

Zastosowanie metod morfologii matematycznej do detekcji i dekompozycji obrazów

Grzegorz Szczurek

Dokonano przeglądu operacji morfologii matematycznej najczęściej stosowanej w przetwarzaniu obrazów. Zademonstrowano również aplikację programową, wykorzystującą algorytm dekompozycji i detekcji obiektów obrazu oparty na jednej z operacji morfologicznych, zwanej „trafi nie trafi”. Przedstawiono wyniki działania algorytmu, wskazując jednocześnie możliwości jego modyfikacji i potencjalne zastosowania. Ponadto poruszono ważną, z punktu widzenia operacji morfologicznych, tematykę doboru odpowiedniego, dla określonej klasy obrazów, elementu strukturalnego.

morfologia matematyczna, przetwarzanie obrazów, element strukturalny, dylacja, erozja, otwarcie, domknięcie, operacja „trafi nie trafi”

Wprowadzenie

Powszechne dążenie do społeczeństwa informacyjnego powoduje, że informacje przekazywane w dzisiejszych czasach mają przede wszystkim formę obrazów (najczęściej w postaci cyfrowej), transmitowanych z zastosowaniem sieci Internet, telewizji, czy też telefonii komórkowej. Rośnie zatem zapotrzebowanie na różne metody przetwarzania obrazów, w zależności od ich przeznaczenia.

Znanych jest wiele metod przetwarzania obrazów stosowanych głównie do ich kodowania (kompresji) i dekodowania (dekompresji). Istnieje również klasa operacji przetwarzających obrazy pod kątem wyszukania ich określonych cech. Do takiej klasy należy dekompozycja kształtów obiektów oraz detekcja obiektów obrazów.

Obecnie, w związku z dynamicznym wzrostem mocy obliczeniowej dostępnych komputerów, jednym z kierunków przetwarzania obrazów o bardzo dużych możliwościach jest zastosowanie metod morfologii matematycznej. Najogólniej rzecz biorąc, metody te wykorzystują pewne właściwości obrazu, które nie są cechami obiektywnymi i dlatego mogą się ujawniać jedynie w wyniku współoddziaływania badanego obiektu (obrazu) z narzędziami badawczymi, zwanymi elementami strukturalnymi. Elementy te modyfikują kształt obiektu i jednocześnie ujawniają jego strukturę.

Najprostszy statyczny obraz cyfrowy jest przykładem funkcji dwóch zmiennych niezależnych $f(x,y)$, dla której x,y są współrzędnymi punktu lub piksela w płaszczyźnie obrazu. W wielu przypadkach analizy obrazu przedstawiającego obiekt(y) zamiast opisu obrazu przy użyciu funkcji można opisać obraz za pomocą zbioru jedynek, które oznaczają przynależność elementu do obiektu pokazanego na obrazie. Wartościami zerowymi są oznaczone wówczas elementy obrazu stanowiące tło. Przy takim podejściu operacje morfologiczne są pierwotnie zdefiniowane jako przekształcenia zbiorów uporządkowanych.

Jedną z istotnych właściwości morfologii matematycznej jest możliwość uogólnienia jej pojęć z klasy obrazów z gradacją poziomów szarości, na klasę obrazów barwnych, czy – w bardziej skomplikowanych przypadkach – na klasę sekwencji obrazów.

Warto też zwrócić uwagę na kolejną, a jednocześnie najważniejszą cechę przekształceń morfologicznych (w odróżnieniu od innych), a mianowicie, że przekształcenia morfologiczne modyfikują tylko te części obrazu, których otoczenie jest zgodne z elementem strukturalnym. Inne przekształcenia uzależniają co prawda punkt od sąsiedztwa, ale operacje są wykonywane zawsze, nawet jeżeli wynik operacji nie wprowadza żadnych zmian. W praktyce, dla określonej klasy obrazu, ta właściwość może oznaczać znaczne zmniejszenie liczby wykonywanych operacji.

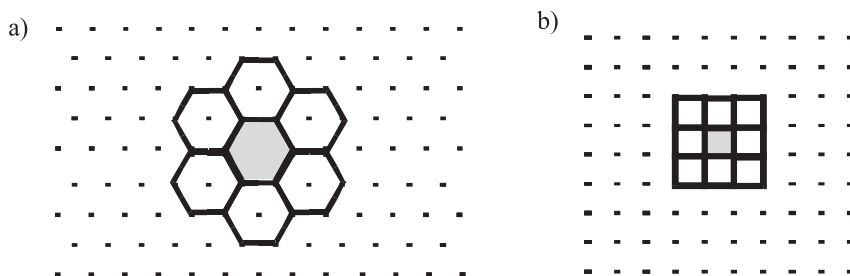
Inną użyteczną właściwością morfologii matematycznej w zastosowaniach przetwarzania obrazów jest występowanie hierarchii operacji, z których pewne, zwane podstawowymi, umożliwiają tworzenie operacji bardziej złożonych o dużo większych możliwościach aplikacyjnych.

W niniejszym artykule przedstawiono kilka operacji morfologii matematycznej stosowanych w przetwarzaniu obrazów oraz zaproponowano sposób wykorzystania niektórych z nich do dekompozycji kształtów obiektów obrazu.

Wybór elementu strukturalnego

Ze względu na sposób analizy obrazu bardzo ważnym zagadnieniem, związanym z poprawnością dekompozycji i rekonstrukcji kształtów obiektów obrazów, jest właściwy wybór oraz zastosowanie elementu strukturalnego użytego do operacji morfologicznych. Najbardziej popularnymi realizacjami elementów strukturalnych, opisanymi w [11], są (rys.1):

- dziewięcioelementowy element strukturalny o siatce kwadratowej,
- siedmioelementowy element strukturalny z siatką heksagonalną, będący kołem o promieniu jednostkowym, ze składowymi tworzącymi strukturę podobną do struktury „plastra miodu”.



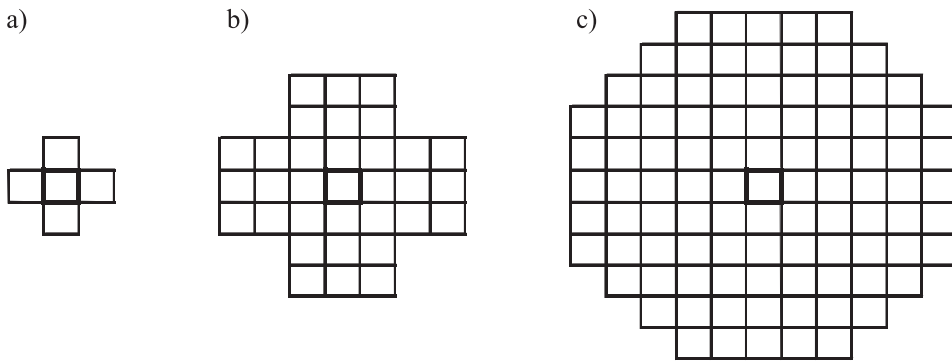
Rys. 1. Najczęściej stosowane elementy strukturalne: a) w siatce heksagonalnej; b) w siatce kwadratowej

W operacjach przybliżania obrazów lepsze rezultaty uzyskuje się, stosując element strukturalny oparty na siatce heksagonalnej. Przemawiają za tym następujące cechy tego elementu:

- lepsze przybliżenie koła, które znacznie częściej analizuje się teoretycznie w większości przekształceń;
- siedem elementów w kole jednostkowym, w porównaniu z dziewięcioma w przypadku podstawowego elementu strukturalnego na siatce kwadratowej, oznacza mniejszą liczbę operacji;

- wynik przekształcenia morfologicznego w siatce heksagonalnej jest bardziej zgodny z intuicją, ponieważ przekształcenie to nie wyróżnia wówczas dwóch abstrakcyjnych kierunków – pionowego i poziomego.

Pomimo wymienionych zalet w zastosowaniach praktycznych powszechniej jest stosowany dziewięcio-elementowy element strukturalny oparty na siatce kwadratowej. Jego powszechność wynika głównie z faktu, że większość obrazów jest przedstawianych jako matryca prostokątna, której piksele sąsiadują ze sobą zarówno w poziomie, jak i w pionie. Obrazy w takiej formie w bardzo przejrzysty sposób zapisuje się w pamięci komputera czy pliku jako mapy bitowe w postaci tablic.



Rys. 2. Element strukturalny wykorzystany przy rekonstrukcji obrazu: a) element najmniejszy; b) element użyty o jedną iterację wcześniej niż najmniejszy; c) element użyty o dwie iteracje wcześniej niż najmniejszy

Jednym z rozwiązań poszukiwania optymalnego elementu strukturalnego jest znalezienie takiego jego kształtu, który możliwie najlepiej przybliży koło jednostkowe na siatce heksagonalnej i daje się opisywać na siatce kwadratowej. Element o takich właściwościach ma kształt pokazany na rys. 2. Ze względu na prostszą implementację programową jest to element symetryczny. Przybliży on koło jednostkowe, co jest jego niewątpliwą zaletą, jednak charakteryzuje się dość poważną niedogodnością. Wprawdzie swym kształtem przybliży on obszar koła lepiej niż odpowiadający mu rozmiarami element sześcioboczny, ale nie umożliwi dobrego pokrywania jednolitych obszarów. Oznacza to, że dla prawidłowego zobrazowania jednorodnego obszaru konieczne jest wypełnienie powstałych kwadratowych „luk”.

Upraszczenie informacji zawartej w obrazie

Kształt elementu strukturalnego odpowiada za detekcję kształtów obiektu obrazu ze względu na jego kontury, natomiast wartości elementów składowych elementu strukturalnego odpowiadają za analizę wnętrza obiektu. Jeżeli obiekty obrazu mają jednolite tło, to do ich detekcji wystarcza jeden jednolity element strukturalny. W przypadku obiektów, których piksele zawierają różne wartości odcieni szarości, detekcję lub dekompozycję można realizować z wykorzystaniem elementu strukturalnego składającego się z elementów o różnych wartościach poziomów jasności lub kilku jednolitych elementów strukturalnych, z których każdy ma inną wartość poziomu szarości. W analogiczny

sposób można potraktować obrazy barwne, gdzie sytuacja przedstawiona dla obrazów z gradacją poziomów szarości znajduje odniesienie w każdej ze składowych RGB. Przy takim podejściu do problemu informacją o detekcji obiektu będzie zbiór znaczników, z których każdy będzie ustawiany po uzyskaniu zgodności odpowiedniego elementu strukturalnego z częścią poszukiwanego obiektu. Jeśli poszukiwany obiekt ma złożoną budowę, to jego detekcja z zastosowaniem wielu elementów strukturalnych jest procesem długotrwałym i złożonym obliczeniowo. Z tego względu poszukuje się takich metod upraszczania obrazów, które znacznie zmniejszają liczbę poziomów jasności, lecz nie zniekształcają obrazu na tyle, aby uniemożliwić detekcję poszukiwanego obiektu. Metody te zależą od charakteru poszukiwanego obiektu.

Podstawowe metody upraszczania treści przekazywanej przez obraz są opisane między innymi w pracach [6, 9, 11]. Metody upraszczania informacji, które dopuszczają pewne straty informacji w stosunku do oryginału, można podzielić na metody [9]:

- zmniejszające liczbę poziomów kwantowania,
- zmniejszające rozmiar funkcji szarości (przez zastępowanie kilku pikseli jednym).

Do grupy metod zmniejszających liczbę poziomów kwantowania można zaliczyć redukcję poziomów szarości (*compress gray*). Jest to jedna z prostszych, a jednocześnie często stosowanych, metod upraszczania informacji zawartej w obrazach z gradacją poziomów szarości. Operacja redukcji poziomów szarości polega na zastąpieniu pewnej grupy pikseli, których różnice wartości szarości są możliwie najmniejsze, pikselami mającymi jedną, charakterystyczną dla tej grupy wartość poziomu szarości. Wynika stąd, że liczba poziomów szarości obrazu wynikowego jest równa liczbie grup wartości szarości pikseli, na które zostanie podzielona paleta wartości szarości obrazu wejściowego (256). Dobór wartości poziomu szarości charakterystycznej dla danego przedziału może być realizowany wieloma sposobami. Najczęściej wartość ta jest równa wartości środkowej tego przedziału szarości, który reprezentuje.

Morfologia matematyczna i podział operacji morfologicznych

Ze względu na złożoność operacje morfologiczne mają strukturę hierarchiczną. Można je podzielić na operacje podstawowe i złożone. Do grupy operacji złożonych należą operacje morfologiczne pierwszego, drugiego i wyższych rzędów [2]. Kryterium takiego podziału stanowi sposób konstrukcji algorytmów zaliczanych do metod morfologii matematycznej. Poniżej przedstawiono przykłady kilku najważniejszych operacji morfologicznych, koncentrując się głównie na operacjach podstawowych. Definiowanie wszystkich przekształceń bardziej złożonych jest pozbawione sensu ze względu na ich nieograniczoną ilość (mogą one bowiem powstawać przez dowolne złożenie operacji prostych a także operacji złożonych o mniejszym stopniu komplikacji), dlatego omówiono tylko te najczęściej stosowane.

Podstawowe operacje morfologiczne

Podstawowymi metodami morfologii matematycznej są takie operacje, których nie można przedstawić w postaci innych, prostszych operacji morfologicznych pod względem ich działania [2]. Zalicza się do nich takie operacje, jak [2]: infimum, supremum, inwersja, różnica, progowanie, przesunięcie, symetryczne odbicie, dylacja i erozja.

Infimum dwóch obrazów f_1, f_2 w punkcie (x, y) stanowi ta z funkcji f_1, f_2 , której wartość dla danego piksela jest mniejsza:

$$(f_1 \cap f_2)(x, y) = \min \{f_1(x, y), f_2(x, y)\}, \quad (1)$$

gdzie:

$f_1(x, y), f_2(x, y)$ – wartości poziomów szarości pikseli dwóch obrazów wejściowych.

Supremum dwóch obrazów f_1, f_2 w punkcie (x, y) stanowi ta z funkcji f_1, f_2 , której wartość dla danego piksela jest większa:

$$(f_1 \cup f_2)(x, y) = \max \{f_1(x, y), f_2(x, y)\}, \quad (2)$$

gdzie:

$f_1(x, y), f_2(x, y)$ – wartości poziomów szarości pikseli dwóch obrazów wejściowych.

Inwersja jest operacją, w wyniku której wartość poziomu szarości piksela obrazu wyjściowego jest różnicą między liczbą k możliwych do uzyskania poziomów szarości a poziomem szarości piksela o tych samych współrzędnych, pochodzących z obrazu wejściowego:

$$(\sim f)(x, y) = k - f(x, y), \quad (3)$$

gdzie:

$(\sim f)(x, y)$ – funkcja poziomów szarości obrazu wyjściowego (inwersji),
 $f(x, y)$ – funkcja poziomów szarości obrazu wejściowego.

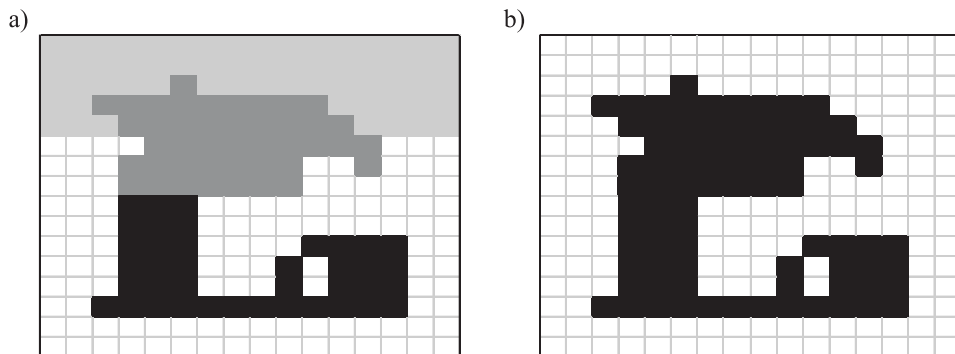
Wartość parametru k zależy od rodzaju obrazu. Dla obrazów z gradacją poziomów szarości $k \in \{0, 255\}$, natomiast dla binarnych – $k \in \{0, 1\}$.

Różnica dwóch obrazów opisanych funkcjami f_1, f_2 jest takie przekształcenie, w wyniku którego powstaje obraz z poziomami szarości pikseli, będących różnicą wartości poziomów szarości pikseli obrazu f_1 i odpowiadających im pikseli obrazu f_2 , przy czym różnica ta jest zdefiniowana w następujący sposób:

$$(f_1 - f_2)(x, y) = \begin{cases} f_1(x, y) - f_2(x, y), & \text{jeśli } f_1(x, y) \geq f_2(x, y) \\ 0 & \text{w przeciwnym przypadku} \end{cases} \quad (4)$$

Progowanie jest operacją polegającą na zmianie wartości poziomu szarości piksela na jedną z dwóch wartości, w zależności od wyniku porównania odpowiednich pikseli obrazów wejściowych f_1 i f_2 . Jedną jest wartość maksymalna, jaką może przyjąć poziom szarości piksela w danym rodzaju obrazu (dla obrazów o k odcieniach szarości jest to liczba k , niekoniecznie występująca w obrazie). Występuje ona w zależności od kryterium, jakie się przyjmuje i na tym etapie rozważań nie jest istotne, jakie ono jest. Ważne jest natomiast, że w przypadku niespełnienia zadanego kryterium wartość poziomu szarości piksela obrazu wyjściowego przyjmuje drugą z możliwych wartości – wartość zero. Wyniku działania operacji progowania nie można zaobserwować, gdy obrazami wejściowymi są obrazy binarne, ponieważ obrazy te już są obrazami dwupoziomowymi.

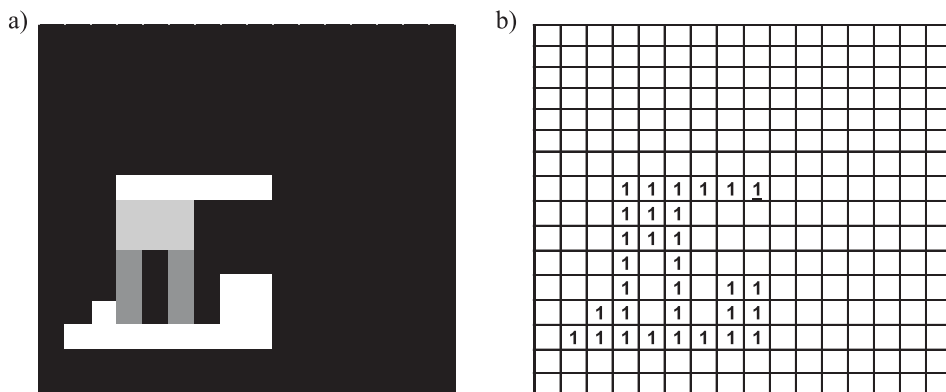
Jako kryterium progowania można przyjąć konkretną wartość poziomu szarości piksela. Jeżeli wartość poziomu szarości piksela obrazu wejściowego jest większa od ustawionej wartości progów, to wynikowa wartość poziomu szarości badanego piksela będzie uzyskiwać maksimum. W przeciwnym przypadku wartości te będą zerowane, a wówczas do przeprowadzenia operacji progowania wystarczający jest jeden obraz wejściowy. Sytuacja taka występuje podczas segmentacji z określonym progiem, która



Rys. 3. Prezentacja operacji progowania z progiem o określonej wartości poziomu szarości piksela: a) obraz przed operacją; b) obraz po operacji

proceeds to a binary image. An example of this type of thresholding is shown in Fig. 3. For the threshold value, the half of the maximum grayscale value of the pixel was chosen.

In many implementations of morphological operations, a simplification is used, treating images as algebraic sets. One set (represented by ones) consists of elements belonging to the object shown in the image, the other (represented by zeros) consists of elements belonging to the background. Since zero elements representing the background are the complement of the set of ones to all elements of the image, it is not necessary to mark them as zeros. It is sufficient to determine the elements belonging to the object. The situation is shown in Fig. 4, where the element marked with the origin of the coordinate system is underlined, elements marked with „1” are the object, and elements not marked are elements belonging to the background.



Rys. 4. Przedstawienie obrazu jako zbioru: a) obraz oryginalny; b) jego reprezentacja

Aby zdefiniować kolejne operacje, należy określić położenie punktu odniesienia (początku układu współrzędnych), na podstawie którego można ocenić wyniki operacji dokonywanych na obrazie – zbiorze A . Ponadto, aby zdefiniować większość operacji, należy przyjąć określony element (zbiór) strukturalny B wraz z jego elementem składowym b . Element b wyznacza kierunek i odległość przemieszczenia zbioru A . Zbiór B zawiera w sobie elementy b i określa wszystkie przemieszczenia, jakie są wykonywane w danej operacji. Przykładowo, jeżeli zbiór B ma postać:

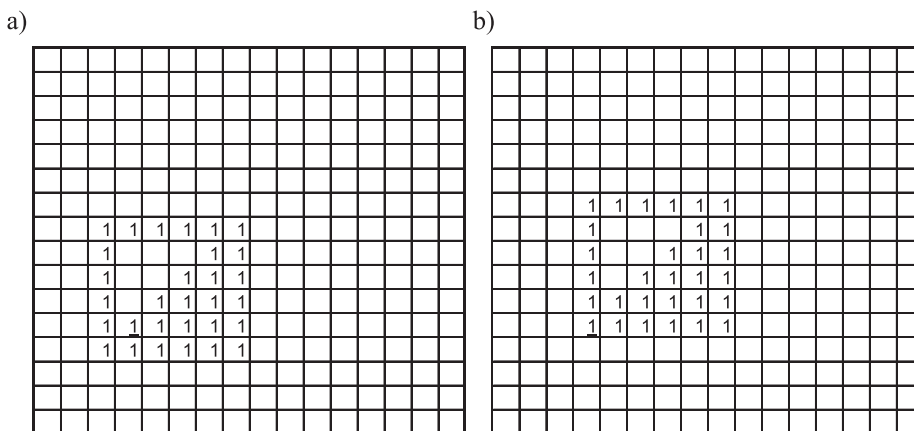
$$B = \begin{bmatrix} 0 & 1 \\ \underline{0} & 0 \end{bmatrix},$$

to element $b = (1, 1)$ oznacza przemieszczenie o jeden w kierunku poziomym i pionowym względem początku układu współrzędnych, określonego przez x .

Przesunięcie zbioru A o element b – należący do zbioru elementu strukturalnego B – jest to operacja, dająca w wyniku zbiór A_b , którego każdy element powstaje przez przesunięcie (translację) odpowiedniego elementu obrazu A o element b :

$$A_b = \{a + b : a \in A\}; \quad b \in B. \quad (5)$$

Przykład tej operacji podano na rys. 5, gdzie element strukturalny $b = (1, 1)$.

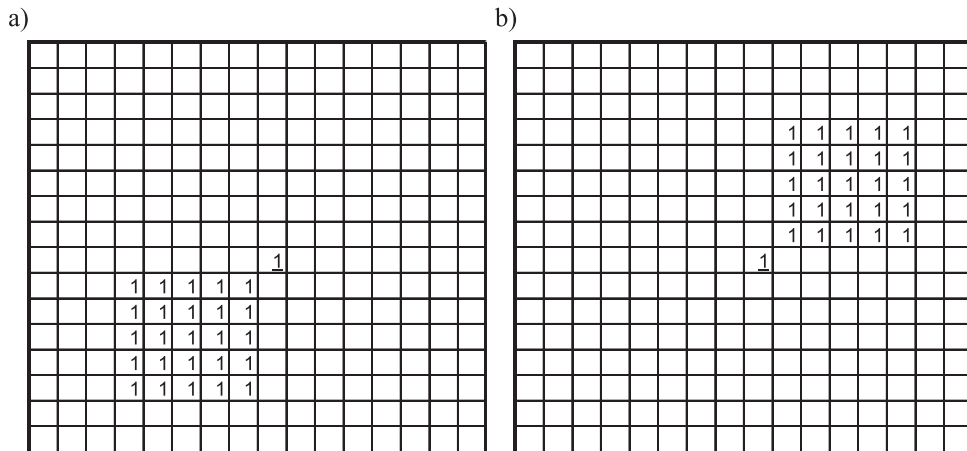


Rys. 5. Ilustracja operacji przesunięcia: a) obraz oryginalny; b) obraz przetworzony

Symetryczne odbicie obrazu (zbioru) A względem początku układu współrzędnych jest to operacja, w wyniku której otrzymuje się obraz (zbiór) A_R , którego każdy element powstaje przez symetryczne odbicie odpowiedniego elementu obrazu A względem początku układu współrzędnych:

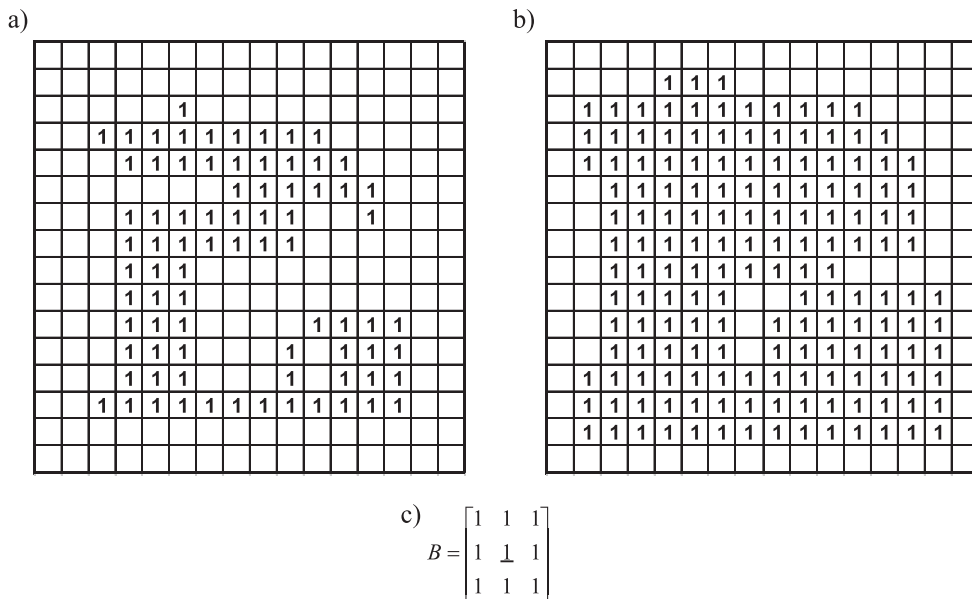
$$A_R = \hat{A} = \{a : -a \in A\}. \quad (6)$$

Na rysunku 6 zaprezentowano prosty przypadek odbicia względem początku układu współrzędnych.



Rys. 6. Przykładowa operacja odbicia symetrycznego: a) obraz oryginalny; b) obraz przetworzony

Kolejno definiowane operacje dylacji i erozji będą przedstawione w sposób poglądowy z pominięciem bardziej szczegółowego opisu matematycznego, który można znaleźć np. w pracach [1, 5–7].



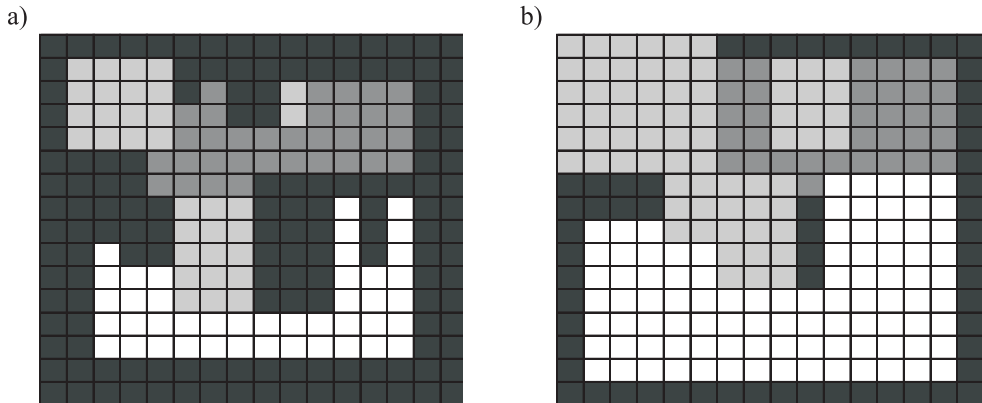
Rys. 7. Przykład operacji dylacji: a) zbiór wejściowy; b) zbiór po operacji dylacji; c) element strukturalny

Dylacja^① obrazu A przez element strukturalny B jest zdefiniowana jako suma przesunięć zbioru A o wszystkie elementy b , przy czym $b \in B$

$$A \oplus_s B = \bigcup_{b \in B} A_b. \quad (7)$$

Operację dylacji pokazano na rys. 7.

Praktycznie dylację wyznacza się, przemieszczając element strukturalny B po wszystkich elementach obrazu A . Wartość elementu, dla którego jest wyznaczana dylacja, jest wówczas sumą logiczną elementu strukturalnego B i części obrazu A , który jest nim przysłonięty. Dylację obrazów z gradacją poziomów szarości można traktować jako poszukiwanie lokalnego maksimum, przy czym obszar poszukiwań jest wyznaczony przez wielkość i kształt elementu strukturalnego. Ilustracją dylacji obrazu o czterech poziomach szarości jest rys. 8.



Rys. 8. Operacja dylacji obrazu z gradacją szarości: a) obraz oryginalny; b) obraz przetworzony

Dylacja jako operacja fundamentalna morfologii matematycznej ma wiele różnych właściwości, m.in. ekstensywność^②, monotoniczność^③ oraz niezmienniczość^④ względem przesunięcia.

Erozja obrazu A przez element strukturalny B jest zdefiniowana jako przecięcie przesunięć zbioru A o wszystkie elementy b , przy czym $b \in B$

$$A \ominus_s B = \bigcap_{b \in B} A_{-b}. \quad (8)$$

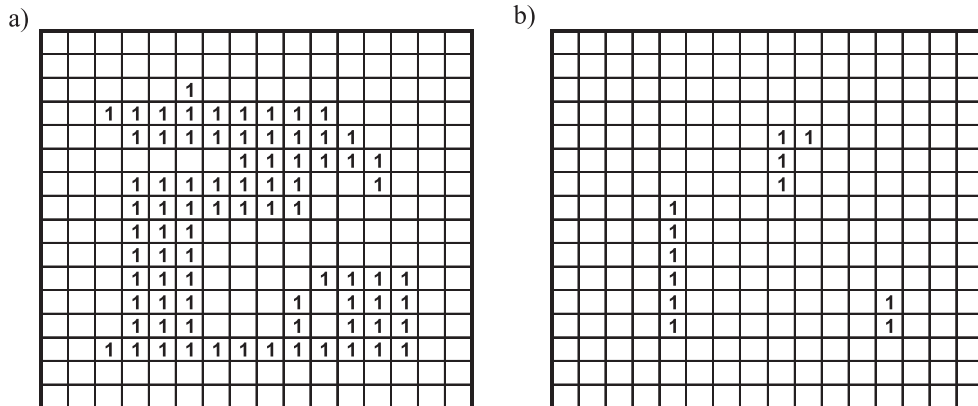
Operację erozji (z elementem strukturującym jak w przypadku dylacji) przedstawiono na rys. 9.

① Oznaczenie operacji dylacji symbolem „ \oplus ” pochodzi od nazwiska jej twórcy (Jean Serra) [7, 8].

② Operacja ekstensywna – operacja, której wynik jest nadzbiorem zbioru przed przekształceniem.

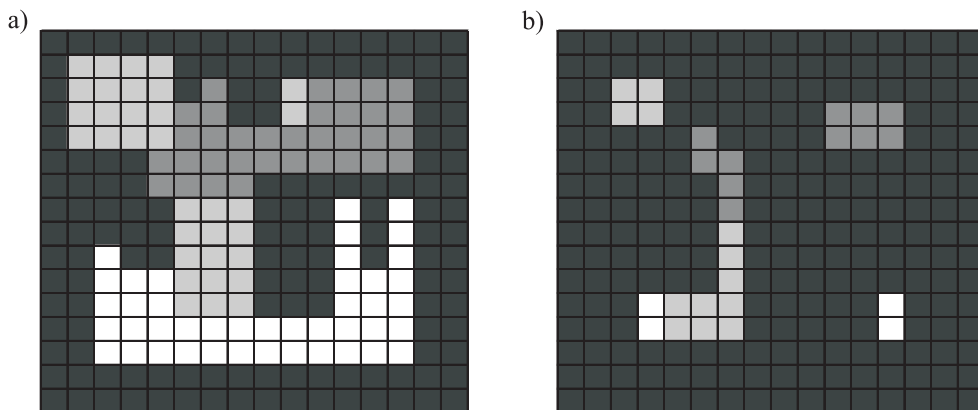
③ Monotoniczność – zachowanie zależności między zbiorami w stosunku do zależności przed wykonywaną operacją.

④ Zbiór powstały w wyniku dylacji zbioru przesuniętego jest równy przesuniętemu wynikowi dylacji zbioru oryginalnego.



Rys. 9. Przykład operacji erozji: a) obraz oryginalny; b) obraz przetworzony

Praktycznie erozję wyznacza się, przemieszczając element strukturalny B po wszystkich elementach obrazu A . Wartość elementu, dla którego jest wyznaczana erozja, jest wówczas iloczynem logicznym elementu strukturalnego B i części obrazu A , który jest nim przysłonięty. Erozję obrazów z gradacją poziomów szarości można traktować jako poszukiwanie lokalnego minimum, przy czym obszar tego przeszukiwania jest określony w analogiczny sposób, do obszaru przeszukiwania w przypadku dylacji. Ilustracją erozji obrazu o czterech poziomach szarości jest rys. 10.



Rys. 10. Operacja erozji obrazu z gradacją szarości: a) obraz oryginalny; b) obraz przetworzony

Erozja jest również fundamentalnym przekształceniem morfologicznym cechującym się niezmienniczością, monotonicznością oraz antyekestensywnością^① (gdy początek układu współrzędnych należy do zbioru B).

^① Operacja antyekestensywna – operacja, której wynik jest podzbiorem zbioru przed przekształceniem.

Złożone operacje morfologiczne

Operacje morfologiczne pierwszego rzędu

Operacje morfologiczne pierwszego rzędu nie są już tak elementarne jak wcześniej omówione operacje podstawowe. Stopień ich komplikacji jest nadal jednak niewielki. Działanie ich opiera się na złożeniu operacji podstawowych, przy czym liczba wystąpień tych operacji w złożeniu, ze względu na przejrzystość takiej klasyfikacji, ogranicza się do jednego [2]. Poniżej zostaną zdefiniowane wybrane operacje tej klasy, takie jak: gradient morfologiczny, otwarcie oraz domknięcie.

Gradient morfologiczny jest operacją, która powstaje jako różnica wyniku operacji dylacji i erozji:

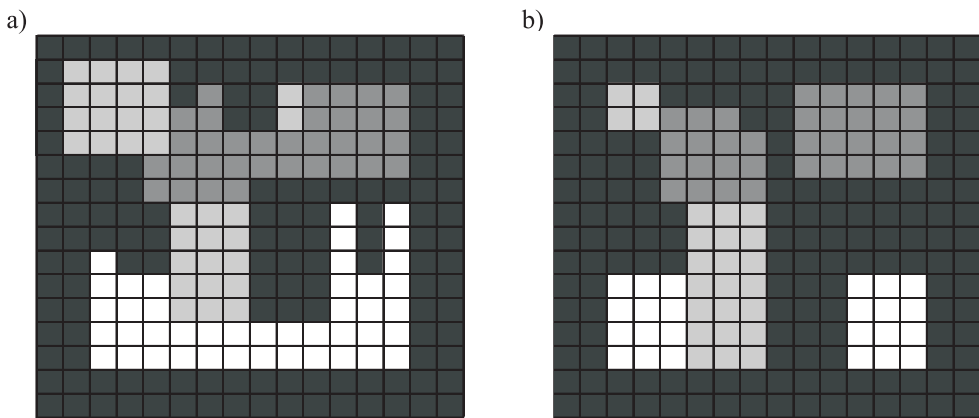
$$\gamma = (A \oplus_s B) - (A \ominus_s B). \quad (9)$$

Operacja ta jest stosowana głównie do wykrywania krawędzi występujących w obrazie [2].

Otwarcie dla klasy obrazów reprezentowanych zbiorami jest operacją powstającą przez złożenie najpierw operacji erozji, a następnie dylacji [1, 2, 6, 7]:

$$A \circ_s B = [A \ominus_s B] \oplus_s B. \quad (10)$$

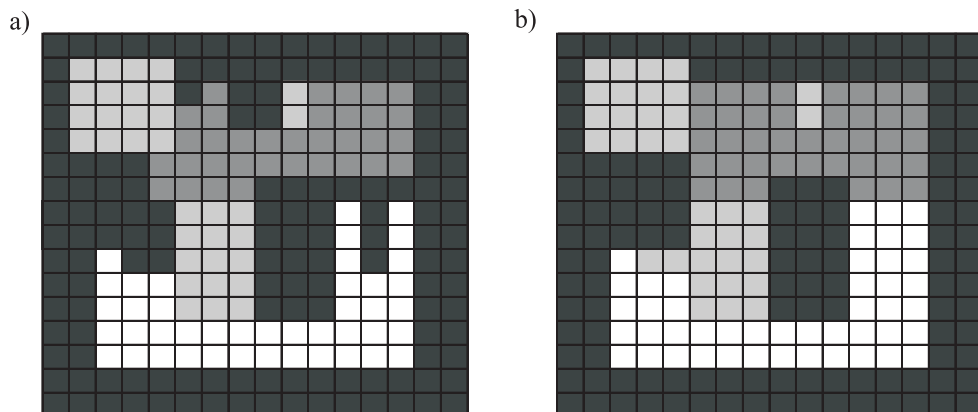
Przykład ilustrujący operację otwarcia dla obrazów z gradacją poziomów szarości podano na rys. 11. Jak wynika z tego rysunku, operacja otwarcia może być wykorzystywana do realizacji prostej segmentacji oraz eliminacji zakłóceń w postaci pojedynczych pikseli. Jednocześnie jednak są usuwane wąskie elementy obiektu. Otwarcie należy do fundamentalnych operacji morfologicznych stosowanych w przetwarzaniu obrazów.



Rys. 11. Operacja otwarcia dla obrazu z gradacją poziomów szarości: a) obraz przed operacją; b) obraz po operacji

Domknięcie [1, 2, 6, 7] jest operacją, która powstaje przez złożenie operacji dylacji, a następnie erozji:

$$A \bullet_s B = [A \oplus_s B] \ominus_s B. \quad (11)$$



Rys. 12. Domknięcie dla obrazów z gradacją szarości: a) obraz wejściowy; b) obraz wyjściowy

Działanie operacji domknięcia dla klasy obrazów z gradacją poziomów szarości pokazano na rys. 12.

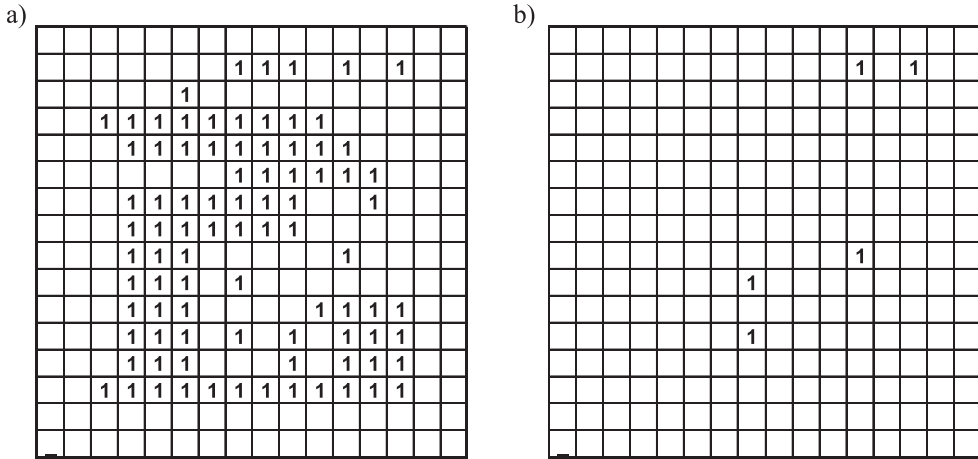
Operacje morfologiczne wyższych rzędów

Przez złożone operacje morfologiczne rozumie się operacje rzędu drugiego, trzeciego itd. Operacje morfologiczne drugiego rzędu charakteryzują się tym, że liczba wystąpień operacji podstawowych nie jest w zasadzie ograniczona, jednak powinna być określona w momencie startu algorytmu. Dla operacji morfologicznych trzeciego rzędu liczba wystąpień operacji podstawowych również nie jest ograniczona, a nawet nie jest określona w punkcie startowym algorytmu [2]. Jedną z operacji złożonych, najczęściej używanych w praktyce, jest „trafi – nie trafi” [1, 2, 4, 6, 7]. Jej cechą charakterystyczną jest możliwość oddziaływania jednocześnie zarówno na elementy zbioru A , obiektu, jak i na elementy zbioru A^C , stanowiącego tło obrazu. Definiując tę operację należy założyć, że dane są dwa elementy strukturalne $B_1, B_2 \subseteq E^d$ takie, że $B_1 \cap B_2 = \emptyset$. Operację „trafi nie trafi”, zwaną także *HMT* (*hit or miss*), opisuje się wyrażeniem:

$$A \otimes (B_1, B_2) = \{h \in E^d : B_{1h} \subseteq A \quad \text{oraz} \quad B_{2h} \subseteq A^C\}. \quad (12)$$

Definicję tę należy interpretować następująco: piksel h należy do zbioru $A \otimes (B_1, B_2)$, jeżeli przecięcie zbioru B_{1h} ze zbiorem A^C oraz przecięcie B_{2h} ze zbiorem A są zbiorami pustymi. Jeżeli warunek $B_1 \cap B_2 = \emptyset$ nie jest spełniony, to wynik operacji jest zawsze zbiorem pustym. Operacja *HMT* nadaje się szczególnie dobrze do wykrywania pewnych charakterystycznych punktów w obrazie, takich jak: narożniki, punkty krawędziowe, pewne obszary, czy też punkty izolowane w obrazie. Przykład operacji *HMT* wykrywającej punkty izolowane przedstawiono na rys. 13. W elemencie strukturalnym przez T oznaczono zbiór elementów należących do tła obrazu, natomiast przez N – zbiór elementów należących do obiektu:

$$(B_1, B_2) = \begin{bmatrix} T & T & T \\ T & N & T \\ T & T & T \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$



Rys. 13. Przykład operacji „trafi – nie trafi” do identyfikacji punktów izolowanych: a) obraz wejściowy; b) obraz po operacji

Z operacji *HMT* wywodzą się m.in. bardzo ważne złożone operacje morfologiczne, czyli pocienianie, pogrubianie, operacja znajdowania powłoki wypukłej oraz szkieletowania. Operacje te są opisane w [6, 7].

Algorytm dekompozycji kształtu obiektu

Opis teoretyczny

Dany jest obraz analizowany A oraz wypukły element strukturalny B , przy czym zakłada się, że ma on kształt koła. Zaczynając od dowolnego elementu obrazu A poszukuje się możliwie największego promienia r_1 , dla którego jest spełniony warunek:

$$A \wedge r_1 B \geq K, \quad (13)$$

definiując przy tym pierwsze przybliżenie kształtu obiektu jako:

$$A_1 = A \wedge r_1 B. \quad (14)$$

Następnie oblicza się największy promień $r_2 < r_1$, dla którego jest spełniona zależność:

$$(A \setminus A_1) \wedge r_2 B \geq K. \quad (15)$$

Przybliżenie drugiego kroku rozpatrywanego obiektu może być przedstawione w postaci wyrażenia:

$$A_2 = A_1 \cup (A \setminus A_1) \wedge r_2 B. \quad (16)$$

Dla dalszych przybliżeń postępowanie jest analogiczne. Algorytm ten jest rekursywny, a jego ogólne wyrażenie może być opisane wzorem:

$$\begin{cases} A_0 = \emptyset \\ A_{t+1} = A_t \cup (A \setminus A_t) \wedge r_{t+1}B \end{cases} \quad (17)$$

gdzie:

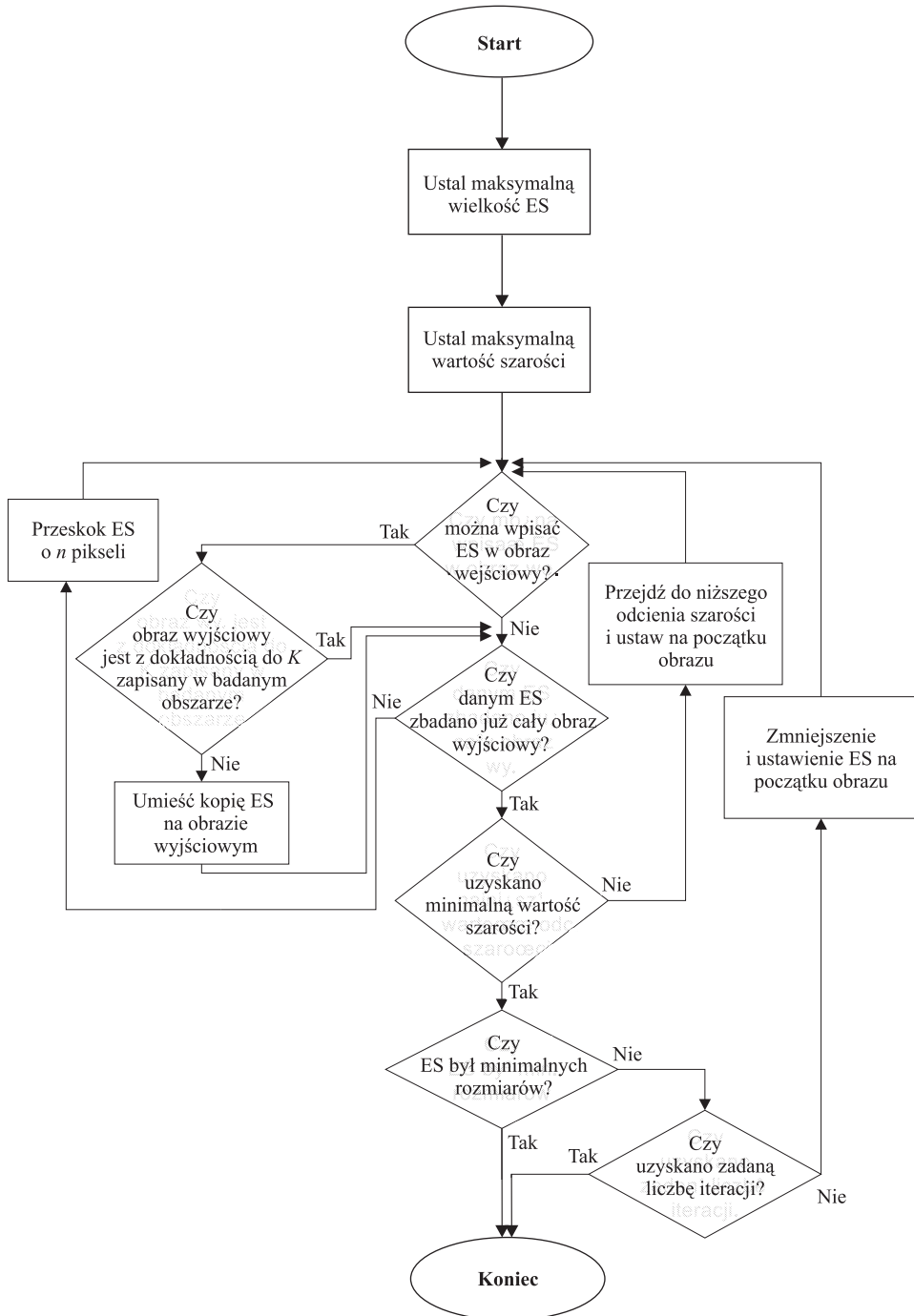
- r_t – maksymalny promień koła rB wpisanego w $A \setminus A_{t-1}$,
- K – współczynnik oznaczający wielkość obszaru,
- \wedge – iloczyn logiczny.

W szczególnym przypadku gdy $K = \emptyset$, algorytm będzie przedstawiał przypadek opisany w [2], w którym elementy odzwierciedlające obraz nie mogą się na siebie nakładać. Dla takiego przypadku zaprezentowano przykład funkcjonowania algorytmu.

Sieć działań

W celu opisanego algorytmu realizacji programowej dekompozycji kształtów obiektu obrazów z gradacją szarości przyjęto, że obraz dekomponowany jest zapisany w tablicy, natomiast na obraz zrekonstruowany jest przygotowana druga tablica. Obraz znajdujący się w tabeli wejściowej jest wynikiem zastosowania operacji redukcji poziomów szarości, dlatego liczba jego poziomów może wynosić od 2 do 255. Pierwszym etapem działania algorytmu jest wyznaczenie, w dalej opisany sposób, elementu strukturalnego ES o możliwie największych rozmiarach. Operacją wstępną jest również inicjacja wektora zmiennych odpowiadających występującym w obrazie odcieniom. Wektor ten został wyznaczony w procesie redukcji liczby poziomów jasności. Dalsza część algorytmu jest realizowana w sposób iteracyjny. W ramach każdej iteracji dekompozycja odbywa się dla wszystkich wyznaczonych w wektorze poziomów szarości, poczynając od poziomu szarości, który ma wartość największą. Sieć działań algorytmu, realizującego dekompozycję kształtu obiektu dla współczynnika K równego zero, przedstawiono na rys. 14.

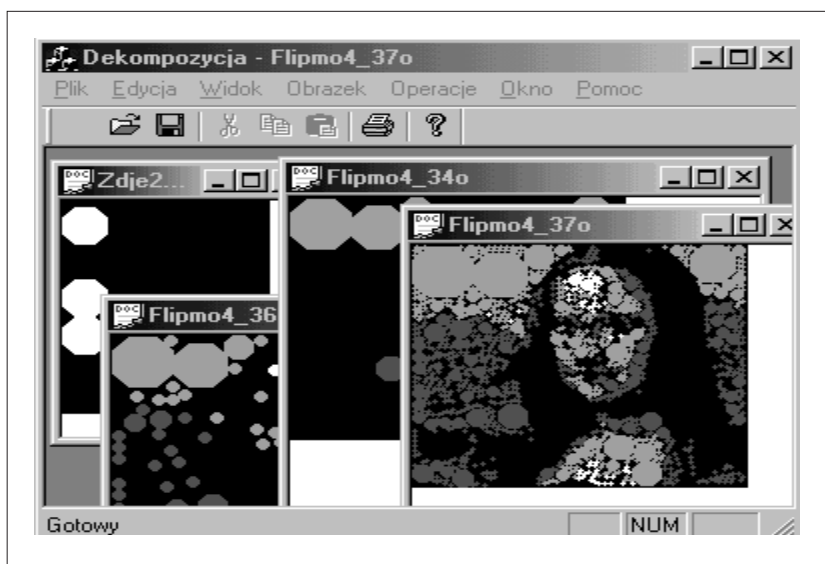
Maksymalny rozmiar elementu, za pomocą którego dokonuje się pierwszej iteracji dekompozycji obiektów obrazu wejściowego, jest ustalany na podstawie wymiarów analizowanego obrazu. Dla elementu kwadratowego wymiary te oblicza się w ten sposób, że maksymalna wielkość boku elementu jest równa (z dokładnością do jednego, dwóch lub trzech pikseli) szerokości albo wysokości obrazu – w zależności, która z tych wielkości jest mniejsza. Kolejne wielkości elementów są wyznaczone przez erozję elementu używanego w poprzedniej iteracji. Wyznaczenie rozmiaru maksymalnego elementu w kształcie ośmiokąta foremnego dokonuje się w podobny sposób jak w elemencie dziewięcioelementowym. Jednak z uwagi na kształt (uzyskanie różnych rozmiarów przy zachowaniu proporcji między bokami), sposób doboru wielkości pośrednich cechuje się swoistą osobliwością. Element podstawowy w kształcie krzyża 3×3 jest na stałe wpisany do tablicy kwadratowej, której wymiar ustala się na podstawie krótszego z wymiarów analizowanego obrazu. Następnie na elemencie tym dokonuje się dylacji na przemian z dodatkowymi elementami strukturalnymi w postaci krzyża i kwadratu 3×3 , z jednoczesnym przepisaniem do zarezerwowanej tablicy. Powiększany element, który ze względu na rozmiar może nie zmieścić się w zarezerwowanej tablicy, jest do niej zawsze wpisywany centralnie. Czynność ta zapewnia utrzymanie elementu w obszarze tablicy. Dylacje trwają na przemian aż do uzyskania maksymalnego, wykorzystywanego elementu strukturalnego – ośmiokąta, którego rozmiar (tzn. odległość w pionie lub poziomie między przeciwległymi bokami) jest równy lub mniejszy o jeden, dwa albo trzy piksele od rozmiarów tablicy. Każdy mniejszy rozmiar elementu jest wyznaczany przez operacje erozji dokonywane na przemian z elementami krzyżowymi lub prostokątnymi, przy czym jako pierwszy powinien być użyty ten, który był zastosowany jako ostatni do operacji dylacji do uzyskania ośmiokąta o największych rozmiarach.



Rys. 14. Algorytm dekompozycji obrazów z gradacją odcieni szarości realizowanej przez program

Opis reprezentacji programowej, realizującej dekompozycję kształtu obiektów na obrazie z gradacją szarości

Opisany algorytm zastosowano w aplikacji programowej, pracującej w środowisku Windows 95/98. Środowiskiem programistycznym, przy użyciu którego utworzono tę aplikację, jest produkt firmy Microsoft Visual C++ w wersji 6.0. Ze środowiskiem tym jest nieodzownie związana biblioteka MFC (*Microsoft Foundation Classes*), z której wykorzystano wiele funkcji dotyczących obsługi zdarzeń okna standardowego dla systemu Windows i jego oprogramowania. Interfejs użytkownika w tej aplikacji ma strukturę okna standardowego, z możliwością jednoczesnego otwarcia kilku plików (w formacie bitmap) – *Multiple Document Interface*. Ogólną postać aplikacji przedstawiono na rys. 15.



Rys. 15. Widok utworzonej aplikacji

Interfejs udostępnia także dynamiczne menu, które w zależności od tego czy został otwarty plik, może mieć różną postać. Najważniejsza z punktu widzenia pracy jest opcja menu *Operacje*, gdzie są udostępnione operacje *HitOrMiss* oraz *HitMiss*, realizujące operacje *HMT* przy użyciu elementu strukturalnego odpowiednio kwadratowego i ośmiokątnego. Dodatkowymi parametrami, potrzebnymi do wykonywania tych operacji, są: liczba poziomów szarości, do jakiej może być zredukowany obraz odcieniowoszary, oraz liczba iteracji, przekładająca się na wielkości najmniejszych elementów możliwych do rekonstrukcji obrazu wynikowego. Aplikacja realizuje również inne operacje, takie jak: negacja, erozja, dylacja, otwarcie i domknięcie.

Przykład funkcjonowania algorytmu

Działanie opisanego algorytmu zademonstrowano w prostym obrazie z odcieniami szarości, o wymiarach 168 x 148, pokazanym na rys. 16. Do dekompozycji zastosowano tu dwa opisane typy elementów strukturalnych, oznaczonych jako EK oraz EO. Analizowany obraz poddano wstępnie operacji redukcji poziomów szarości do 2, 4, 7 oraz 16.



Rys. 16. Obraz oryginalny

Wyniki dekompozycji uproszczonych obrazów przedstawiono na rys. 17 ÷ 20. Dla wymienionych reprezentacji na rys. 21 podano też wartość błędu średniokwadratowego, liczonego zgodnie z zależnością [13]:

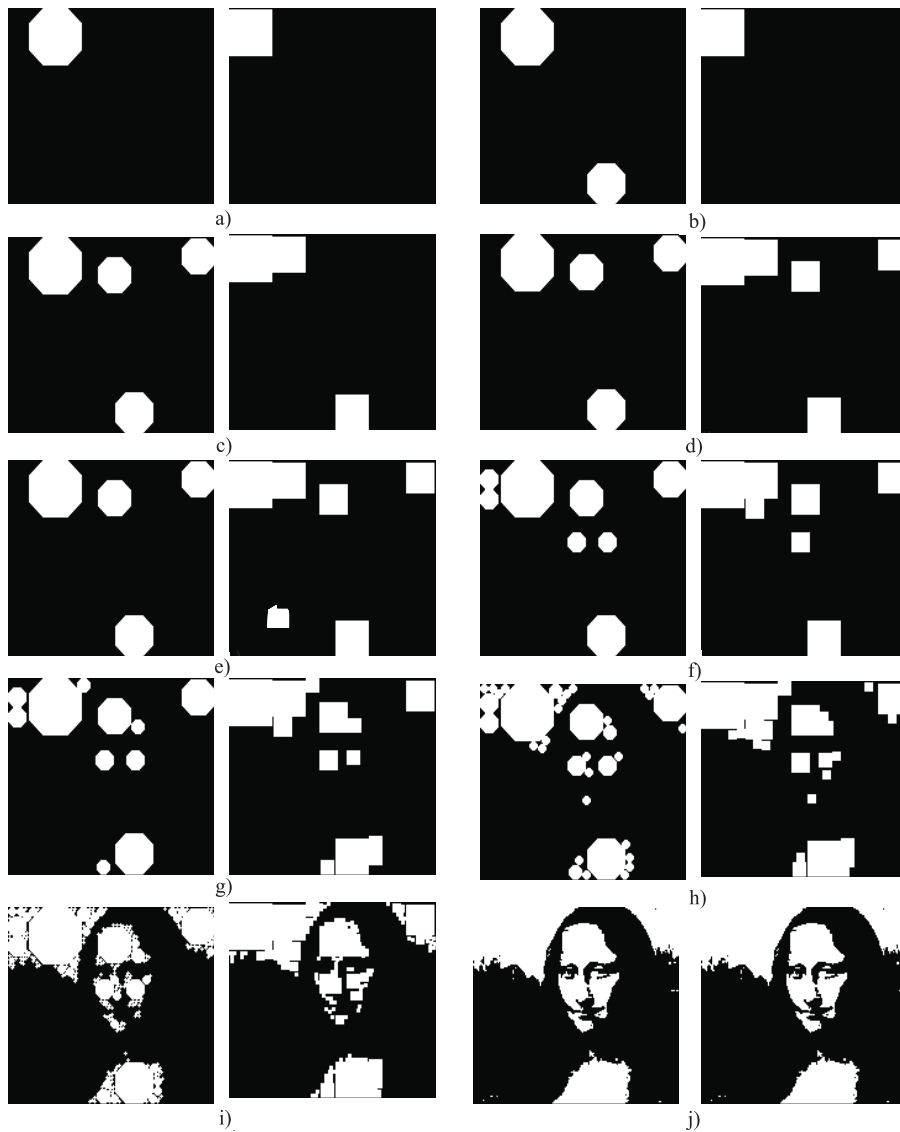
$$MSE = \frac{1}{MN} \sum_{x=1}^M \sum_{y=1}^N [f(x,y) - \varphi(x,y)]^2, \quad (18)$$

gdzie:

- f – funkcja opisująca wartości poziomów szarości obrazu oryginalnego,
- φ – funkcja opisująca wartości poziomów szarości obrazu rekonstruowanego,
- M, N – wymiary obrazów.

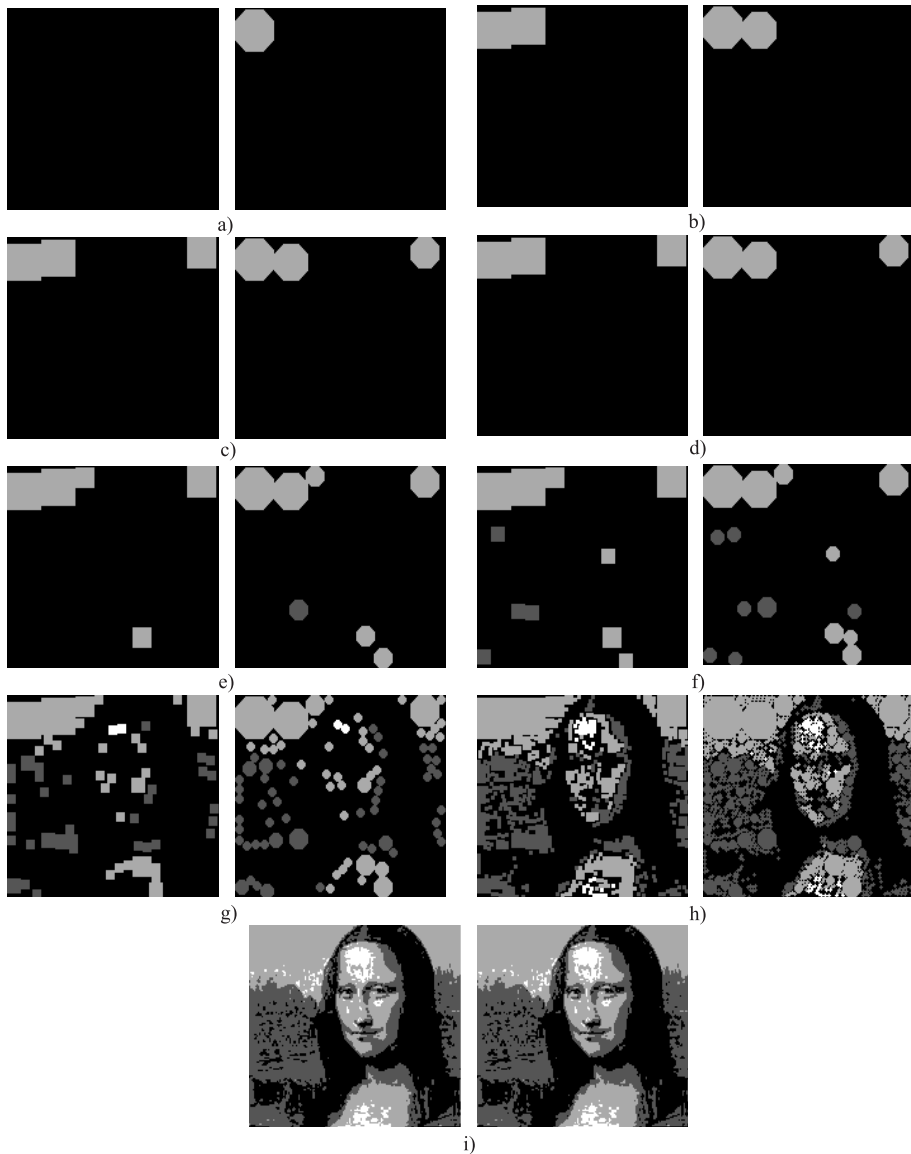
Wyznaczony błąd średniokwadratowy jest wynikiem dwóch składowych, z których jedna jest błędem wynikającym z zastosowania operacji redukcji poziomów szarości, druga – błędem operacji *HMT*. W zależności od specyfiki obrazu oraz zastosowanego elementu strukturalnego składowe te mogą się dodawać lub odejmować. Dla mniejszej liczby poziomów szarości wystąpią większe przedziały, do których będą należeć wartości poziomów szarości poszczególnych pikseli. Wobec tego istnieje większe prawdopodobieństwo, że piksele o różnych wartościach poziomów szarości zostaną zakwalifikowane do jednego przedziału. Spowoduje to powstawanie większych obszarów i w konsekwencji możliwość przybliżania obiektu odpowiednio większymi elementami. Wynik takich operacji jest jednak obciążony dużo większym błędem, wynikającym z operacji redukcji liczby poziomów szarości.

Jak wynika z przedstawionych rezultatów oraz zastosowania algorytmu dla innych obrazów, zdecydowaną większość obrazów rzeczywistych znacznie lepiej rekonstruuje się, stosując element w kształcie



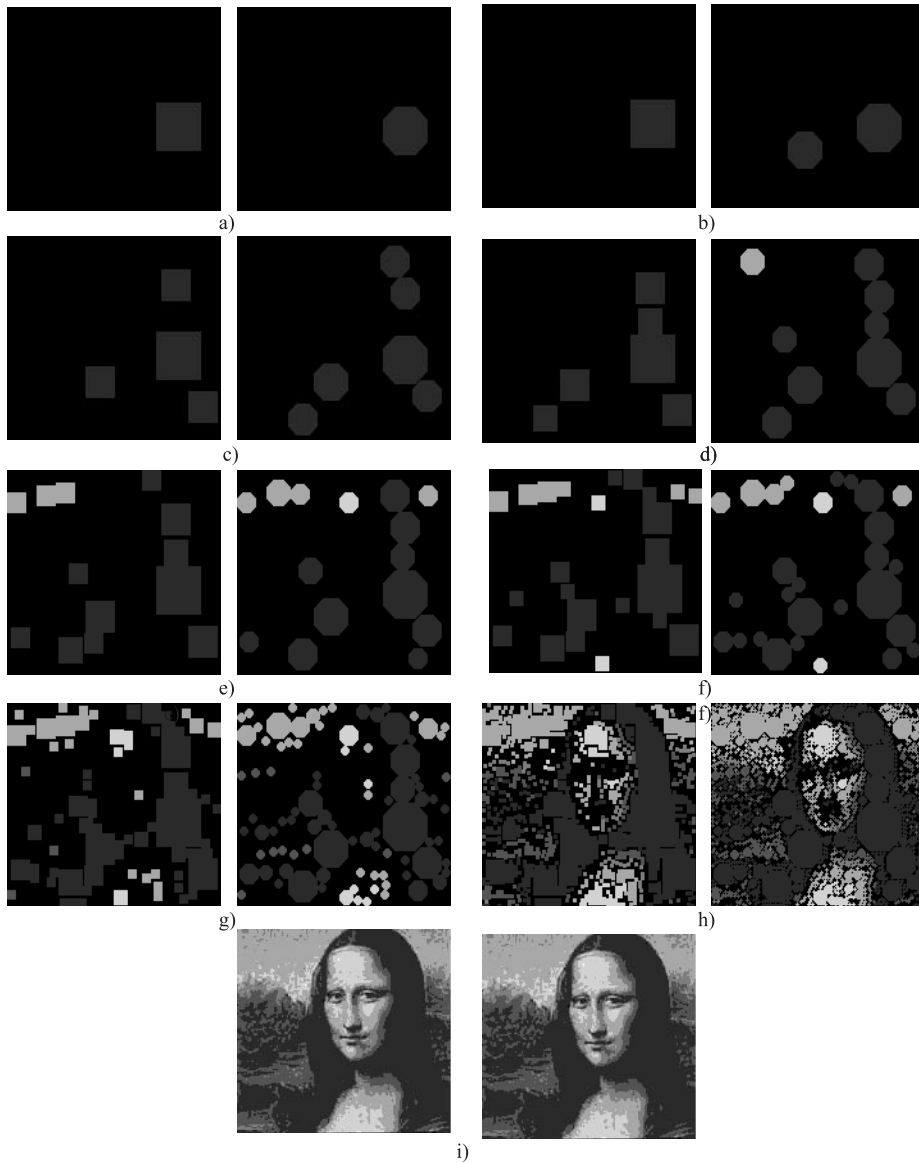
Rys. 17. Ilustracja wyników dla dekompozycji z redukcją do 2 poziomów szarości; liczba iteracji odpowiednio: a) 29; b) 30; c) 31; d) 32; e) 33; f) 34; g) 35; h) 36; i) 37; j) 38

ośmiokąta foremnego. Przemawiają za tym głównie względy wynikające ze specyfiki percepcji wzroku ludzkiego, dla którego znacznie ważniejszy przy odtworzeniu obiektu z obrazu jest zarys jego kształtu. W obrazach rzeczywistych kształt jest zdecydowanie lepiej odtwarzany z wykorzystaniem elementów, które jak najmniej różnią się od koła. Elementem o bardzo dobrych właściwościach jest element



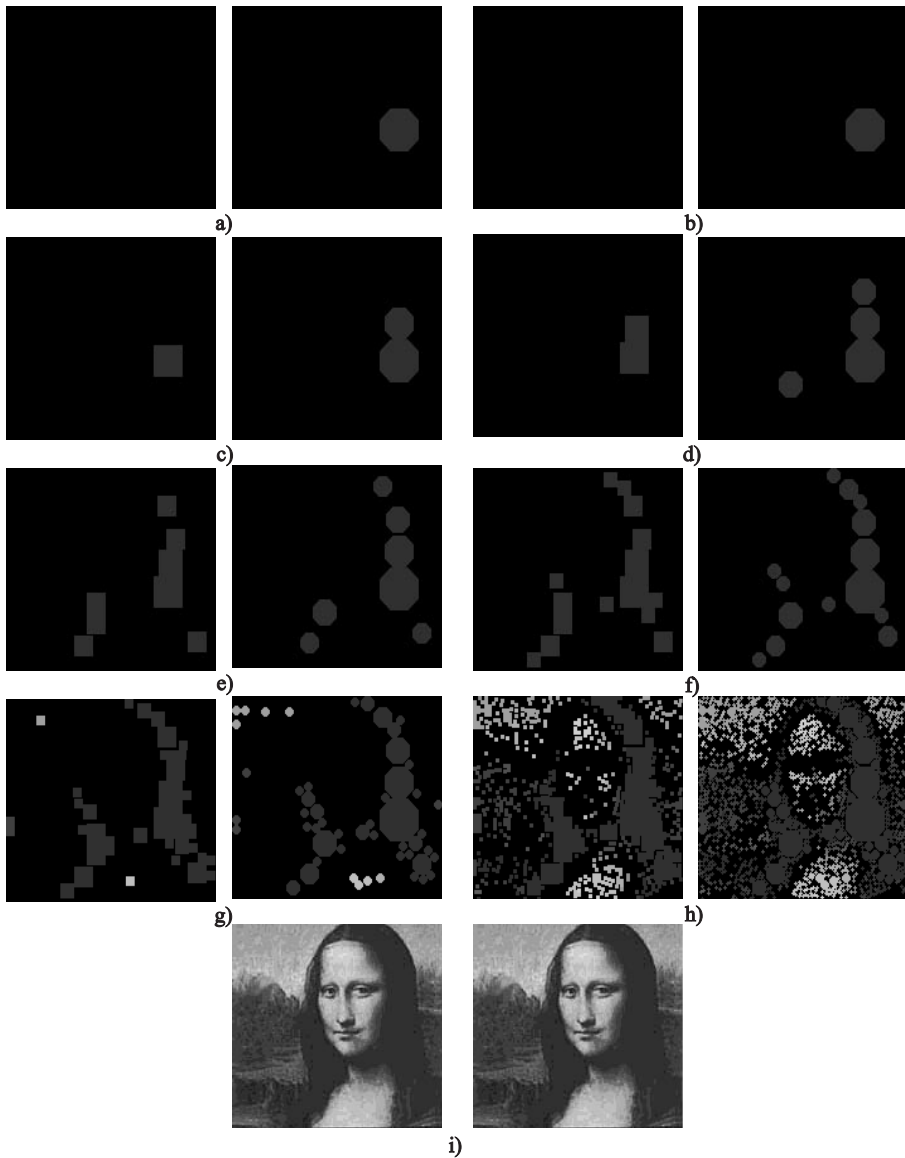
Rys. 18. Ilustracja wyników dla dekompozycji z redukcją do 4 poziomów szarości; liczba iteracji odpowiednio: a) 30; b) 31; c) 32; d) 33; e) 34; f) 35; g) 36; h) 37; i) 38

w kształcie sześciokąta foremnego. Świadczą o tym jego zalety wymienione w części dotyczącej doboru optymalnego elementu strukturalnego. Element ośmiokątny ma właściwości zbliżone do sześciokątnego. Istotną jednak zaletą elementu ośmiokątnego jest większa możliwość zastosowania do obrazów bitmap, które są opisane w klasycznej siatce kwadratowej. Zasadniczą jego wadą jest brak



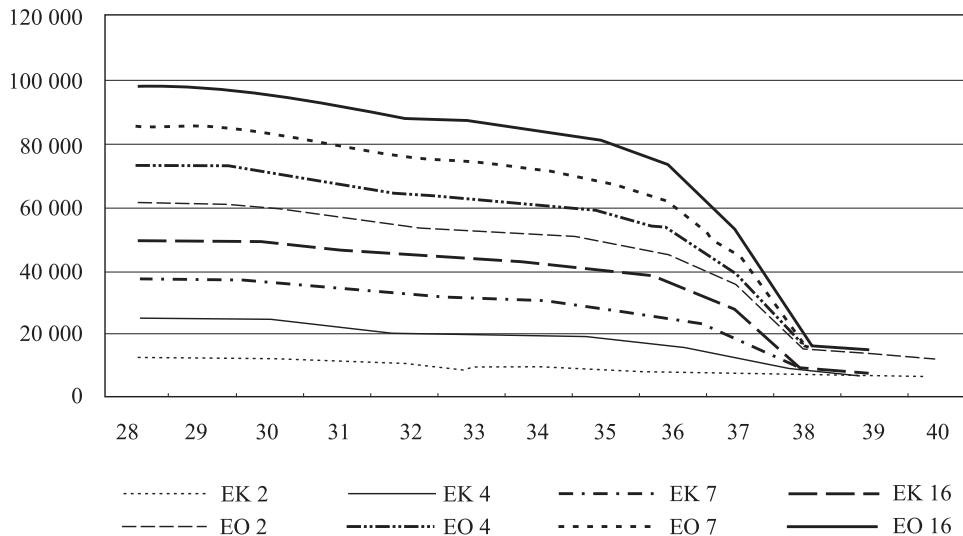
Rys. 19. Ilustracja wyników dla dekompozycji z redukcją do 7 poziomów szarości; liczba iteracji odpowiednio: a) 30; b) 31; c) 32; d) 33; e) 34; f) 35; g) 36; h) 37; i) 38

całkowitego pokrycia jednolitego obszaru z wykorzystaniem ośmiokątów. Klasą obrazów, w której zastosowanie kwadratowego elementu strukturalnego przynosi lepsze rezultaty dekompozycji, są obrazy z obiektami mającymi różnego rodzaju narożniki, długie odcinki linii prostych, czyli plany miast, zdjęcia obwodów drukowanych itp.



Rys. 20. Ilustracja wyników dla dekompozycji z redukcją do 16 poziomów szarości; liczba iteracji odpowiednio: a) 30; b) 31; c) 32; d) 33; e) 34; f) 35; g) 36; h) 37; i) 38

Dość interesujące może okazać się stosowanie do rekonstrukcji obrazów zestawu elementów strukturalnych, o różnych kształtach, co umożliwia ich dokładne łączenie się między sobą. Jednakże przy takim podejściu pojawiają się problemy z implementacją programową oraz z odpowiednio dłuższym wykonywaniem operacji z uwagą na większą złożoność algorytmu.



Rys. 21. Błąd średniokwadratowy w zależności od liczby iteracji dekompozycji

EK – element strukturalny kwadratowy; *EO* – element strukturalny ośmiokątny; 2, 4, 7, 16 – liczba poziomów szarości

Kolejnym krokiem zmierzającym do poprawy jakości dekompozycji może być dopuszczenie możliwości nakładania się na siebie elementów rekonstruujących obraz, co w algorytmie oznacza niezerową wartość współczynnika K . Przy takim podejściu pojawia się problem związany ze specyfiką dozwolonego sposobu nakładania się na siebie elementów. W poszczególnych sytuacjach elementy mogłyby się pokrywać całymi obszarami. W takim przypadku rekonstrukcja obszarów byłaby bardzo bliska uzyskiwaniu tego obrazu z pojedynczych pikseli. Istotny problem wystąpić może również w przypadku możliwości nakładania się obszarów o różnych poziomach szarości. Działanie takie może prowadzić do dodatkowego zniekształcenia kształtów obiektów obrazu.

Na poprawę jakości analizowanej operacji może wpływać sposób „przemieszczania” się elementu strukturalnego po obrazie poddawanym dekompozycji. Omówiony algorytm zakłada bowiem, że przepisanie obszaru obrazu odbywa się natychmiast po spełnieniu warunków „wpasowania” się w niego elementu strukturalnego, co może uniemożliwić wpisanie elementów obiektu dla innych, lepszych z punktu widzenia pokrywanego obszaru, położeń elementu strukturalnego. Jest to związane z możliwością dokonywania optymalizacji rekonstrukcji przez czas trwania kilku iteracji. Dodatkowo może zdarzyć się sytuacja, w której wpisany we wcześniejszej iteracji duży element uniemożliwi rekonstrukcję obrazu przez większą liczbę elementów średniej wielkości, które mogłyby efektywniej dokonać rekonstrukcji obrazu w dalszej iteracji. Obok dużego elementu będą mogły się zmieścić dużo mniejsze elementy, które będą stosowane w dalszych iteracjach w stosunku do elementów średniej wielkości. To automatycznie spowoduje uzyskanie odpowiedniej jakości dopiero po wykonaniu większej liczby iteracji. Buforowanie wyników pośrednich wiąże się z bardzo dużymi zasobami pamięci komputera oraz z podejmowaniem decyzji, dotyczącej liczby zapamiętywanych wyników pośrednich.

Dodatkową komplikacją zagadnień dekompozycji kształtów obiektów jest rozszerzenie algorytmu na obrazy barwne. Towarzyszy temu jednak wzrost złożoności niektórych metod morfologii matematycznej, implikujący wysokie wymagania na moc obliczeniową i zasoby pamięciowe komputera. Pomimo tych trudności, przekształcenia morfologii matematycznej, ze względu na specyficzny hierarchiczny układ operacji tworzących szkielet, który można wypełniać, dysponując wieloma stopniami swobody, niosą ze sobą ogromny potencjał możliwości, który jest zachętą do prowadzenia badań w tym zakresie przetwarzania obrazów. Stopień komplikacji i wynik działania nowo powstającej operacji zależą od zastosowanych kombinacji operacji stanowiących ten szkielet.

Potencjalne zastosowania

Jak wynika z przedstawionych rezultatów, aby dokonać detekcji obiektu nie jest konieczne wierne odzwierciedlenie oryginalnego obrazu. Zasadnicze znaczenie w procesie detekcji obiektu obrazu należy przypisać odpowiedniemu układowi decyzyjnemu, stwierdzającemu czy analizowany obszar obrazu jest reprezentacją obiektu.

Opisana metoda dekompozycji kształtów obiektów może być zastosowana w systemach ochrony mienia do detekcji występowania i zmian położenia obiektów. Specyfika tych systemów polega bowiem na analizowaniu obrazu przez układ znajdujący się przy kamerze. W przypadku braku rejestracji zmian w obrazie do punktu dowodzenia nie są przekazywane żadne informacje z wyjątkiem zarządzania. Dopiero wykrycie zmian w obrazie rejestrowanym przez kamerę może powodować inicjację rejestracji lub/oraz przekaz sekwencji obrazów do punktu dowodzenia. Taki sposób działania umożliwia alarmowanie punktu dowodzenia systemu, zebranie materiału dowodowego w przypadku wykrycia włamania oraz uzyskanie oszczędności zasobów systemu, ponieważ przekaz obrazu lub jego rejestracja występuje tylko w przypadku zaistnienia sytuacji alarmowej. Margines błędu, przy którym układ nie zareaguje wykryciem sytuacji alarmowej, zależy od samego układu decyzyjnego, a także od klasy obrazu (co wynika m.in. z wielkości obszarów chronionych) oraz jego obiektów, czyli zastosowanego powiększenia. Powinien on być określony w sposób uwzględniający specyfikę analizowanego obszaru.

Drugim z potencjalnych zastosowań zaprezentowanej metody jest układ identyfikacji przedmiotów znajdujących się na przesuwnikach taśmowych w pomieszczeniach przemysłowych, sklepowych itp., w systemach sortowania asortymentu. Zastosowanie takie wydaje się szczególnie ważne teraz, gdy obserwuje się tendencję do skracania serii produkowanych dóbr, a nawet wykonywanie różniących się egzemplarzy w ramach jednej serii produkcyjnej.

W wymienionych zastosowaniach metody dekompozycji kształtów obiektów wystarczająca może okazać się analiza tylko części obrazu (przykładowo w układach ochrony pomieszczeń najbardziej interesującymi obszarami obrazu są te jego części, które odzwierciedlają drzwi i okna). Jest to jedna z cech przekształceń morfologicznych, wyróżniająca je spośród innych metod przetwarzania obrazów, dająca możliwości zmniejszenia liczby obliczeń systemu.

Problemem wymagającym wciąż rozwiązania jest poszukiwanie odpowiednich elementów strukturalnych, a także dobór wielkości obszarów, w jakich elementy strukturalne mogą się pokrywać, co odpowiada znalezieniu odpowiedniej wartości dla parametru K w przedstawionym algorytmie dekompozycji. W zależności od tych czynników dekompozycja kształtów obiektów oraz detekcja obiektów w indywidualnych przypadkach będzie realizowana z większą lub mniejszą dokładnością. Nadal otwarte zostaje również poszukiwanie odpowiednich, dla określonych zastosowań, układów decyzyjnych, kwalifikujących, które z przybliżeń należy traktować jako obiekty rzeczywiste.

Bibliografia

- [1] Ghosh P. K.: *A unified computational framework for Minkowski operations*. Computer and Graphics, 1993, vol. 17, s. 357–378
- [2] Kukiełka G., Woźnicki J.: *Praktyczne aspekty wykorzystania metod morfologii matematycznej w cyfrowym przetwarzaniu obrazów*. W: Materiały konferencyjne z III sympozjum naukowego „Techniki przetwarzania obrazu”, Serock, 1997, s. 54–61
- [3] Malina W., Ablameyko S., Pawlak W.: *Podstawy cyfrowego przetwarzania obrazów*. Warszawa, Akademicka Oficyna Wydawnicza EXIT, 2002
- [4] Maragos P.: *Morphological systems: slope transforms and max-min difference and differential equations*. Signal Processing, 1994, vol. 38, s. 57–77
- [5] Maragos P.: *Slope transforms: theory and application to nonlinear signal processing*. IEEE Trans. Signal Proc., 1995, vol. 43, s. 864–877
- [6] Nieniewski M.: *Morfologia matematyczna w przetwarzaniu obrazów*. Warszawa, Akademicka Oficyna Wydawnicza PLJ, 1998
- [7] Serra J.: *Image Analysis and Mathematical Morphology*. London, Academic Press, 1988, vol. 1
- [8] Serra J.: *Image Analysis and Mathematical Morphology*. London, Academic Press, 1989, vol. 2 (Theoretical advances)
- [9] Śluzek A.: *Komputerowa analiza obrazów*. Warszawa, Wydawnictwa Politechniki Warszawskiej, 1991
- [10] Tadeusiewicz R., Flasiński M.: *Rozpoznawanie obrazów*. Seria: Współczesna Nauka i Technika. Informatyka. Warszawa, PWN, 1991
- [11] Tadeusiewicz R., Korohoda P.: *Komputerowa analiza i przetwarzanie obrazów*. Kraków, Wydawnictwo Fundacji Postępu Telekomunikacji, 1997
- [12] Wojciechowski K.: *Rozpoznawanie obrazów*. Gliwice, Politechnika Śląska, 1997
- [13] Woźnicki J.: *Podstawowe techniki przetwarzania obrazu*. Warszawa, WKŁ, 1996

Grzegorz Szczurek



Mgr inż. Grzegorz Szczurek (1975) – absolwent Wydziału Elektroniki i Technik Informatycznych Politechniki Warszawskiej (2001); pracownik Instytutu Łączności w Warszawie (od 2000); zainteresowania: cyfrowe przetwarzanie sygnałów, analiza i rozpoznawanie obrazów, sieci dostępowe (DPL/PLC).
e-mail: G.Szczurek@itl.waw.pl