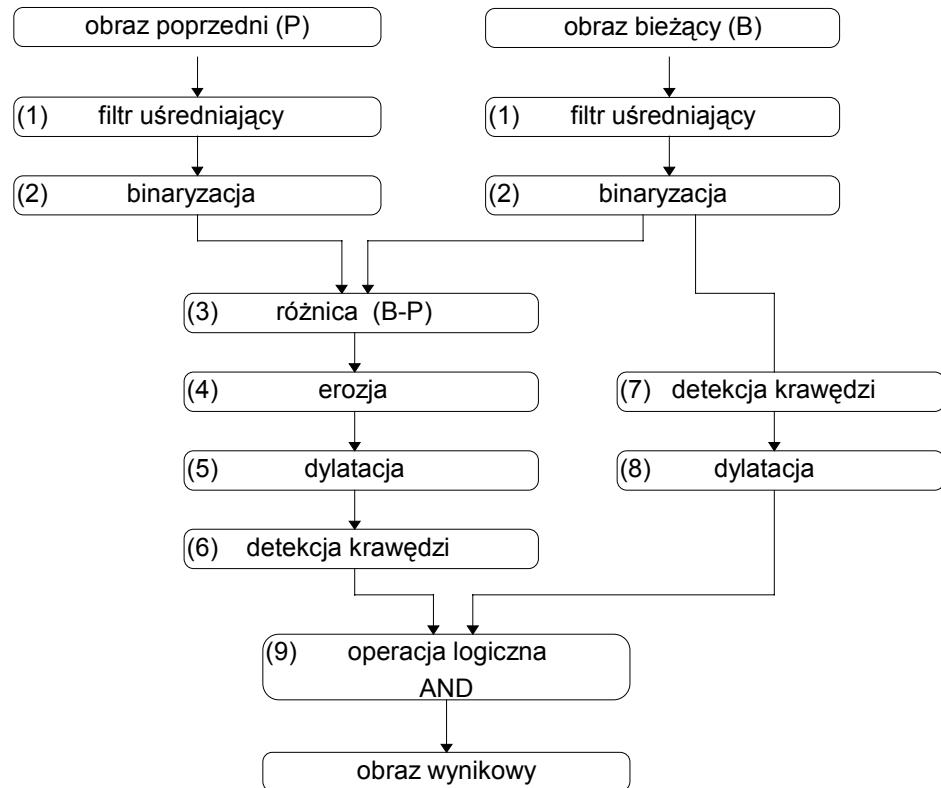
Na rysunku można zauważyć, że największe zmiany wykazuje współczynnik Fereta i współczynnik Danielssona. Natomiast pozostałe współczynniki ulegają nieznacznym zmianom.

## 6.6 Wykrywanie obiektów ruchomych

Do zagadnień związanych z analizą obrazu należy też technika wydzielania obiektów ruchomych. Zagadnienia wykrywania i śledzenia ruchomych obiektów na złożonych obrazach nie będą w tej książce szczegółowo omawiane, poprzedzającym więc tylko na pokazaniu (na rysunku 6.28) przykładowego algorytmu wydzielania ruchomych obiektów - a na rysunku 6.27 pokazano wyniki jego działania.



Rys. 6.27. Wykrywanie obiektów ruchomych na obrazie.



Rys. 6.28. Schemat blokowy algorytmu wydzielenia obiektu.

## 7 Literatura

1. Bubliński Z., Cetnarowicz K.: *Zastosowanie elementu składowego do przetwarzania obrazów optycznych*. Elektrotechnika, Tom 10, Zeszyt 1, 1991.
2. Cetnarowicz K.: *A Practical Transformation Method for the Restoration of Degraded Images*. Scientific Bulletin of the faculty of Engineering Ain Shams University., Vol.2., No.17., pp.78 - 85, 1985.
3. Close R., Tamura S., Naito H.: *Estimation of motion from sequential images using integral constraints*, Pattern Recognition., Vol.28, No.1, pp.1 - 9, 1995.
4. Ćwirko R., Lissowski J., Mierzejewski Z., Walczyk J.: *Vademecum techniki audio video*, WNT, Warszawa 1991.
5. Deok J. Park, Kwon M. Nam: *Multiresolution Edge Detection Techniques*, Pattern Recognition, Vol.28, No.2, pp.211 - 229, 1995.
6. Deok J. Park, Kwon M. Nam: *Edge Detection in Noisy Images Based on the Co-occurrence Matrix*, Patern Recognition, Vol.27, No.6, pp.765 - 775, 1994.
7. Ens J., Lawrence P.: *An Investigation of Methods for Determining Depth from Focus*, IEEE Transactions on Pattern Analysis and Machine Intelligence., Vol.15, No.2, 1993.
8. Ens J., Lawrence P.: *A Matrix Based Method For Determining Depth From Focus*, IEEE Transactions on Pattern Analysis and Machine Intelligence., pp.600 - 606, 1991.
9. Folk A.: *Telewizja. Podstawy ogólne*, WKiŁ, Warszawa 1991.
10. Gonzales R.: *Digital Image Processing*. Addison - Wesley Publishing Company. 1987.
11. Kamiński R.: *Jak działa przetwornik CCD?*, SAT-Audio - Video nr 5, 1991.
12. Kamiński R.: *Cyfrowe przetwarzanie obrazu*, SAT-Audio - Video nr 3, 1991.
13. Kierzenkowski K.: *Algorytmy przetwarzania obrazów cyfrowych*, Elektronizacja nr 3, 1995.
14. Kuriański A.: *Detekcja i śledzenie ruchu przy użyciu czasowo - przestrzennego modelowania obrazów za pomocą pól losowych*, IPPT PAN, Warszawa 1994.
15. Metzger P.: *Anatomia PC*, Helion 1993.
16. Pavlidis T.: *Grafika i przetwarzanie obrazów*, WNT, Warszawa 1987.
17. Pentland A., Darrel T., Turk M., Huang W.: *A Simple, Real - Time Range Camera*, IEEE Comput. Soc. Conf. Comput. Vision Patt. Recogn., pp.256 - 261, 1989.
18. Pentland A. P.: *A New Sense for Depth of Field*, IEEE Transactions on Pattern Analysis and Machine Intelligence., Vol.PAMI-9, pp.523 - 531, 1987.

19. Postaire J. - G., Bontry F., Vieren C.: *Visual motion understanding for motion planning of an urban transport system*, Control, Computers, Communications in Transportation '89., pp. 95 - 102., Paris 1989.
20. Pratt K.: *Digital Image Processing*, Wiley & Sons, New York 1978.
21. Rusin M.: *Wizjne przetworniki optoelektroniczne*, WKiŁ, Warszawa 1990.
22. Sitharama S. Iyengar, Weian Deng: *An Efficient Edge Detection Algorithm Using Relaxation Labeling Technique*, Pattern Recognition, Vol.28, No.4, pp.519 - 536, 1995.
23. Tadeusiewicz R.: *Systemy wizjne robotów przemysłowych*, WNT, Warszawa 1992.
24. Wojnar L., Majorek W.: *Komputerowa analiza obrazu*, Fotobit Design, Kraków 1994.