

231/2006

Raport Badawczy

RB/45/2006

Research Report

Obrazowa baza danych

Tatiana Jaworska

Andrzej Partyka

Instytut Badań Systemowych
Polska Akademia Nauk

Systems Research Institute
Polish Academy of Sciences



POLSKA AKADEMIA NAUK

Instytut Badań Systemowych

ul. Newelska 6

01-447 Warszawa

tel.: (+48) (22) 8373578

fax: (+48) (22) 8372772

Kierownik Pracowni zgłaszający pracę:
Prof. dr hab. inż. Janusz Kacprzyk

Warszawa 2006

INSTYTUT BADAŃ SYSTEMOWYCH PAN
PRACOWNIA SYSTEMÓW INTELIGENTNYCH

Tatiana Jaworska
Andrzej Partyka

Obrazowa baza danych

Zadanie: A3.1.3/2006

Inteligentne metody analizy danych i odkrywania wiedzy dla
potrzeb wspomagania decyzji

Warszawa, grudzień 2006

Spis treści

1. Wstęp	6
2. Ogólna charakterystyka obrazowych bazy danych	7
3. Wstępna obróbka obrazu	11
3.1. Modyfikacja histogramu	11
3.2. Wydzielanie krawędzi	13
3.3. Operacje morfologiczne	16
3.4. Segmentacja	20
3.4.1. Segmentacja oparta na histogramie bimodalnym	20
3.4.2. Segmentacja oparta na analizie skupień	21
3.4.3. Segmentacja oparta na przestrzeni kolorów RGB	25
3.4.4. Segmentacja oparta na rozmytej analizie skupień	30
3.5. Wydzielenie obiektów z tła	31
4. Model danych	35
4.1. Ilościowe modele danych	36
4.1.1. Kolor średni	37
4.1.2. Analiza tekstur	37
4.1.3. Deskryptory kształtu	52
4.2. Przestrzenne modele danych	60
5. Struktura bazy danych	62
6. Indeksowanie danych	66
6.1. Filtrowanie	66
6.2. R-drzewa	67
6.3. Indeksowanie w przestrzeni transformat	69
6.3.1. Indeksowanie w oparciu o θ -transformację	69
6.3.2. Indeksowanie w oparciu o transformację falkową	70
7. Przeszukiwanie danych – zapytania do bazy danych	71
7.1. Miary oparte na metrykach	72
7.2. Rozmyte miary podobieństwa	73
7.3. Miary oparte na łącznym rozkładzie prawdopodobieństwa	75
8. Instrukcja obsługi użytkownika	76
9. Matematyczne podstawy przetwarzania obrazu – metody stosowane w tym opracowaniu	80

9.1. Analiza skupień.....	80
9.2. Segmentacja oparty na progowaniu rozmytym.....	81
9.3. Rozmyta analiza skupień.....	84
9.4. Metoda wektorów wspierających SVM.....	86
9.5. Metoda aktywnych konturów.....	89
9.6. Transformacja falkowa.....	91
10. Dokumentacja techniczna programu.....	94
10.1. Potrzebne pliki i ich lokalizacja	94
10.2. Sposób wybierania koloru.....	95
10.3. Konfiguracja oprogramowania	95
11. Podsumowanie.....	97
12. Bibliografia.....	98
13. Spis ilustracji.....	103

1. Wstęp

Świat obrazów otacza nas od urodzenia. Świat komputerowych obrazów również zaczyna coraz częściej wchodzić w nasze życie. Zalew obrazami cyfrowymi postępuje lawinowo wraz z rozwojem coraz szybszych i pojemniejszych łączy internetowych. Wraz ze zwiększaniem się ilości obrazów następuje konieczność ich magazynowania w bazach danych, przetwarzania, rozpoznawania i wyszukiwania. Coraz częściej zaczynają powstawać duże bazy obrazowe. W celu sprawnego przeszukiwania tych baz trzeba stworzyć efektywne mechanizmy indeksowania obrazów, a później wyszukiwania poszczególnych obrazów lub wręcz określonej informacji na pojedynczych obrazach. Opis sposobu tworzenia takiej bazy i budowania odpowiednich mechanizmów do jej przeszukiwania jest celem niniejszego opracowania.

Obrazowe bazy danych różnią się między sobą w zależności od rodzaju przechowywanych obrazów i celów do jakich mają służyć. Inaczej będzie zatem konstruowana baza obrazów dla galerii i muzeów, a zupełnie inaczej jeśli będzie to system informacji geograficznej (GIS) lub kolekcja zdjęć satelitarnych do celów geologicznych lub prognoz pogody. Jeszcze inne mechanizmy przeszukiwania będą wykorzystywane w policyjnych bazach odcisków palców i zdjęć twarzy przestępców, a inne przy obrazach kosmosu dla astronomów.

W naszym opracowaniu zajmujemy się budową bazy zdjęć nieruchomości. W tym celu wykorzystywane są zdjęcia domów jednorodzinnych pobrane z Internetu w formacie JPEG. Następnie przedstawiona jest struktura bazy danych i procedury analizy obrazów w celu wyselekcjonowania elementów architektonicznych na zgromadzonych zdjęciach domów. Każdy rekord bazy poza całym obrazem zawiera elementy architektoniczne i ich cechy wyekstrahowanymi z danego obrazu oraz dane o lokalizacji elementów, a nawet pewne dane tekstowe. Baza danych i mechanizmy analizy obrazu są zaimplementowane z pomocą pakietu MatLab (wersja R2006b) ze wspomaganiami następujących ToolBoxów: ImageProcessing, Fuzzy, Statistics, Wavelet i DataBase. Baza danych jest dostępna w trybie interaktywnym przez Internet. Interface internetowy jest wykonany w Perlu. Internauci nie działają bezpośrednio na MatLabie, a jedynie uruchamiają go *off-line* poprzez interface strony WWW.

2. Ogólna charakterystyka obrazowych bazy danych

Budowa obrazowej bazy danych wymaga implementacji podobnych operacji jak tekstowa baza danych, czyli indeksowania, przeszukiwania, porównywania, formułowania zapytań itp., ale przy użyciu trochę innych mechanizmów wewnętrznych umożliwiających wykonywanie tych wszystkich operacji na obrazach. Poniżej zostaną chronologicznie przedstawione etapy rozwoju baz obrazowych, podstawowe operacje wykonywane na tych bazach danych oraz zostaną omówione dotychczas stosowane algorytmy.

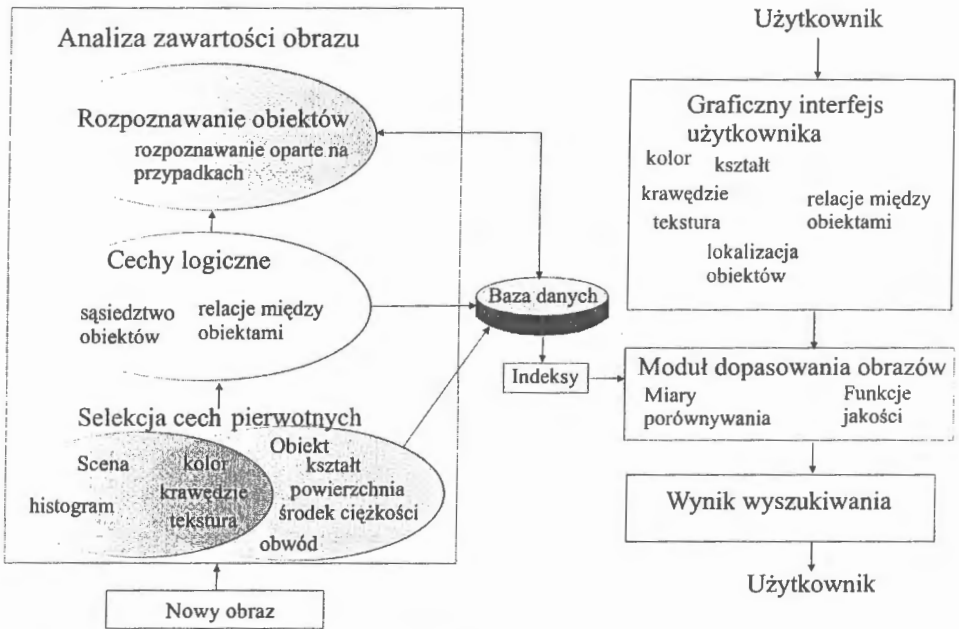
Historycznie jako pierwsze pojawiło się atrybutowe podejście do obrazowych bazach danych [18], [26], w którym do obrazu dodawano ręcznie zbiór atrybutów dotyczących zawartości obrazu. Następnie baza danych była indeksowana i przeszukiwana według tych atrybutów. Podejście takie sprowadzało się *de facto* do przetwarzania danych tekstowych z dołączonym jedynie obrazem. W przypadku gdy obrazy były pobierane z Internetu i podejmowano próby rozpoznawania wszystkich rodzajów obrazów wykorzystywano nazwy obrazów do preklasyfikacji [49], [62].

Następne chronologicznie podejście było oparte na analizie informacji, którą można było uzyskać z pojedynczych pikseli, czyli kolorze, kształcie, itp. Taką bazę na uniwersytecie w Berkeley, o nazwie *Chabot*, stworzyli w latach 90-tych V. Ogle i M. Stonebreak [Ogle 25]. Natomiast w IBM Almaden Research Centra Flickner wraz ze współpracownikami [Flickner 10], [46] opracowali do swojej bazy specjalny interfejs użytkownika umożliwiający zadawanie pytań w formie graficznej, ale nadal w oparciu o wyżej wspomniane cechy. Dodatkową wadą tej metody był fakt, że zapytania dotyczyły całego obrazu. Następnym podejściem poprawiającym sposób przeszukiwania zgromadzonych w bazie obrazów była możliwość znalezienia obiektu będącego na pierwszym planie w oparciu o kolor lub kształt [60], [25]. W tym nurcie utrzymana jest też baza ryb i innych zwierząt morskich wykonana przez Abbasi'ego i Kittlera na Uniwersytecie w Suvrey w Wielkiej Brytanii [44]. Podejście to było jednak ciągle wąskim gardłem w szybko rozwijających się systemach multimedialnych, ponieważ istnieje ogromny rozróżnienie pomiędzy opisem obrazu na niskim poziomie abstrakcji, na którym pracuje komputer i postrzeganiem obrazu przez człowieka na wysokim poziomie abstrakcji.

Obecnie powstające obrazowe bazy danych, które opierają się na automatycznym rozpoznawaniu treści obrazu, próbują w jak największym stopniu zniwelować ten rozróżnienie poprzez wprowadzenie wielopoziomowej analizy treści obrazu. W tym podejściu bierze się

pod uwagę od trzech do pięciu poziomów abstrakcji na których opisuje się obraz. W 1999 r. Castelli [9] zastosował pięciopoziomową strukturę analizy treści obrazu. Wydzielił on następujące poziomy: poziom regionów, poziom postrzeganych regionów, poziom elementów obiektów, całych obiektów i poziom sceny. Były też próby stworzenia sprzężenia zwrotnego pomiędzy zapytaniami do bazy, z strukturą cech [71], [53].

Obiecujące wyniki w oparciu o hierarchiczną strukturę uzyskali już Jaimes [28] i Hong [24], którzy wydziłili trzy poziomy analizy obrazu. W przypadku naszej bazy, której schemat ideowy jest pokazany na rys 1, też opieramy się na podobnych założeniach.



Rysunek 1 Ogólny schemat obrazowej bazy danych

Najniższym poziomem analizy obrazu jest selekcja cech pierwotnych czyli tych, które są uzyskiwane bezpośrednio z obrazu, jak np.: kolor, tekstura, kształt, powierzchnia, itp. Na podstawie tych cech próbuje się wydzielić obiekty z obrazu, które będą elementami analizowanymi na wyższych poziomach. W tym celu prowadzi się obróbkę wstępną obrazu dołączanego do bazy. Obróbka wstępna może obejmować wszystkie lub tylko niektóre z poniższych operacji:

- modyfikację histogramu,
- odsumienie lub inne techniki poprawiania jakości obrazu,
- segmentację obrazu,
- wydzielenie krawędzi,
- wydzielenie obiektów w obrazie.

Przy tworzeniu obrazowej bazy danych trzeba określić jakie cechy są z obrazu ekstrahowane, aby w możliwie najpełniejszy sposób móc potem korzystać ze zgromadzonych zasobów, a jednocześnie nie przechowywać danych redundantnych. Najczęściej wykorzystywane cechy pierwotne obiektu to:

- średni kolor
- parametry tekstury
- deskryptory kształtu

Na drugim poziomie analizy obrazu znajdują się cechy logiczne, które nie są już bezpośrednio związane z pikselami. Na tym poziomie posługujemy się już obiektami wydzielonymi na pierwszym poziomie, takimi jak: trójkąt, owal, prostokąt, ale jeszcze nie przyporządkowujemy im deskryptorów semantycznych. Aby sprawnie posługiwać się tymi cechami przechodzi się na tym poziomie do wektorowego opisu danych posługując się pojęciami punktu, odcinka i poligonu. Na cechy logiczne składają się:

- relacje przestrzenne poszczególnych obiektów,
- rozmieszczenie obiektów w całym obrazie,
- położenie obiektów względem siebie,
- relacje sąsiedztwa,
- relacje zawierania,
- relacje przecinania, itp.

Na najwyższym poziomie analizy obrazu posługujemy się elementami semantycznymi w rozumieniu użytkownika, takimi jak: domy, okna, dachy, samochody, drzewa itp. Na tym poziomie najłatwiej jest też użytkownikowi kierować zapytania do systemu.

Przy opisie obrazów poza cechami na każdym poziomie wykorzystuje się też atrybuty tekstowe np.:

- nazwę obrazu
- datę uzyskania
- właściciela lub inne cechy pochodzenia

- powiązanie z innymi obrazami (film video)
- przyporządkowaną kategorię

co łącznie tworzy meta-dane przechowywane w bazie.

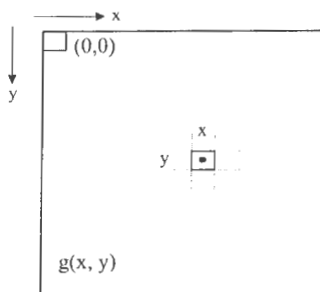
W momencie dołączania nowego obrazu do bazy następuje półautomatyczna, a w bardziej zaawansowanych wersjach automatyczna analiza cech na każdym poziomie abstrakcji i informacja uzyskana na każdym poziomie jest zapisywana do bazy danych. W ten sposób zgromadzona informacji pozwala na indeksowanie, a następnie wyszukiwanie obrazów za pomocą zapytań odwołujących się do cech występujących w nich obiektów [19]. Do zadawania pytań do bazy obrazowej istnieje interfejs graficzny, ponieważ użytkownik musi mieć możliwość łatwego formułowania zapytań o cechy uzyskiwane na każdym poziomie analizy obrazu.

Indeksowanie obrazowej bazy danych powinno się opierać w pierwszym rzędzie na zawartości obrazów i na potrzebach użytkownika, ułatwiając szybkie i sprawne uzyskanie odpowiedzi. Jest to jedno z trudniejszych zagadnień po samym rozpoznawaniu obrazu. Indeksowanie bazy multimedialnej musi być ściśle związane ze strukturami danych wykorzystywanymi w bazie. O ile indeksowanie obrazów po cechach niskiego poziomu, które mają wartości liczbowe nie stanowi w tej chwili problemu, ale indeksowanie według zależności przestrzennych jest już znacznie trudniejsze.

Kolejnym etapem użytkowania takiej baz, jest jej przeszukiwanie w zależności od zadawanych zapytań. Muszą być stworzone miary podobieństwa cech umożliwiające porównywanie obrazów i formalizm do formułowania zapytań tak, by użytkownik otrzymywał obraz lub kilka obrazów najbardziej zbliżonych to tego, którego szuka. Rysunek 1 przedstawia ogólny przepływ informacji w omawianej strukturze bazy danych. Lewa strona schematu dotyczy analizy zawartości obrazu, natomiast prawa strona to część służąca użytkownikowi i opierająca się przede wszystkim na zgromadzonej uprzednio informacji.

3 Wstępna obróbka obrazu

Obraz można definiować w różny sposób, ale my w tym opracowaniu zajmiemy się obrazem cyfrowym rozumianym jako dwuwymiarowa funkcja dyskretna $g(x,y)$ taką, że jej argumentami są dyskretne piksele, a wartościami kolory w przestrzeni RGB, czyli trójki liczb z przedziału $[0,255]$ ¹. Początek układu współrzędnych przyjmujemy w lewym górnym rogu obrazu tak jak to pokazuje rys. 2. Całe oprogramowanie operuje na tak skonstruowanych obrazach zwanych dalej obrazami rastrowymi lub kolorowymi bitmapami.



Rysunek 2 Obraz rastrowy

Czasem, dla uproszczenia, operuje się na obrazach o różnych poziomach szarości zamiast na obrazach kolorowych. W takim przypadku wartościami funkcji g są pojedyncze liczby z zakresu $[0, 255]$. Natomiast najprostszą wersją obrazów rastrowych jest obraz binarny czyli taki, którego piksele mają tylko kolor biały lub czarny w zapisie oznaczany jako $[0,1]$. Niesie on mało informacji, ale w zamian za to istnieją szybkie algorytmy obróbki takich obrazów. (Patrz rozdz. 3.3).

3.1 Modyfikacja histogramu

Histogram jest to funkcja pokazująca jaka liczba pikseli w obrazie ma określony poziom szarości lub poziom koloru. Na osi rzędnych odkładamy liczbę pikseli, a na osi odciętych odkładamy numery poziomów szarości lub kolory (patrz przykład na rys. 3).

¹ Przykładowy zakres kolorów 255 nie oznacza, że prezentowane dalej algorytmy dotyczące obrazów kolorowych nie będą działać na obrazach o większej liczbie kolorów np. True Color. Ograniczenie wynika tylko z faktu, że obrazy o większej liczbie kolorów zajmują więcej pamięci.

W przypadku obrazów drastycznie różniących się między sobą jasnością lub kontrastem zachodzi potrzeba modyfikacji histogramów w celu ich ujednoczenia (do dalszego porównywania) lub poprawienia jakości obrazu. Jeśli zatem mamy dany obraz, to jego histogram $P(g)$ opisujemy przez:

$$P(g) = \sum_{g_{\min}}^{g_{\max}} I(g) \quad (1)$$

gdzie: $I(g)$ – liczba punktów o danym poziomie szarości,

g – poziom szarości,

g_{\max} – najwyższy poziom szarości,

g_{\min} – najniższy poziom szarości.

Zmodyfikowany obraz o nowym histogramie opisanym jako:

$$P(f) = \sum_{f_{\min}}^{f_{\max}} H(f) \quad (2)$$

gdzie: $H(f)$ – liczba punktów o danym poziomie szarości,

f – poziom szarości,

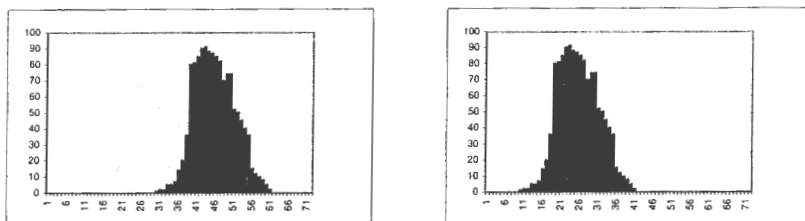
f_{\max} – najwyższy poziom szarości,

f_{\min} – najniższy poziom szarości.

Zatem modyfikacja polegająca na zmianie jasności nowego obrazu polega na przyporządkowaniu pikselom nowych wartości poziomów szarości. Obrazowo mówiąc „przesuwamy” jedynie argument funkcji $P(g)$, co daje w wyniku przesunięcie całego histogramu, czyli :

$$P(f) = P(g - [g_{\min} - f_{\min}]) \quad (3)$$

Przykład takiej modyfikacji histogramów pokazany jest na rys. 3.

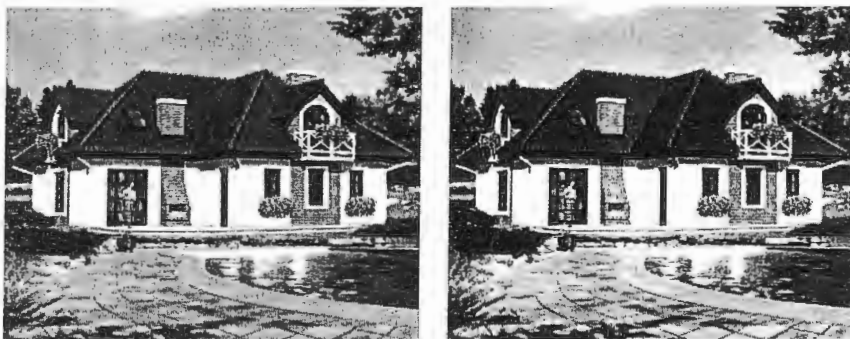


Rysunek 3 Zmiana jasności obrazów przez „przesunięcie” histogramów

Aby wyostrzyć obraz trzeba już użyć większego zakresu poziomów szarości lub kolorów. Obrazowo mówiąc trzeba „rozciągnąć” nowy histogram. $P(f)$ „rozciąga” się do $P(g)$ jeśli argumenty przeliczamy według wzoru:

$$g = \frac{g_{\max} - g_{\min}}{f_{\max} - f_{\min}} f + g_{\min} \quad (4)$$

Mozna też dodatkowo uśrednić sąsiadujące ze sobą wartości co daje efekt zwiększonej ostrości widoczny na rys. 4. Histogram bywa czasem modyfikowany nie w sposób liniowy, ale również w sposób wykładniczy, logarytmiczny lub hiperboliczny, co w określonych przypadkach pozwala uzyskać poszukiwane cechy obrazu [51], [10].



Rysunek 4 Przykład obrazu domu jednorodzinnego przed (po lewej) i po zmianie histogramu (po prawej). Wartości kolorów zostały obliczone według wzoru (4), a następnie uśredniono każde trzy sąsiednie poziomy.

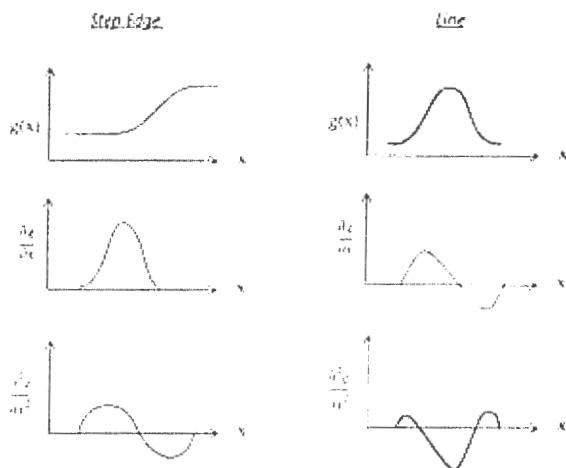
3.2 Wydzielanie krawędzi

Poza zliczaniem liczby punktów w obrazie można również wyznaczyć krawędzie obiektów. Przedstawione poniżej metody dotyczą obrazów o poziomych szarości, ponieważ dla obrazów kolorowych jest to znacznie bardziej kłopotliwe. Najpierw przyjrzyjmy się standardowym algorytmom wyznaczania krawędzi. Najczęściej stosuje się laplasjan o wzorze:

$$\nabla^2 g(x, y) = \frac{\partial^2}{\partial x^2} g(x, y) + \frac{\partial^2}{\partial y^2} g(x, y) \quad (5)$$

gdzie: $g(x, y)$ jest poziomem szarości piksela o współrzędnych (x, y) .

Laplasjan jest operatorem liniowym, niezmienniczym względem przesunięcia. Stosując go otrzymujemy zerowanie się drugiej pochodnej, co wyznacza nam poszukiwaną krawędź. Cała procedura jest zilustrowana na rys. 5.



Rysunek 5 Wykres rodzaju krawędzi (u góry), pierwszej pochodnej (na środku) i drugiej pochodnej (u dołu).

W zastosowaniach dyskretnych (czyli gdy nie liczymy pochodnej dla funkcji ciągłej tylko dyskretniej dla każdego dyskretnego piksela obrazu) w zależności od operacji używa się odpowiednich macierzy (masek) do splotu z obrazem $G(i, j)$:

$$H = G * M \quad (6)$$

$$H(i, j) = \sum_m \sum_n G(m, n) M(i - m, j - n) \quad (7)$$

W przypadku obliczania dyskretnego laplasjanu $H(i, j)$ używa się następujących macierzy (masek) M:

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

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

Na podobnej idei oparte są też algorytmy Sobela [57] i Priwetta [52]. Używają one do wyznaczenia krawędzi w kierunkach x i y masek 3×3 odpowiednio przedstawionych w tabelach 1 i 2. Maksimum ze splotu w obu kierunkach jest uważana za wartość krawędzi w danym punkcie. Wynik działania tych masek w przypadku algorytmu Sobela jest przedstawiony na rys. 6.

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

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

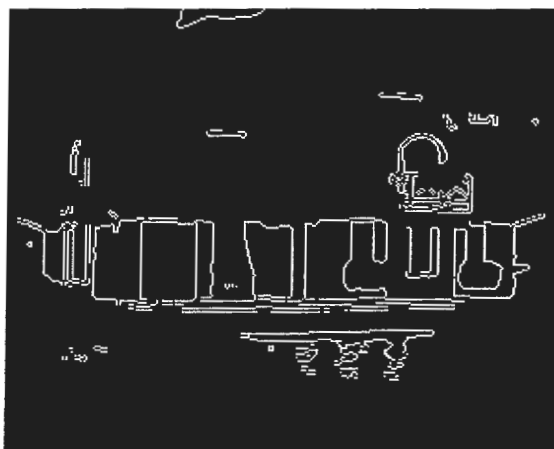
Tabela 1 Maski do wyznaczania krawędzi pionowych i poziomych zaproponowane przez Sobela

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

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

Tabela 2 Maski do wyznaczania krawędzi pionowych i poziomych zaproponowane przez Prewitta

Sobel edges for cluster 5



Rysunek 6 Wyznaczenie krawędzi w sposób zaproponowany przez Sobela

Znacznie bardziej rozbudowanym algorytmem do wyznaczania krawędzi jest algorytm Canny [6]. Składa się on z kilku kroków, a mianowicie w pierwszym z nich następuje odfiltrowanie drobnych szumów z obrazu oryginalnego. Następnie używa się maski Gaussa 5x5 do wygładzania obrazu. W ten sposób unika się błędów w lokalizacji krawędzi. W drugim kroku do wygładzonego już obrazu stosuje się operator Sobela w celu znalezienia gradientów w kierunkach x i y . „Wysokość” krawędzi G jest aproksymowana przy pomocy wzoru:

$$|G| = |G_x| + |G_y| \quad (8)$$

gdzie: G_x – jest gradientem uzyskanym ze splotu z maską w kierunku x (patrz tabela 1),

a G_y – gradientem w kierunku y . W następnym kroku znajdujemy kierunek krawędzi. Ogólnie można przyjąć, że nachylenie krawędzi znajdujemy z funkcji:

$$\theta = \arctan\left(\frac{G_y}{G_x}\right) \quad (9)$$

Oczywiście w przypadku gdy $G_x = 0$ kierunek krawędzi wynosi 0° lub 90° w zależności od tego, ile wynosi gradient w kierunku y . Jeśli do określania krawędzi użyjemy maski 5×5 to uzyskamy również możliwość określania krawędzi o kątach nachylenia 45° i 135° . Algorytm ten daje w większości krawędzie zamknięte dookoła obiektów (rys. 7), co jest jego dużą zaletą z naszego punktu widzenia, ponieważ kontury zamknięte dają możliwość łatwego wyznaczania tych obiektów na obrazie.

Canny edges for cluster 2

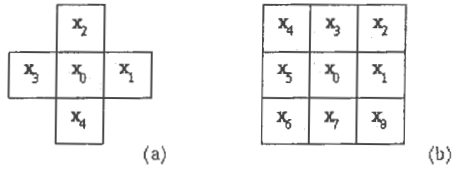


Rysunek 7 Krawędzie wyznaczone algorytmem Canny.

3.3 Operacje morfologiczne

Operacje morfologiczne na obrazie wyrosły z morfologii matematycznej. Podstawowa koncepcja została oparta na algebrze zbiorów wprowadzonych przez Minkowskiego [43] i pojęciach topologicznych Matherona [40]. Jest to zbiorcza nazwa określająca operacje boolowskie na wartościach określonych pikseli w obrazach binarnych. Aby mówić o jakichkolwiek operacjach morfologicznych należy zdefiniować sąsiedztwo pojedynczego piksela. Cztero-sąsiedztwo opisujemy jako zbiór 5 pikseli (X_0, X_1, X_2, X_3, X_4),

patrz rys. 8a), natomiast ośmio-sąsiedztwo opisujemy jako zbiór 9 pikseli (X_0, X_1, \dots, X_7), patrz rys. 8b).



Rysunek 8 a) 4-sąsiedztwo względem punktu x_0 , b) 8-sąsiedztwo względem punktu x_0 .

Najprostsze sąsiedztwo, czyli 4-sąsiedztwo punktu $g(i,j)$ określamy jako wektory

$[-1, 0, 1]$ i $\begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}$, czyli w innej formie sąsiedztwo dla piksela $g(i,j)$ opisujemy jako

$\{(i,j-1), (i,j), (i,j+1)\}$ i $\{(i-1,j), (i,j), (i+1,j)\}$. Wszystkie operacje morfologiczne używają elementu strukturalizującego (structuring element²) H (często o wymiarze 3×3) składającego się z zer i jedynek, zależnie od tego jaki kształt jest dla nas istotny. Wykonuje się splot tego elementu z obrazem, a wynik splotu jest zapisywany w obrazie wynikowym na pozycji danego piksela. Zatem dla cztero-sąsiedztwa dokonujemy operacji boolowskich dla następujących pikseli:

$$g(i,j) = B((i,j-1), (i,j), (i,j+1))$$

$$g(i,j) = B((i-1,j), (i,j), (i+1,j))$$

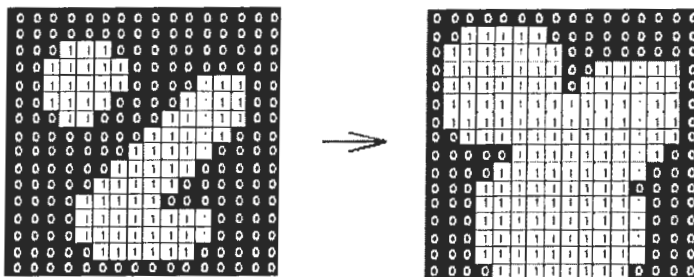
Dwoma podstawowymi operacjami morfologicznymi są dylatacja i erozja. Ogólnie dylatację można wyrazić symbolicznie jako:

$$g(i,j) = f(i,j) \oplus H(i,j) \quad (10)$$

gdzie: $f(i,j)$ jest obrazem binarnym o wymiarze $1 \leq i, j \leq N$, a $H(i,j)$ jest elementem strukturalizującym o wymiarze $1 \leq i, j \leq L$. Rys. 9 pokazuje jak wygląda obraz binarny po operacji dylatacji w stosunku do obrazu oryginalnego. Głównym celem dylatacji jest pogrubienie granic obiektów na obrazie. Dylatacja jest operacją przechodnią

$$A \oplus B = B \oplus A$$

² Structuring element jest też w literaturze polskie tłumaczony jako element strukturalny



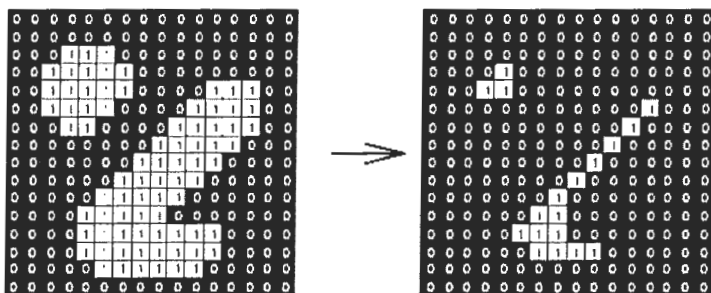
Rysunek 9 Obraz binarny przed i po dylatacji elementem strukturalizującym 3x3.

Natomiast drugą podstawową operację morfologiczną – erozję można wyrazić symbolicznie jako:

$$g(i,j) = f(i,j) \ominus H(i,j) \quad (11)$$

gdzie: $f(i,j)$ jest obrazem binarnym o wymiarze $1 \leq i, j \leq N$, a $H(i,j)$ jest elementem strukturalizującym o wymiarze $1 \leq i, j \leq L$. Rys. 10 pokazuje jak wygląda obraz binarny po operacji erozji w stosunku do obrazu oryginalnego. Głównym celem erozji jest utworzenie cieńszych linii i powiększenie już istniejących dziur na obrazie. Erozja w przeciwieństwie do dylatacji nie jest operacja przechodnią

$$A \ominus B \neq A \ominus B.$$



Rysunek 10 Obraz binarny przed i po erozji elementem strukturalizującym 3x3.

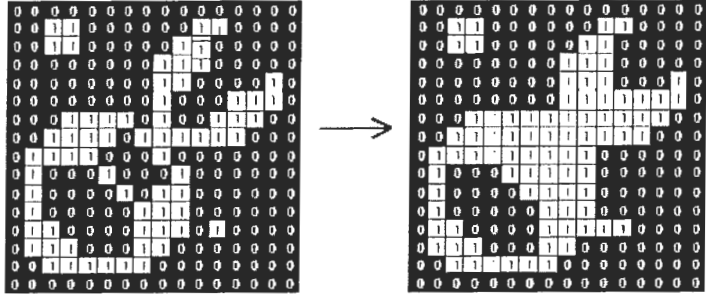
Bardziej złożoną operacją jest operacja domknięcia, którą definiujemy jako dylatację poprzedzoną przez erozję, co symbolicznie zapisujemy jako:

$$g(i, j) = f(i, j) \bullet H(i, j), \quad (12)$$

a możemy też zapisać jako:

$$g(i,j) = [f(i,j) \oplus H(i,j)] \ominus \hat{H}(i,j) \quad (13)$$

gdzie: $\hat{H}(i,j)$ jest lustrzanym odbiciem elementu strukturalizującego. Operacja domknięcia jest stosowana w celu zamknięcia małych dziur i krótkich przerw w krawędziach, co jest pokazane na rys. 11.



Rysunek 11 Obraz binarny przed i po operacji domknięcia elementem strukturalizującym 3x3.

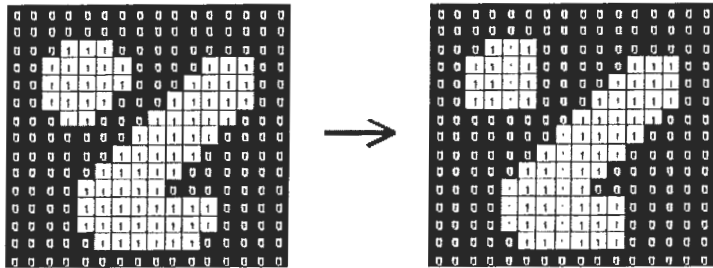
Następną złożoną operacją jest operacja otwarcia, którą definiujemy jako erozję poprzedzoną przez dylatację, co symbolicznie zapisujemy jako:

$$g(i,j) = f(i,j) \circ H(i,j), \quad (14)$$

a możemy też zapisać jako:

$$g(i,j) = [f(i,j) \ominus \hat{H}(i,j)] \oplus H(i,j) \quad (15)$$

Operacja otwarcia jest stosowana w celu wygładzenia i pocienienia konturów i eliminacji „kropek” na obrazie oraz wygładzenia ostrych załamów.



Rysunek 12 Obraz binarny przed i po operacji otwarcia elementem strukturalizującym 3x3.

3.4 Segmentacja

Celem segmentacji jest redukcja liczby danych poprzez stworzenie kategorii i zgrupowanie razem podobnych danych. Takie grupowanie jest naturalne dla ludzkiego przetwarzania informacji. W naszym przypadku to grupowanie zachodzi dla pikseli obrazu o podobnych atrybutach. Zatem, aby w procesie dalszej analizy wydzielić z obrazu jakiegokolwiek obiekty należy najpierw przeprowadzić segmentację tego obrazu. Segmentacja jest to podział obrazu na rozdzielne regiony. W przypadku analizowanych przez nas obrazów rastrowych region jest to połączony zbiór pikseli. Każdy region charakteryzowany jest przez określone atrybuty. Najczęściej używanymi atrybutami przy segmentacji obrazu są:

- Jasność dla obrazów monochromatycznych,
- Kolor dla obrazów barwnych,
- Tekstura,
- Krawędzie, itp.

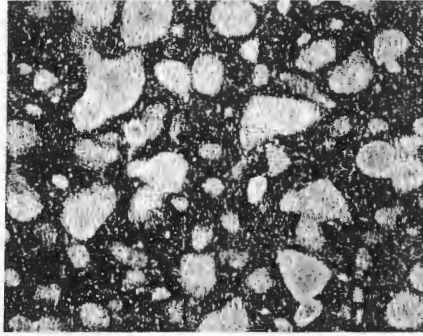
Możemy wyróżnić trzy różne filozofie segmentacji:

1. podejście oparte na regionach, w którym każdy piksel jest w zależności od atrybutów przyporządkowywany do któregoś z regionów;
2. podejście oparte na granicach, w którym znajdujemy tylko granice pomiędzy regionami;
3. podejście oparte na krawędziach, w którym wyznaczamy krawędzie i określamy regiony jako obszary wewnątrz tych krawędzi.

Każda z tych filozofii jest przydatna przy innych celach segmentacji. Oddzielnym problemem jest liczba regionów na które dzielimy obraz. Nie można przyjąć zasady, że im mniej tym lepiej, lub im więcej tym lepiej. Trzeba postarać się przyjąć najbardziej dystynktywny parametr dla danego obrazu i wartości tego parametru podzielić w miarę równomiernie.

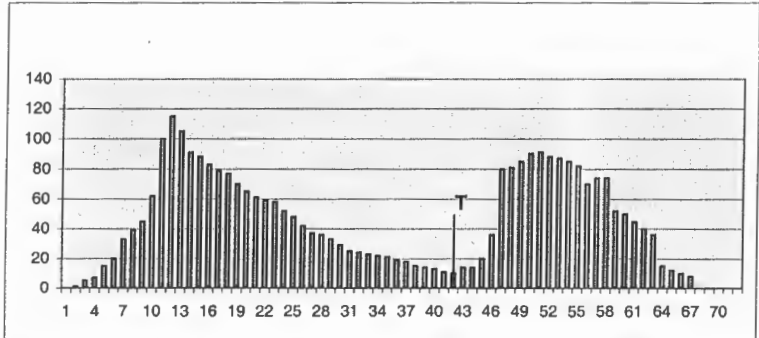
3.4.1 Segmentacja oparta na histogramie bimodalnym

Do podejścia opartego na regionach możemy zaliczyć progowanie. Polega ono na wyborze obiektów wyraźnie odcinających się od tła jak na rysunku 13. Obrazy z wyraźnym rozróżnieniem obiektów i tła mają histogram bimodalny (patrz rys. 14) co oznacza, że jeden z pików histogramu odpowiada za punkty należące do obiektów, a drugi - za punkty należące do tła.



Rysunek 13 Przykład obrazu o bimodalnym histogramie. Ciemne tło tworzy jeden pik na takim histogramie, a jasne komórki drugi.

Trzeba więc wybrać minimum pomiędzy dwoma modami histogramu i tam ustawić próg T , co jest pokazane na rys. 9. Metoda ta niestety przestaje poprawnie działać w przypadku obrazów zaszumionych.

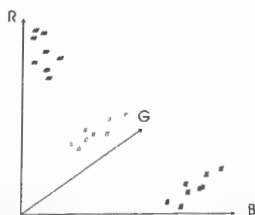


Rysunek 14 Bimodalny histogram z zaznaczonym progiem T . Punkty o mniejszym poziomie szarości uznajemy jako tło, a o większym jako punkty należące do poszukiwanych obiektów.

3.4.2 Segmentacja oparta na analizie skupień

Wykorzystywanie analizy skupień jest popularnym podejściem do segmentacji [42]. Dokładny opis algorytmu jest zamieszczony w rozdz. 9.1. W naszym przypadku przestrzenią cech, w której poszukujemy skupień jest przestrzeń kolorów. Kolor jest cechą najlepiej chyba postrzeganą przez człowieka [64]. Dlatego też stworzono różne modele opisu koloru takiej jak RGB, VHS, HSI, itd. [45], [35] by oddać choć w części problemy związane z nieliniowością i skorelowaniem pomiędzy barwami w postrzeganiu koloru przez człowieka.

W naszym przypadku przestrzeń RGB traktujemy jako trójwymiarową przestrzeń cech koloru (rys 15). Przeanalizujemy teraz wyniki działania algorytmu dla obrazów kolorowych w tej przestrzeni.



Rysunek 15 Przestrzeń kolorów RGB (traktowana jako trójwymiarowy histogram)



Rysunek 16 Obraz oryginalny narożnego okna

Rezultaty eksperymentów dla obrazów kolorowych

Niech obrazy oryginalne pokazane będą na rysunkach 4 i 16. Niezależnie od wprowadzanych przez Bezdeka i jego następców miar efektywności tego algorytmu do naszych celów przyjmuje się $c = 5,6,7$. Wynik działania tego algorytmu widoczny jest na

zdjęciu domu jednorodzinnego rys. 17 i narożnego okna na rysunkach 19 i 21. Dla tych dwóch zdjęć segmentacja jest dokonywana dla pojedynczych składowych RGB. Jak widać czerwone cegły i zielone odbicia w oknach należą według tego algorytmu do tego samego skupienia.

Podjęto też próbę zsumowania wartości poszczególnych składowych RGB, ponieważ powstają wtedy większe różnice pomiędzy poszczególnymi pikselami, a co za tym idzie większe różnice pomiędzy poszczególnymi skupieniami. Dla porównania zamieszczono poniżej segmentację okna narożnego (rys. 20 i 22) dla zsumowanych składowych RGB i otrzymane skupienia dalej nie odzwierciedlają istniejących w rzeczywistości kolorów. Nie ratuje nas też zwiększenie liczby skupień, bo zaczynają być wtedy dzielone obiekty, jak to widać na rys. 18, na którym niebo jest podzielone między dwa skupienia, w tym jedno takie, które poza niebem zawiera też dach.



Rysunek 17 Efekt segmentacji algorytmem c-środków dla $c = 6$ z nałożonymi krawędziami dla lepszego podkreślenia segmentów. Każdy poziom szarości reprezentuje jeden segment.



Rysunek 18 Efekt segmentacji algorytmem c-środków dla $c = 12$

image labeled by cluster index

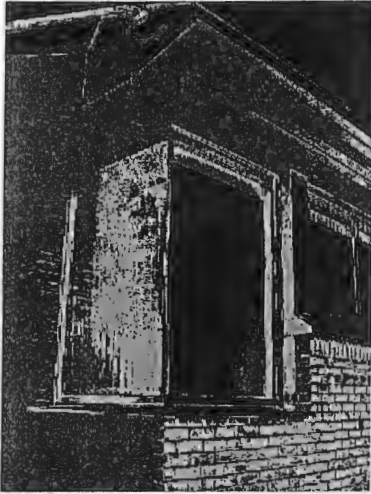
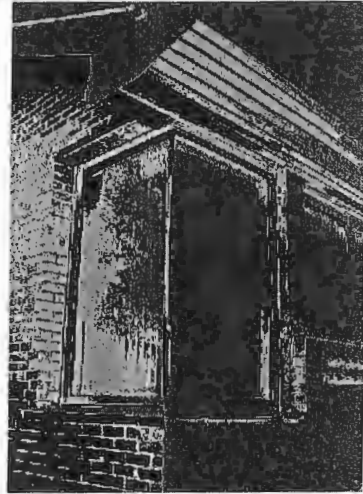


image labeled by cluster index sumowany RGB



Rysunek 19 Efekt segmentacji algorytmem c-
środków dla $c=5$. Segmentacja jest
wykonywana dla poszczególnych
składowych RGB

Rysunek 20 Efekt segmentacji algorytmem c-
środków dla $c=5$. Przed segmentacją
wartości RGB dla każdego piksela
zostały zsumowane.

image labeled by cluster index

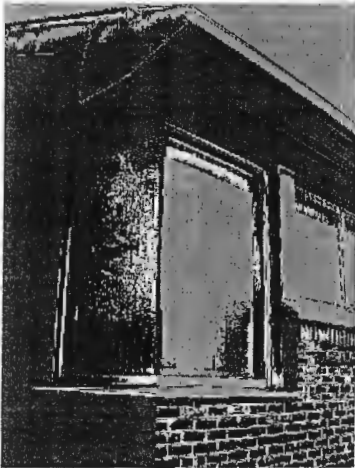
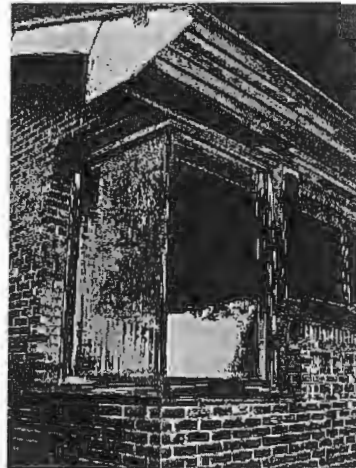


image labeled by cluster index sumowany RGB



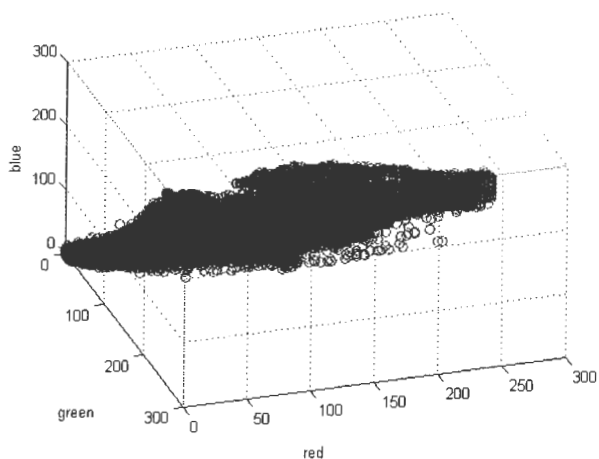
Rysunek 21 Efekt segmentacji algorytmem c-
środków dla $c=7$. Segmentacja jest
wykonywana dla poszczególnych
składowych RGB.

Rysunek 22 Efekt segmentacji algorytmem c-
środków dla $c=7$. Przed segmentacją
wartości RGB dla każdego piksela
zostały zsumowane.

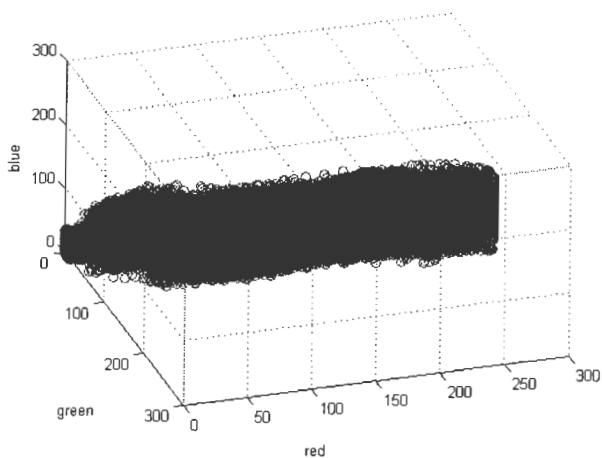
3.4.3 Segmentacja oparta na przestrzeni kolorów RGB

Aby wyjaśnić tę sytuację należy przeanalizować rozkład kolorów na rysunkach w przestrzeni kolorów RGB. Taką analizę pokazują rysunki 23 i 24. Jeśli wykonamy wykres rozkładu kolorów w przestrzeni RGB z pominięciem współrzędnych przestrzennych pikseli, to otrzymamy praktycznie dla wszystkich dotychczas analizowanych obrazów jednolitą chmurę punktów w otoczeniu przekątnej. Wynika stąd, że rozkład punktów nie jest ani równomierny, ani nie wykazuje wyraźnych skupisk, więc algorytm c-środków nie jest w stanie poradzić sobie z takim przypadkiem.

W związku z tym, aby uzyskać skupienia oparte na informacji o kolorze i rozdzielające istniejące na obrazie obiekty zajęliśmy się kolorem każdego punktu. W nowym algorytmie segmentacji opieramy się jedynie na wartości koloru dla każdego punktu, traktowanego jako trójka liczb (R,G,B); (położenie punktu jest mniej istotne) [31]. Z tej trójki wybiera się największą wartość i ta wartość decyduje o kolorze segmentu, do którego przyporządkowujemy dany punkt. W ten sposób powstają trzy segmenty – czerwony, zielony i niebieski. Jeśli wartości R, G i B są sobie równe, to punktowi przyporządkowuje się kolor szary. Dodatkowo w zależności od jasności koloru dzieli się każdy segment na obszar I - jasny, II - średni i III - ciemny. W ten sposób powstaje 12 segmentów odzwierciedlających rzeczywisty rozkład kolorów na obrazie oryginalnym, ale wystarczająco zagregowany do wydzielenia obiektów z obrazu.



Rysunek 23 Rozkład punktów z rys. 4 w przestrzeni kolorów RGB.

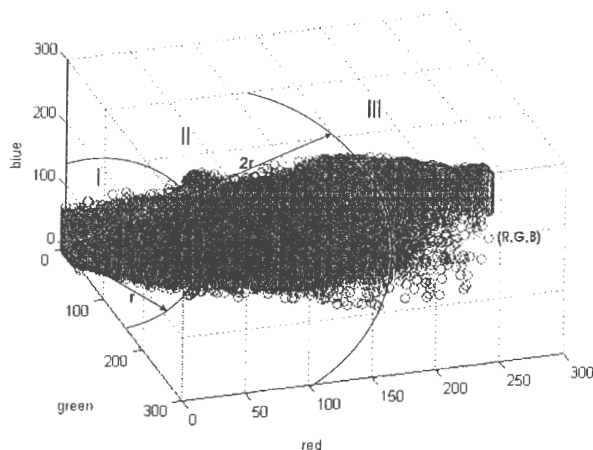


Rysunek 24 Rozkład punktów z rys. 16 w przestrzeni kolorów RGB.

Idea tego podziału jest przedstawiona na rys. 25. Promień r rozdzielający fragmenty kul jest liczony w przestrzeni euklidesowej według wzoru:

$$r = \frac{\sqrt{R_{\text{max}}^2 + G_{\text{max}}^2 + B_{\text{max}}^2}}{3} \quad (10)$$

gdzie: $R_{\max} = G_{\max} = B_{\max} \leq 255$. Jeśli wartości $R = G = B$ są równe lub prawie równe, to punkty należą do segmentu I - białego, II - szarego lub III - czarnego, w zależności od poziomu szarości. *Prawie równe* oznacza, że mieszczą się w przedziale $|R-G| < \sigma$ i $|R-B| < \sigma$, gdzie $10 < \sigma < 15$.



Rysunek 25 Sposób podziału punktów na segmenty. Obszary I, II, III pokazują jasność punktu, natomiast największa wartość z trójki (R,G,B) określa kolor segmentu, do którego przyporządkujemy dany punkt.

Poniżej na rysunkach 26, 27, 29 i 31 są przedstawione wyniki działania algorytmu opartego na kolorach dla obrazów o różnych wymiarach. Wszystkie z nich oczywiście przedstawiają domy lub ich elementy.

Opisany algorytm jest bardzo szybkim algorytmem w porównaniu z metodą c-środków. Wynika to z faktu, że w tym algorytmie są stosowane jedynie porównania trzech liczb i nie ma żadnego mnożenia. Dokładne czasy obliczeń dla segmentacji metodą c-środków i algorytmem opartym na kolorach w zależności od wielkości obrazu w pikselach przedstawia tabela 3. Łatwo zauważyć, że algorytm kolorowy jest od jednego do nawet dwóch rzędów wielkości szybszy niż metoda c-środków przy podziale na tę samą liczbę skupisk.



Rysunek 26 Segmentacja oparta na kolorach dla obrazu domu jednorodzinnego.



Rysunek 27 Segmentacja oparta na kolorach dla obrazu narożnego okna



Rysunek 28 Originalny obraz domu z garażem



Rysunek 29 Segmentacja oparta na kolorach dla obrazu domu z garażem.



Rysunek 30 Oryginalny obraz rezydencji



Rysunek 31 Segmentacja oparta na kolorach dla obrazu rezydencji

Rysunek oryginalny	Rozmiar obrazu [w pikselach]	algorytm	5 skupień [w sekundach]	12 skupień [w sekundach]
Rys. 4	240×300	C-średnich	11,22	62,82
		kolorowy	-	0,99
Rys. 16	600×450	C-średnich	105,36	721,65
		kolorowy	-	7,05
Rys. 28	100×150	C-średnich	2,95	20,30
		kolorowy	-	0,32
Rys. 30	325×430	C-średnich	51,52	212,83
		kolorowy	-	2,89
Rys. 50	480×640	C-średnich	115,23	730,52
		kolorowy	-	7,12

Tabela 3 Porównanie czasów obliczeń dla dwóch algorytmów segmentacji przeprowadzanych na procesorze AMD 2500+, 1.75 GHz, z pamięcią RAM 512 MB.

3.3.4 Segmentacja oparta na rozmytej analizie skupień (fuzzy c-means clustering)

Istnieje też podejście rozmyte do segmentacji obrazów, które można z kolei podzielić na 4 rodzaje:

- segmentacja oparta na progowaniu rozmytym
- rozmyta analiza skupień
- segmentacja nadzorowana
- segmentacja w oparciu o przyjęte reguły

Segmentacja oparta na progowaniu rozmytym jest głównie stosowana do obrazów monochromatycznych, natomiast w przypadku obrazów kolorowych jest bardziej problematyczna.

W przypadku rozmytej analizy skupień fakt, że każdy punkt należy do danego skupienia w stopniu określonym przez funkcję przynależności $\mu_{C_j}(x_{mn})$ nie zmienia naszego zadania. Jeśli nawet wprowadzimy rozmytą przynależność punktu do segmentu nie zmieni to w znaczący sposób rozkładu segmentów dopóki funkcja przynależności będzie zależna od współrzędnych punktu x_{mn} . Obiecująca wydaje się jedynie myśl, że funkcja przynależności będzie zależna od wartości koloru RGB - $\mu_{C_j}(x_{RGB})$. Nie przeprowadziliśmy jeszcze takich eksperymentów, więc nie wiadomo, czy położenie punktów wokół przekątnej układu RGB nie będzie jednak silniejszym uwarunkowaniem.

3.5 Wydzielenie obiektów z tła

Jeśli obrazy mają bimodalny histogram to łatwo jest wydzielić obiekty z tła przez opisany powyżej algorytm progowania histogramu (Patrz pkt. 3.4.1.). Jeśli wydzielimy z obrazu skupienia to również możemy wydzielić obiekt w postaci jednego lub kilku segmentów z tła. Należy wtedy jedynie znać element charakteryzujący segment obiektu np. kolor.

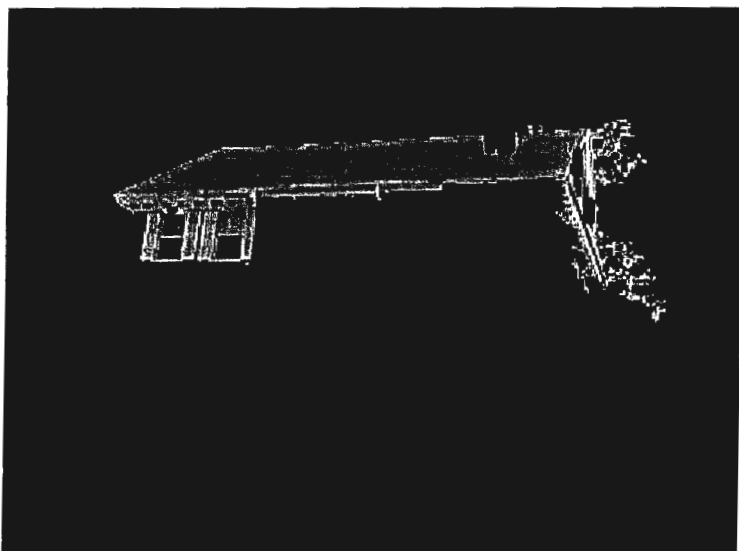
Następną metodą wydzielenia obiektów z tła są aktywne kontury. Dokładny opis metody znajduje się w rozdz. 9.5.



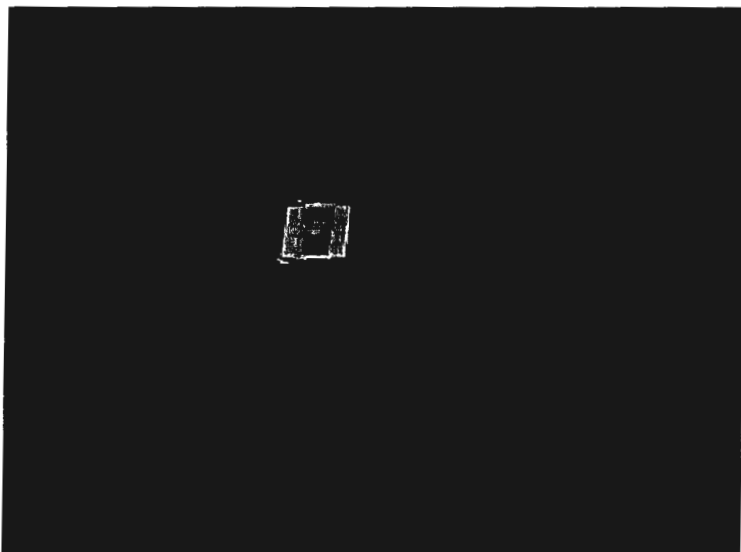
Rysunek 32 Segment w kolorze średnim niebieskim dla rezydencji, na podstawie którego są wydzielane poszczególne obiekty

W tym rozdziale skupimy się jednak na wydzieleniu obiektów w oparciu o zaproponowaną wyżej metodę opartą na segmentacji kolorowej. Aby to osiągnąć należy każdy z 12 otrzymanych segmentów poddać operacjom najpierw zamknięcia, a potem otwarcia, aby pozbyć się małych dziurek w obiektach i drobnych kropek, które traktujemy na tym poziomie jako szum. Elementy strukturalizujące dla tych operacji są dobierane automatycznie w zależności o wielkości obrazów. Również wydzielane obiekty nie mogą być

mniejsze niż pewien założony próg, też zależny od wielkości obrazu. Obiekty muszą mieć wielkość rozróżnialną przez nasze oko czyli ok. 2-3% powierzchni obrazu.



Rysunek 33 Dach i dwa górne okna rezydencji z okiennicami wydzielone z niebieskiego segmentu i przedstawione w kolorach oryginalnych

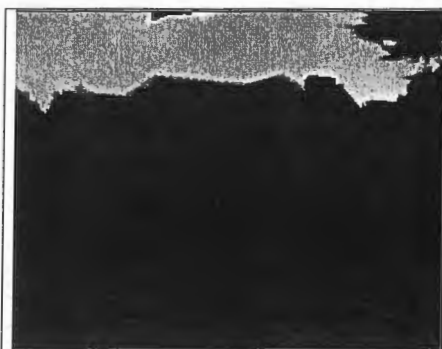


Rysunek 34 Okno górne z okiennicami wydzielone z warstwy niebieskiej rezydencji i przedstawione w kolorach oryginalnych

Wyberzmy jeden segment na obrazie rezydencji do ilustracji całej procedury. Przykładowy segment jest w kolorze średnim niebieskim rys. 32 wydzielony z rys. 31. Każdy obiekt po wydzieleniu jest oddzielnie rysowany i jest też dla niego liczony kolor średni. Rys. 33 pokazuje dach i lewe górne okna z okiennicami wydzielone z oryginalnego obrazu rezydencji.



Rysunek 35 Okno dolne z okiennicami wydzielone z warstwy niebieskiej rezydencji i przedstawione w kolorach oryginalnych



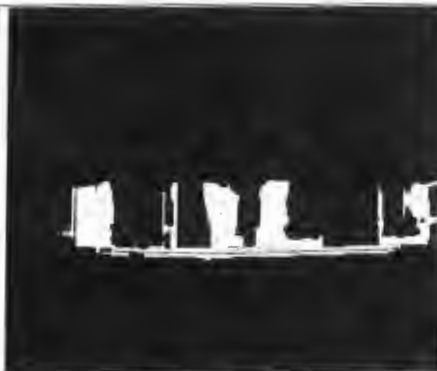
Rysunek 36 Niebo nad domem jednorodzinny wydzielone z warstwy średniej niebieskiej i przedstawione w kolorach oryginalnych



Rysunek 37 Komin domu jednorodzinny wydzielony z warstwy średniej czerwonej i przedstawiony w kolorach oryginalnych



Rysunek 38 Balustrada balkon domu jednorodzinne wydzielona z warstwy białej i przedstawiona w kolorach oryginalnych



Rysunek 39 Ściany domu jednorodzinne wydzielone z warstwy średniej czerwonej i przedstawione w kolorach oryginalnych



Rysunek 40 Ceglane ściany wokół narożnego okna wydzielone z warstwy czerwonej i przedstawione w kolorach oryginalnych



Rysunek 41 Blacha nad narożnym oknem wydzielona z warstwy niebieskiej i przedstawiona w kolorach oryginalnych

4. Modele danych

Po wydzieleniu obiektów z poszczególnych obrazów można zająć się umieszczaniem ich w obrazowej bazie danych. Jako dane (rekordy) w bazie są umieszczane całe obrazy oraz wydzielone z obrazów widoki obiektów. Obiekty są połączone z obrazami poprzez odpowiednie odnośniki w bazie danych, ale mogą też być wyświetlane samodzielnie. Poza tym każdy obiekt jest charakteryzowany przez następujący zbiór cech pierwotnych:

- Średni kolor
- Parametry tekstury
- Obwód
- Powierzchnia
- Środek ciężkości
- Momenty główne bezwładności
- Osie bezwładności
- Mimośród
- Kod granic

Wszystkie powyższe cechy są liczone w oparciu o model rastrowy obiektów.

Na drugim poziomie analizy obrazu znajdują się cechy logiczne, które nie są już bezpośrednio związane z pikselami. Na tym poziomie posługujemy się już obiektami wydzielonymi na pierwszym poziomie, takimi jak: trójkąt, owal, prostokąt, ale jeszcze nie przyporządkowujemy im deskryptorów. Na cechy logiczne składają się relacje przestrzenne poszczególnych obiektów, takie jak:

- rozmieszczenie obiektów w całym obrazie,
- położenie obiektów względem siebie,
- relacje sąsiedztwa,
- relacje zawierania,
- relacje przecinania, itp.

Wszystkie powyższe cechy są liczone w oparciu o model wektorowy obiektów.

Na najwyższym poziomie analizy obrazu posługujemy się elementami semantycznymi w rozumieniu użytkownika, takimi jak: domy, okna, dachy, samochody, drzewa itp. Na tym poziomie najłatwiej jest też użytkownikowi kierować zapytania do systemu. Zostaną więc stworzone listy przypadków, które będą odzwierciedlać grupy

elementów istniejące już w bazie danych i jednocześnie zakwalifikowane przez system jako okna, dachy, itp.

4.1 Ilościowe modele danych



Rysunek 42 Obiekty wydzielone z domu jednorodzinnego i przedstawione w kolorach średnich



Rysunek 43 Obiekty wydzielone z domu z garażem i przedstawione w kolorach średnich



Rysunek 44 Obiekty wydzielone z narożnego okna i przedstawione w kolorach średnich bez refleksów w szybach

4.1.1 Kolor średni

Aby kolor opisać w postaci ilościowej należy w określony sposób zagregować kolory każdego piksela. W naszym opracowaniu przyjęliśmy najprostszy sposób, a mianowicie cechą obiektu jest jego kolor średni, liczony jako suma składowych R, G i B i dzielona przez liczbę wszystkich pikseli należących do danego obiektu. Przykłady obrazów wyświetlanych jako zbiór obiektów w ich kolorach średnich przedstawiają następujące rysunki:



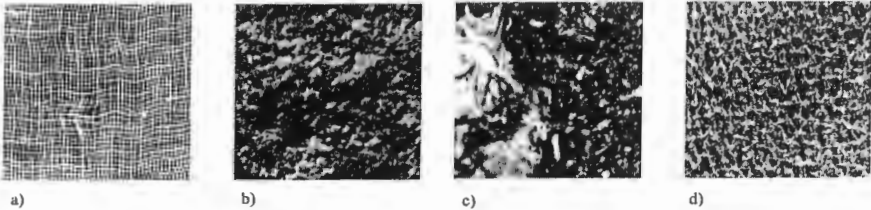
Rysunek 45 Obiekty wydzielone z rezydencji i przedstawione w kolorach średnich

4.1.2 Analiza tekstur

Tekstura naturalna według Pratta [51] to obrazy pochodzące z realnych scen zawierające samopowtarzające się układy pikseli np.: cegły, piasek, trawa, rafia, oraz tkanina, marmur, korek, itp., jak na rys. 46. Teksturę można charakteryzować przy pomocy bardzo różnorodnych parametrów zależnych między innymi od jej skali, poziomu rozdzielczości koloru itp. Przykładowymi parametrami opisującymi teksturę mogą być:

- jednorodność
- gęstość

- porowatość
- regularność
- liniowość
- kierunkowość
- częstotliwość i faza oraz inne



Rysunek 46 Przykładowe tekstury naturalne a) tkanina, b) kamień, c) marmur i d) korek

Część z wymienionych cech jest od siebie zależna, więc w opisie tekstury nie trzeba wyznaczać ich wszystkich. Fakt, że tekstura jest postrzegana poprzez tak różne parametry powoduje, że istnieje wiele różniących się od siebie sposobów opisu tekstury.

Jednym z najwcześniejszych podejść do opisu jakościowego tekstur było wykorzystanie ich cech statystycznych. Dotyczy ono obrazów monochromatycznych o wymiarach $N \times N$, o G poziomach szarości. W tym podejściu chcemy zobaczyć, jak rozkładają się położenia punktów o tym samym stopniu szarości względem siebie. W tym celu Haralick [23] zaproponował użycie macierzy wzajemnych wystąpień punktów o tym samym poziomie szarości (*co-occurrence matrices*), które stały się jedną z najlepiej znanych i szeroko stosowanych cech tekstury. Macierz $N-1 \times N-1$ wzajemnych wystąpień punktów o tym samym poziomie szarości P_d dla wektora przemieszczeń $\mathbf{d} = (dx, dy)$ jest definiowana jako wystąpienie takich par pikseli (i, j) , dla których odległość wynosi \mathbf{d} , czyli:

$$P_d(i, j) = |\{(r, s), (t, v) : g(r, s) = i, g(t, v) = j\}| \quad (16)$$

gdzie: $(r, s), (t, v) \in N \times N$, przy czym $(t, v) = (r + dx, s + dy)$, a $|\cdot|$ jest mocą zbioru.

Jako przykład można rozważyć obraz 4×4 zawierający trzy wartości szarości:

$$\begin{matrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 2 & 2 \\ 0 & 0 & 2 & 2 \end{matrix}$$

Dla wektora przemieszczeń $\mathbf{d} = (1, 0)$ otrzymujemy macierz 3×3 wzajemnych wystąpień punktów o tym samym poziomie szarości P_d

$$P_d = \begin{bmatrix} 4 & 0 & 2 \\ 2 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix} \quad (17)$$

Macierz wzajemnych wystąpień punktów o tym samym poziomie szarości ujawnia pewne własności charakteryzujące rozkład przestrzenny poziomów szarości w teksturze obrazu. Jeśli na przykład wystąpienia pikseli koncentrowały się wzdłuż przekątnej macierzy, wtedy tekstura była porowata ze względu na rozkład wektora \mathbf{d} . Haralick zaproponował wiele użytecznych cech tekstury, które można obliczyć na podstawie macierzy wystąpień punktów. Przykłady tych cech są podane w tabeli 4.

Cecha tekstury	Wzór
Energia	$\sum_i \sum_j P_d^2(i, j)$
Entropia	$-\sum_i \sum_j P_d(i, j) \log P_d(i, j)$
Kontrast	$\sum_i \sum_j (i - j)^2 P_d(i, j)$
Jednorodność	$\sum_i \sum_j \frac{P_d(i, j)}{1 + i - j }$ dla $i \neq j$
Korelacja	$\frac{\sum_i \sum_j (i - \mu_x)(j - \mu_y) P_d(i, j)}{\sigma_x \sigma_y}$ <p>gdzie: $\mu_x = \sum_i i \sum_j P_d(i, j)$</p> $\mu_y = \sum_j j \sum_i P_d(i, j)$ $\sigma_x = \sum_i (i - \mu_x)^2 \sum_j P_d(i, j)$ $\sigma_y = \sum_j (j - \mu_y)^2 \sum_i P_d(i, j)$

Tabela 4 Przykładowe cechy tekstury wyekstrahowane na podstawie macierzy wzajemnych wystąpień punktów o tym samym poziomie szarości.

Następną cechą, którą można policzyć traktując teksturę jako proces stochastyczny jest autokorelacja. Funkcja autokorelacji mówi nam jaka jest regularność tekstury oraz jaka

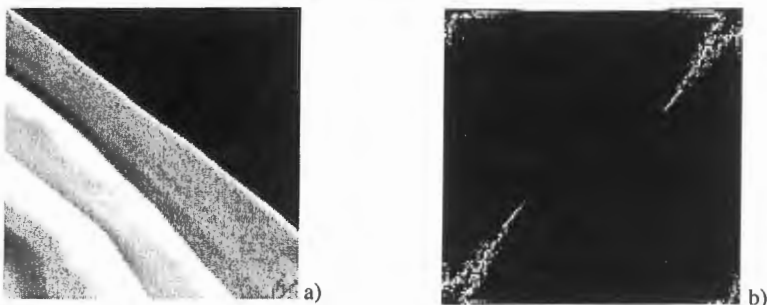
jest jej gładkość lub porowatość. Funkcję autokorelacji dla obrazu $g(i,j)$ możemy zdefiniować jako:

$$\rho(x, y) = \frac{\sum_{u=0}^N \sum_{v=0}^N g(u, v) g(u+x, v+y)}{\sum_{u=0}^N \sum_{v=0}^N g^2(u, v)} \quad (18)$$

Funkcja autokorelacji jest również związana z transformacją Fouriera. Rozważmy teraz funkcję w przestrzeni obrazu $g(i,j)$ oraz jej transformację Fouriera $F(u,v)$. Widmo energii dla tego obrazu w przestrzeni częstotliwości można przedstawić jako:

$$E(u, v) = |F(u, v)|^2 \quad (19)$$

Przykład widma energii dla faktury kolorowej prezentuje rys. 47 b).



Rysunek 47 a) oryginalny obraz faktury kolorowej, b) rozkład energii widma dla tego obrazu. Kierunkowa natura tej tekstury znajduje odbicie w kierunkowym rozkładzie energii widma.

W podejściu tym początkowo wykorzystywano cechy spektralne i dzielono dziedzinę częstotliwości na okręgi według wzoru (20) w celu znalezienia cech częstotliwościowych (rys. 48a)). Wykorzystywano też odzwierciedlenie kierunku faktury w dziedzinie częstotliwości, którą dzielono na poszczególne regiony kątowe, umownie nazywane „różą wiatrów” (rys. 48b)). W obliczeniach wykorzystujemy przejście do współrzędnych biegunowych, a mianowicie:

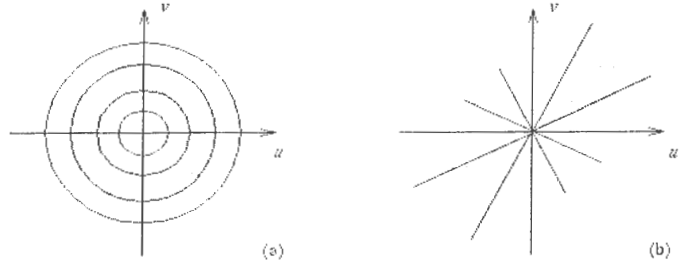
$$f_{r_1, r_2} = \int_0^{2\pi} \int_{r_2}^{r_1} |F(u, v)|^2 dr d\theta \quad (20)$$

gdzie:
$$r = \sqrt{u^2 + v^2} \quad (21)$$

oraz
$$\theta = \arctag\left(\frac{v}{u}\right) \quad (22)$$

Regiony kątowe obliczano według poniższego wzoru:

$$f_{\theta_1 \theta_2} = \int_{\theta_1}^{\theta_2} \int_0^{\infty} |F(u, v)|^2 r dr d\theta \quad (23)$$



Rysunek 48 a) Filtr kołowy dzielący dziedzinę częstotliwościową na okręgi, według których wyznaczano gładkość lub chropowatość tekstury; b) filtr kątowy, który wskazywał kierunkowość tekstury.

Opis cech tekstury kolorowej w ujęciu stochastycznym możemy oprzeć na kowariancji kolorowej. Przyjmijmy, że w przestrzeni kolorów RGB przyporządkowujemy każdemu pikselowi kolor jako trójkę liczb $\{R, G, B\}$, a kowariancję kolorową CC^{ij} pomiędzy dwoma punktami I^i i I^j określamy jako:

$$CC^{ij}(\Delta x, \Delta y) = \frac{1}{\sigma^i \sigma^j} \frac{1}{|I|} \sum_{x, y \in I} (I^i(x, y) - \mu^i)(I^j(x + \Delta x, y + \Delta y) - \mu^j) \quad (24)$$

gdzie: $I^i(x, y)$ jest to piksel o składowych koloru $i \in \{R, G, B\}$ i współrzędnych (x, y) . μ^i jest wartością średnią, a σ^i jest odchyleniem standardowym w przestrzeni kolorów. Kowariancja kolorowa jest liczona dla zbioru różnych odległości między pikselami $(\Delta x, \Delta y)$ w kierunku x i y . Macierz kwadratowa $CC^{ij}(\Delta x, \Delta y)$ może być definiowana dla stałej W dostrzegalnych przez człowieka odległości, takich, że $|\Delta x| \leq W, |\Delta y| \leq W$. Przy takiej konstrukcji otrzymujemy następujące właściwości kolorowej kowariancji: $CC^{ij}(\Delta x, \Delta y) \in [-1, 1]$, gdzie 1 oznacza najwyższą korelację, 0 oznacza, że w ogóle nie ma korelacji, a -1 oznacza, że mamy ujemną korelację, czyli kolory dopełniające się. Korelacja dla tego samego punktu wynosi 1, czyli $CC^{ii}(0, 0) = 1$ i jest symetryczna, czyli $CC^{ij}(\Delta x, \Delta y) = CC^{ji}(-\Delta x, -\Delta y)$. Zatem, wystarczy liczyć tylko połowę macierzy kowariancji dla wszystkich dziewięciu kombinacji kolorów: $CC^{RR}, CC^{BG}, CC^{RG}, CC^{RB}, CC^{GR}, CC^{GG}, CC^{GB}, CC^{BR}, CC^{BB}$.

Tekstura w ujęciu stochastycznym może być też rozumiana jako realizacja dwuwymiarowego procesu stochastycznego. Proces stochastyczny $F(x, y, t)$ może być opisany

dzięki znajomości jego łącznej gęstości prawdopodobieństwa

$$P_{t_1, \dots, t_J}(x_1, \dots, x_J, y_1, \dots, y_J) = P(F(t_1) = x_1, \dots, F(t_J) = x_J) \quad (25)$$

dla wszystkich próbek J , gdzie (x_j, y_j, t_j) reprezentuje czasowo-przestrzenne próbki funkcji obrazu $F(x_j, y_j, t_j)$. W szczególności obraz możemy opisać jako dwuwymiarowy proces Markowa (pole Markowa). Przy takim podejściu naturalnymi parametrami charakteryzującymi teksturę są parametry generującego ją procesu Markowa. Prekursorem opisu tekstury przy pomocy pól Markowa był Julesz [33].

Wiele tekstur poza właściwościami statystycznymi wykazuje też znaczne samopodobieństwo w różnych skalach. Nasuwa to pomysł analizy tekstur za pomocą fraktali. Trzeba więc przypomnieć zasadnicze pojęcie teorii fraktalnej takim jak wymiar fraktalny D . Ogólnie dla dowolnego obiektu samopodobnego istnieje związek pomiędzy współczynnikiem redukcji s , a liczbą części a na które obiekt może być podzielony, czyli:

$$a = \frac{1}{s^D} \quad (26)$$

stąd wymiar fraktalny obliczamy jako:

$$D = \frac{\log a}{\log \frac{1}{s}} \quad (27)$$

Aby scharakteryzować fakturę przy pomocy fraktali musimy również wykonać jej transformację Fouriera i obliczyć widmo energii ze wzoru (19). Na bazie tego widma możemy policzyć trzy miary charakteryzujące daną fakturę czyli trzy momenty oraz jej wymiar fraktalny. Jako zerowy moment przyjmujemy znormalizowane widmo energii w punkcie $(0,0)$, czyli:

$$m_0 = \frac{|F(0,0)|^2}{\text{powierzchnia obrazu}} = \frac{E(0,0)}{\text{powierzchnia obrazu}} \quad (28)$$

Natomiast pierwszy i drugi moment jest obliczany według wzorów [5], [67]

$$m_1 = \frac{\sum \sqrt{u^2 + v^2} E(u, v)}{\text{powierzchnia obrazu}} \quad (29)$$

$$m_2 = \frac{\sum (u^2 + v^2) E(u, v)}{\text{powierzchnia obrazu}} \quad (30)$$

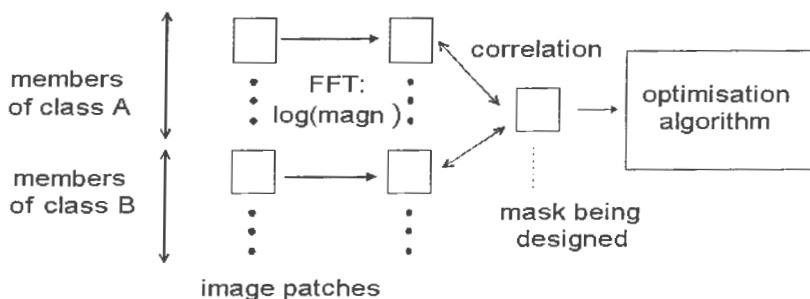
Metoda oparta na wykorzystaniu szybkiej TF do obliczania wymiaru fraktalnego [55] bazuje na fakcie, że uzyskana transformacja obrazu przypomina gaussowski szum lub proces stochastyczny. Zatem przedstawiając wzór na widmo energii (19) we współrzędnych biegunowych otrzymujemy w tym momencie zapis $E(r)$ dla wszystkich częstotliwości.

Możemy jednak połączyć wymiar fraktalny z widmem energii [12] obliczając go według poniższego wzoru:

$$D = \frac{\log E(r)}{\log r} \quad (31)$$

Interpretacja geometryczna powyższego wzoru jest następująca: jeśli na osi poziomej odkładamy $\log r$, a na osi pionowej $\log E(r)$ to nachylenie otrzymanej krzywej dla poszczególnych promieni określa wymiar fraktalny.

Podjęmowane są też prace mające na celu rozróżnianie tekstur przy pomocy algorytmów genetycznych. [15], [63] W tym podejściu najpierw należy wybrać fragmenty tekstur i dokonać ich transformacji Fouriera (FFT). Otrzymane transformaty są sprowadzane do zapisu binarnego wymaganego przez algorytm genetyczny (GA). Następnie otrzymane wyniki koreluje się z wcześniej przygotowanymi maskami charakteryzującymi określony rodzaj tekstury. Ostatnim krokiem jest wybranie odpowiedniej tekstury przez algorytm optymalizujący (i do niego właśnie używa się algorytmu genetycznego). Cały ten proces jest przedstawiony na rysunku 49.



Rysunek 49 Schemat rozpoznawania tekstur przy użyciu transformacji Fouriera i algorytmów genetycznych.

Poza klasyczną transformacją Fouriera, są też stosowane inne dwuwymiarowa transformacje częstotliwościowe, czyli różne operatory „wrażliwe” na fakturę, np. Laplasjan 3x3 lub 5x5, Gaussian 5x5, Hursta, Haralicka, lub Frei i Chena (54). Wszystkie one są przydatne przy fakturach o małej granulacji.

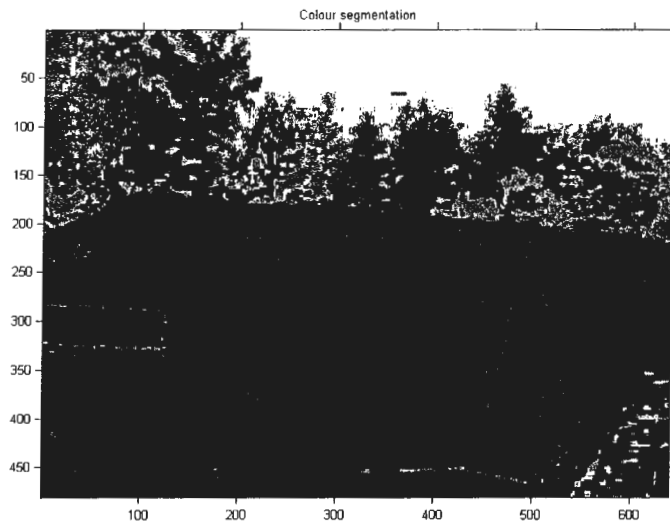
Po przeanalizowaniu powyższych metod i przyjrzeniu się fakturom, które występują na obrazach nieruchomości spróbowaliśmy opisać faktury w oparciu o analizę falkową (szczegóły matematyczne patrz rozdz. 9.6).

Zastosowanie analizy falkowej do opisu tekstur i wzorów proponowali już Balmelli i Mojsilović w 2001 [2]. Znajdowali oni cechy charakterystyczne obrazów regularnych, ornamentów roślinnych i wzorów geometrycznych i dla każdego rodzaju faktury konstruowali wektor cech dystynktywnych. Nie zaproponowali jednak żadnego połączenia swojej metody z obrazami rzeczywistymi. Natomiast w ostatnim czasie Lewis i Fauzi przygotowali automatyczną segmentację tekstur w oparciu o dyskretna TF specjalnie do celów obrazowej bazy danych. Swoją metodę zastosowali do rozpoznawania obrazów w zbiorach muzealnych [36].

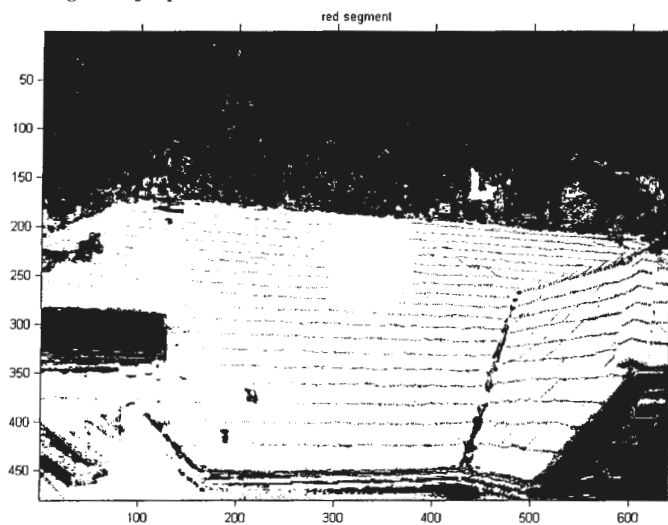


Rysunek 50 Obraz oryginalny. Dach traktujemy jako obiekt pokryty teksturą

Dzięki temu, że szybka transformacja falkowa (FWT) jest zaimplementowana w Toolboxie Wavelet programu MatLab wersja 2006b narzędzie to staje się odpowiednio wydajne do częstego użycia i dlatego postanowiliśmy zastosować je do charakterystyki faktur geometrycznych [30], [32]. Tak więc, wykonujemy dyskretną dwuwymiarową transformację falkową elementu fakturowanego wydzielonego z obrazu, otrzymując macierze współczynników falkowych czyli detali d^1 , d^2 i d^3 .



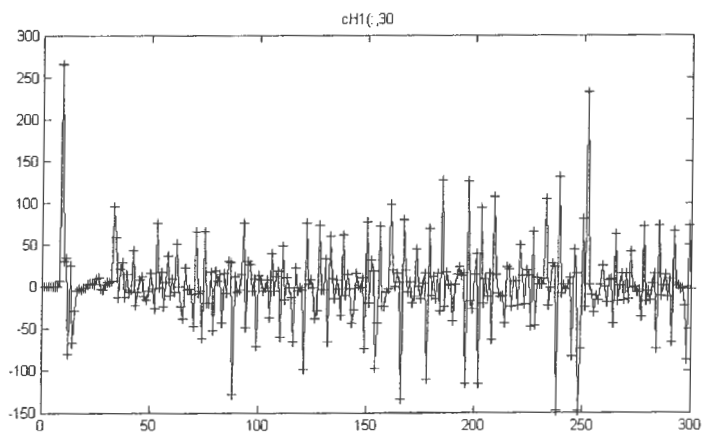
Rysunek 51 Segmentacja oparta na kolorach



Rysunek 52 Segment w kolorze jasnym czerwonym, z którego wydzielono dach.

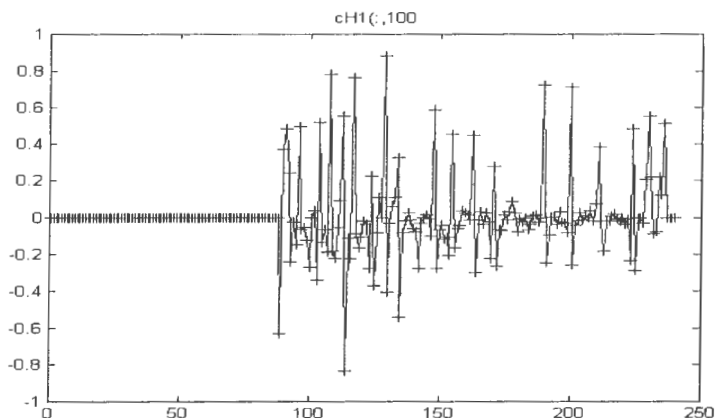
Należy zauważyć, że jedną z własności detali jest ich kierunkowość. Oznacza to, że detale horyzontalne d^1 mają duże wartości, gdy spłot falki następuje z poziomymi elementami obrazu i odwrotnie, detale wertykalne d^2 mają duże wartości, gdy spłot falki

następuje z pionowymi elementami obrazu (stąd pokazywane kierunki na rys. 85). Jeśli skorzysta się z tej cechy splatając obraz regularnych dachówek, czy cegieł z odpowiednimi falkami otrzyma się transformatę dwuwymiarową, której wartości maksymalne będą się pokrywały z połączeniami tych dachówek lub cegieł. Do wszystkich naszych obliczeń stosujemy falki Haara [21]. Są one najprostsze, ale najlepiej się sprawdzają w przypadku elementów geometrycznych ze względu na swój prostokątny kształt.



Rysunek 53 Przekrój przez detale horyzontalne dla 30-tej kolumny (falka Haara, $j=1$) dla ceglanej ściany narożnego okna pokazanej na rys. 40.

Działanie metody zostanie zilustrowane na przykładzie dwóch obrazów. Pierwszy przykład to narożne okno, dla którego bocznych ścian określimy wielkość cegieł, a drugi to dach (rys. 50), dla którego wyznaczymy wielkość dachówki. Dla lepszego zrozumienia przyjrzyjmy się teraz tylko jednej linii współczynników przechodzącego pionowo przez cegły na rys. 40. Przekrój przez 30-ty wiersz macierzy detali horyzontalnych (oznaczanych przez program MatLab jako `cH1`) pokazuje rysunek 53. Na osi odciętych odłożone są numery falek Haara dla pierwszego poziomu analizy wielorozdzielczej, natomiast na osi rzędnych - wartości współczynników `cH1`. Maksima i minima na tym wykresie to punkty pomiędzy cegłami na omawianym przez nas rysunku 40. Analogiczny wykres prezentuje rys. 53, który pokazuje przekrój przez 100-ty wiersz macierzy detali horyzontalnych dla obrazu dachu, co odpowiada 200-tnej linii pikseli na obrazie oryginalnym (patrz dolna skala na rys. 52).

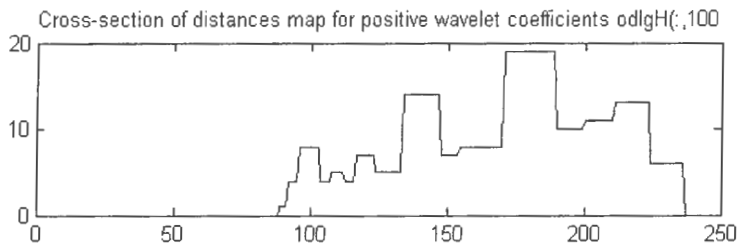


Rysunek 54 Przekrój przez detale poziome dla 100-tej kolumny (falka Haara, $j=1$) dla dachu pokazanego na rys. 50.

Można teraz stworzyć wykres odległości pomiędzy tymi maksymalnymi lub minimalnymi wartościami i okazuje się, że odległości (które odpowiadają odległościom między cegłami lub dachórkami) stają się dobrym parametrem wyróżniającym ten właśnie obszar (rys. 55). Taki wykres odległości pomiędzy wartościami maksymalnymi z rys. 54 jest pokazany na rysunku 56. Próg odcięcia jest przyjęty dla 2% wartości maksymalnej z całej macierzy. Na osi odciętych odłożone są numery falek Haara dla pierwszego poziomu analizy wielorozdzielczej, natomiast na osi rzędnych – odległości pomiędzy współczynnikami cH1 na poziomie 2%.

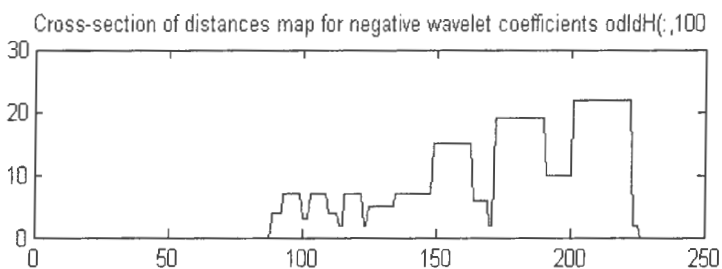


Rysunek 55 Odległości wyznaczone dla progu odcięcia 2% wartości maksymalnej dla całej macierzy detali dla wielkości cegieł dla ściany z rys. 40.



Rysunek 56 Odległości wyznaczone dla progu odcięcia 2% wartości maksymalnej dla całej macierzy detali dla wielkości dachówki z rys. 52.

Analogiczne odległości możemy policzyć dla współczynników ujemnych i są one pokazane na rys. 57.

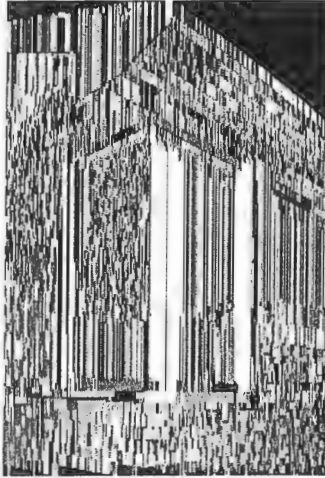


Rysunek 57 Odległości wyznaczone dla progu odcięcia 2% wartości minimalnej dla całej macierzy detali dla wielkości dachówki z rys. 52.

Teraz tworzymy mapy odległości między poszczególnymi maksymalnymi lub minimalnymi wartościami dla całej macierzy detali horyzontalnych $cH1$. Takie mapy odległości górnych i dolnych są pokazane odpowiednio na rysunkach 58, 59, 60, 61. Takie mapy dają informację o wymiarze faktury w kierunku x . Następnie analogiczne mapy tworzy się dla odległości między poszczególnymi maksymalnymi lub minimalnymi wartościami dla całej macierzy detali wertykalnych $cV1$. Te z kolei mapy dają nam informację o wymiarach faktury w kierunku y . Takie mapy odległości górnych i dolnych są pokazane odpowiednio na rysunkach 62 i 63.

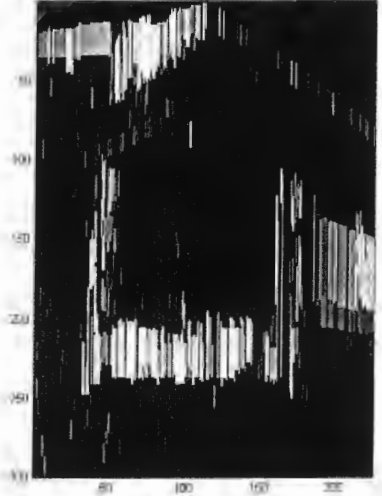
W omawianym przypadku długość cegieł została oszacowana na 5-9 długości falek dla $j=1$, czyli 10-18 pikseli obrazu oryginalnego. (Należy pamiętać, że transformaty są liczone przez podpróbkowanie.) Wykorzystanie drugiego poziomu analizy $j=2$ wielorozdzielczej nie wydaje się już w tym wypadku celowe.

Mapa odleglosci gornych dla cH1



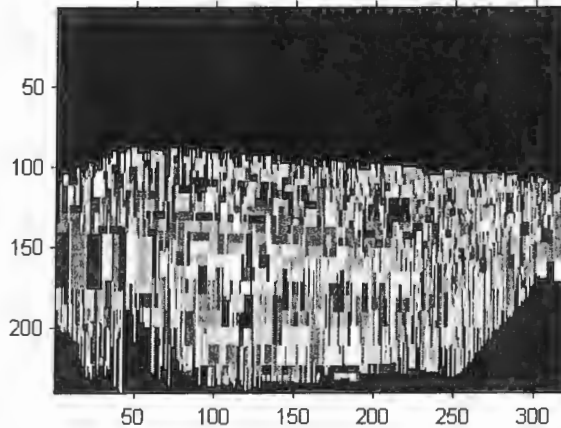
Rysunek 58 Mapa odległości górnych, czyli policzonych na podstawie maksymalnych wartości detali horyzontalnych

Mapa odleglosci dolnych dla cH1

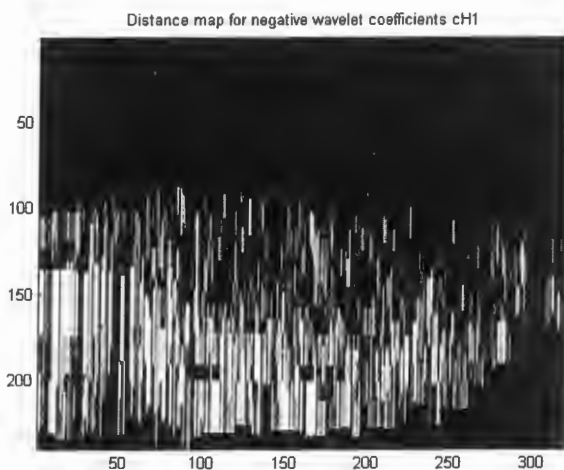


Rysunek 59 Mapa odległości dolnych, czyli policzonych na podstawie minimalnych wartości detali horyzontalnych

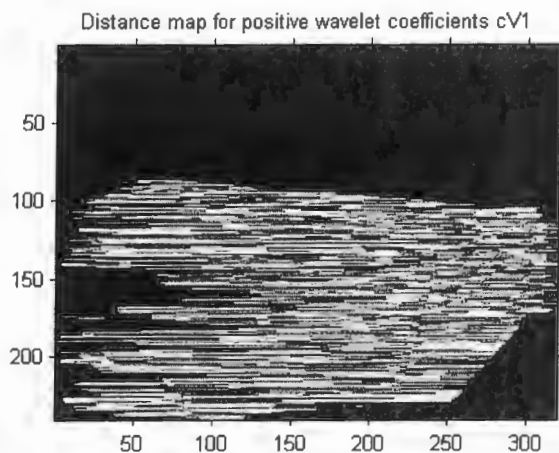
Distance map for positive wavelet coefficients cH1



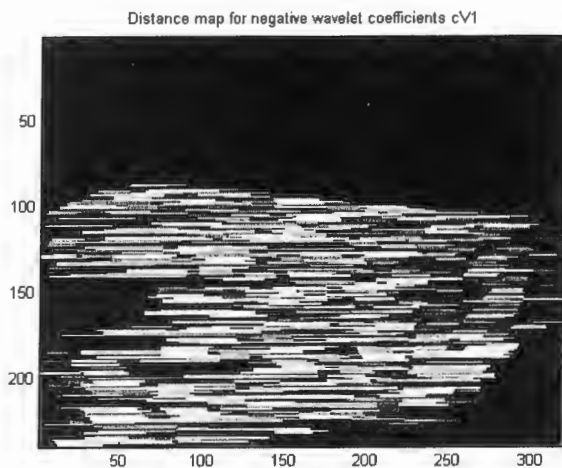
Rysunek 60 Mapa odległości górnych czyli policzonych na podstawie maksymalnych wartości detali horyzontalnych



Rysunek 61 Mapa odległości dolnych, czyli policzonych na podstawie minimalnych wartości detali horyzontalnych

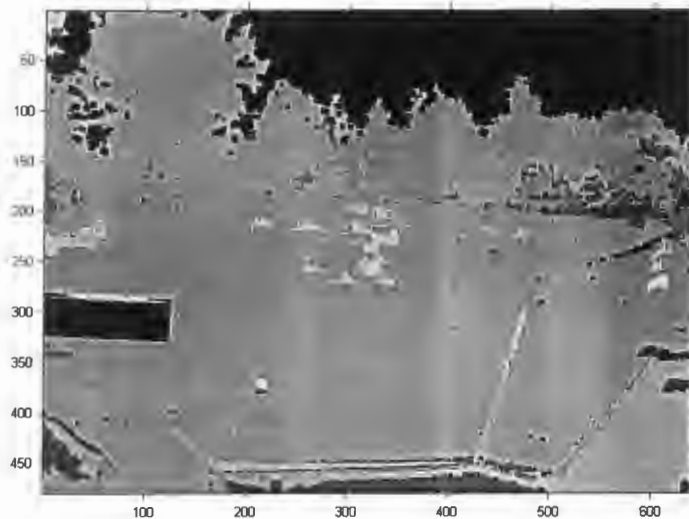


Rysunek 62 Mapa odległości górnych czyli policzonych na podstawie maksymalnych wartości detali horyzontalnych



Rysunek 63 Mapa odległości dolnych, czyli policzonych na podstawie minimalnych wartości detali horyzontalnych

W omawianym przypadku długość dachówek została oszacowana na 15-20 długości falek dla $j=1$, czyli 30-40 pikseli obrazu oryginalnego, a ich szerokość - na 22-40 falek czyli 44-80 pikseli.



Rysunek 64 Obiekty wydzielone z warstwy czerwonej i przedstawione w kolorach średnich

4.1.3 Deskrytory kształtu

Kształt obiektu jest pojęciem złożonym i trudnym do jednoznacznego opisanie. Obiekty mogą być względem siebie przesunięte, obrócone lub przeskalowane, a mimo to trzeba móc pokazać, że kształty takich obiektów są identyczne lub co najmniej podobne do siebie. Dlatego też do opisu kształtów używa się wielu różnych wielkości charakteryzujących obiekt nazywanych deskrytorami. Ponadto staramy się tak dobrać wielkości, które służą za deskrytory, aby były one niezmiennicze względem przesunięcia, obrotu i skalowania. Deskrytory możemy podzielić na dwie podstawowe grupy:

1. Deskrytory oparte na konturach bazują na informacji o obrysie kształtu. Są one popularne, ponieważ są łatwe w użyciu i w wielu zastosowaniach, również bazodanowych, zupełnie wystarczające. Deskrytory te możemy podzielić na dwie zasadnicze grupy, takie jak:
 - Deskrytory strukturalne
 - a. aproksymacje konturu wielokątami
 - b. „róża wiatrów”
 - c. wygładzanie Gaussianem
 - d. niezmienniki kształtu
 - Deskrytory konwencjonalne
 - a. obwód
 - b. przekątna równoważna
 - c. zawartość
 - d. sygnatura kształtu (*Shape Signatures*)
 - e. skałowany przestrzenny deskrytor krzywizny (*Curvature Scale Space Descriptor*)
 - f. deskrytory Fouriera
 - g. deskrytory falkowe
 - h. autoregresja
 - i. dopasowanie elastyczne
2. Deskrytory oparte na regionach biorą pod uwagę nie tylko kontur, ale każdy piksel należący do obiektu. Możemy je zatem podzielić w następujący sposób:
 - a. zawartość
 - b. liczba Eulera
 - c. momenty bezwładności
 - d. mimośród

- e. momenty Zernika
- f. momenty pseudo-Zernika
- g. momenty Legendre'a
- h. metoda gridowa

Jedną z najprostszych, acz skutecznych metod stosowanych do opisu kształtu jest kodowanie kształtu „różą wiatrów” (co proponuje Castleman [10]). Polega ona na przyporządkowaniu przybliżonemu obrysowi obiektu w postaci linii łamanej ośmiu kierunków, jak to pokazuje rysunek 65. W konsekwencji metoda to jest przydatna do indeksowania obiektów po kształcie (patrz rozdz. 6).



Rysunek 65 Reprezentacja kształtu w oparciu o "różę wiatrów"

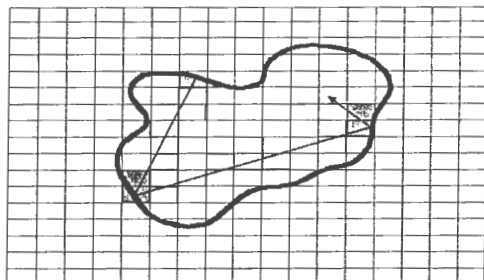
W podejściu strukturalnym kontur obiektu jest dzielony na części zwane prymitywami. Dla niektórych obiektów wystarczy zastosować jako kontur odpowiednią łamaną zamkniętą [41]. Dla innych rozróżniane są prymitywy prostoliniowe i krzywoliniowe. Dla odcinków prostych określany jest ich kierunek, a dla krzywych liczymy promień krzywizny [50]. Dodatkowo kontury są wygładzane przy pomocy filtra Gaussa [16]. Zaletą tego podejścia jest łatwość dopasowania się do obiektów o różnych skalach. Poszczególne metody w podejściu strukturalnym różnią się wyborem prymitywów i ich organizacją w reprezentacji kształtów. Jest to ich główna wada, ale nie jedyna. Problemem jest też złożoność implementacyjna oraz niejednoznaczność dopasowania dla poszczególnych obiektów.

Konwencjonalne deskryptory kształtu opisują całościowy kształt obiektu. Do podstawowych pojęć w tej kategorii należy **obwód** O , który obliczamy jako długość konturu każdego obiektu i **powierzchnia** A liczona jako suma wszystkich pikseli wewnątrz konturu wraz z tym konturem.

Kiedy mamy powierzchnię można policzyć **przekątną równoważną** d rozumianą jako przekątną koła o powierzchni danego obiektu A według wzoru

$$d = \sqrt{\frac{4A}{\pi}} \quad (32)$$

Ciekawą metodę opisu kształtu nazywaną **sygnaturą kształtu** (*shape signatures*). zaproponowali Zauhar i Mayona z Uniwersytetu w Filadelfii [69]. Traktują oni kontur obiektu jako lustro, we wnętrzu którego odbija się śledzony promień światła tworząc triangulację tego obiektu. Zachowana jest też zasada, że kąt padania promienia jest równy kątowi jego odbicia. Cała idea jest pokazana na rys. 66, a triangulacja może być mniej lub bardziej gęsta.



Rysunek 66 Idea wyznaczania sygnatury kształtu.

Następną metodę opisu konturu – **skalowany przestrzenny deskryptor krzywizny** (*Curvature Scale Space Descriptor*) - niezmienniczą względem przekształcenia afinicznego zaproponowali Abbasi i Kittler na Uniwersytecie w Surrey w Wielkiej Brytanii [44]. Analizują oni krzywiznę konturu wpisując w ten kontur okręgi ściśle styczne, a potem opisują wypukłości i wklęsłości konturu licząc punkty, w których zeruje się druga pochodna. W celu znalezienia okręgu ściśle stycznego do krzywej różniczkujemy dwukrotnie obie strony równania okręgu:

$$r^2 = (x - X)^2 + (y - Y)^2 \quad (33)$$

uważając promień r okręgu oraz współrzędne jego środka X i Y za parametry. Otrzymujemy wówczas:

$$(x - X) + (y - Y)y' = 0 \quad (34)$$

$$y''(y - Y) = -(1 + y'^2) \quad (35)$$

Z powyższych trzech równań wstawiamy na miejsce y prawą stronę równania krzywej $y = y(x)$, do której okrąg jest ściśle styczny, a następnie z równań (33), (34) i (35) wyznaczamy współrzędne środka tego okręgu

$$X = x - \frac{1 + y'^2}{y''} y', \quad Y = y + \frac{1 + y'^2}{y''} \quad (36)$$

oraz jego promień

$$r = \frac{(1 + y'^2)^{\frac{3}{2}}}{|y''|} \quad (37)$$

Jeśli użyjemy parametru $t = 0, 1, \dots, N-1$ ($x=x(t)$ i $y=y(t)$) zakładając, że jest to dyskretny podział konturu na fragmenty krzywej, to możemy zapisać równania (36) i (37) w postaci parametrycznej:

$$X = x(t) - \frac{x'^2(t) + y'^2(t)}{x'(t)y''(t) - x''(t)y'(t)} y'(t), \quad Y = y(t) + \frac{x'^2(t) + y'^2(t)}{x'(t)y''(t) - x''(t)y'(t)} x'(t) \quad (38)$$

oraz

$$r(t) = \frac{[x'^2(t) + y'^2(t)]^{\frac{3}{2}}}{|x'(t)y''(t) - x''(t)y'(t)|} \quad (39)$$

gdzie:

$$\frac{dy}{dx} = \frac{dy'(t)}{dx'(t)} \quad i \quad \frac{d^2y}{dx^2} = \frac{x'(t)y''(t) - x''(t)y'(t)}{x'^3(t)} \quad (40)$$

Ponieważ krzywizna jest odwrotnie proporcjonalna do promienia, otrzymujemy parametryczne równanie krzywizny w postaci:

$$k(t) = \frac{1}{r(t)} = \frac{x'(t)y''(t) - x''(t)y'(t)}{[x'^2(t) + y'^2(t)]^{\frac{3}{2}}} \quad (41)$$

Następnie punkty zerowania się drugiej pochodnej są zaznaczane na krzywej konturu i iteracyjnie wygładzane przy użyciu wygładzającego filtra Gaussa:

$$G(t, \sigma) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{r^2(t)}{2\sigma^2}\right) \quad (42)$$

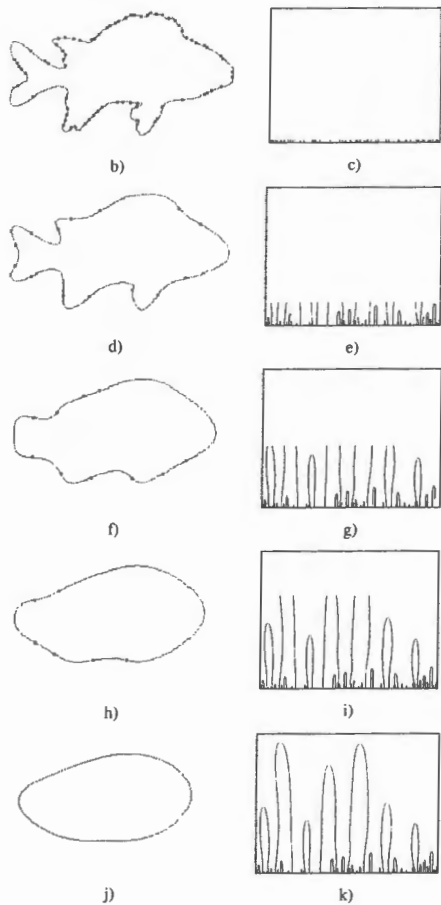
Poszczególne iteracje liczone są ze splotu

$$x_\sigma(t) = x(t) * G(t, \sigma), \quad y_\sigma(t) = y(t) * G(t, \sigma) \quad (43)$$

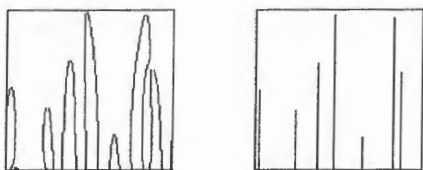
a rosnące odchylenie standardowe σ traktowane jako współczynnik skali daje coraz gładszy kształt. Nowe punkty zerowania się drugiej pochodnej są nanoszone na poszczególne kontury w poszczególnych skalach. Proces ten jest zilustrowany na rysunkach 67 b), d), f) h), j). W praktyce nie stosuje się tylko całkowitych wartości σ dzięki czemu można uzyskać mapę konturu, czyli wykres na płaszczyźnie (t, σ) . Poszczególne etapy tworzenia takiej mapy są pokazane na rysunkach 67 c), e), g), i), k).



a)



Rysunek 67 a) obraz oryginalny, b), d), f), h), j) iteracyjne wygładzanie kształtu przy zmianach odchylenia standardowego w operatorze Gaussa odpowiednio dla $\sigma = 1, 4, 7, 10, 12, 14$, c), e), g), i), k) poszczególne etapy tworzenia mapy konturu. [56]



Rysunek 68 Mapa konturu (po lewej) i wyekstrahowane z niej piki (po prawej)

Na mapie konturu wyznacza się tylko występujące lokalne maksima i nanosi się je na nowy wykres zaznaczając w formie pików (rys. 68), które w dalszej części bazy danych służą jako indeksy, gdy indeksujemy obrazy według kształtu obiektów.

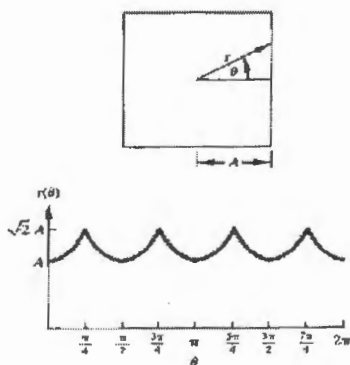
Następnym deskryptorem kształtu jest **deskryptor** oparty na współczynnikach transformacji **Fouriera**. Kształt obiektów na podstawie tych współczynników zaczęli porównywać Zhang i Lu od 2002 [70]. Analizowali oni krzywiznę konturu w zależności od środka obiektu. W pierwszym kroku liczyli promień konturu w zależności od środka ciężkości obiektu. **Środek ciężkości** obiektu (\bar{x}, \bar{y}) obliczają jako

$$\bar{x} = \frac{1}{n} \sum_x x \quad \text{i} \quad \bar{y} = \frac{1}{n} \sum_y y, \quad (44)$$

a następnie wyznaczają funkcję odległości $r(\theta)$ od tego środka z krokiem $\theta = 0, 1, \dots, N-1$, gdzie $N-1$ jest równoważne kątowi 2π .

$$r(\theta) = \sqrt{[x(\theta) - \bar{x}]^2 + [y(\theta) - \bar{y}]^2} \quad (45)$$

Przykładowa funkcja odległości dla konturu kwadratu jest pokazana na rys. 69.



Rysunek 69 Przykładowy kształt i jego rozwinięcie w funkcję odległości konturu od środka tego obiektu.

Współczynniki Fourier a_n są niezmiennicze względem przesunięcia i rotacji opisywanego obiektu, są też niezależne od miejsca na konturze, od którego zaczniemy opisywać funkcję $r(\theta)$.

$$a_n = \frac{1}{N} \sum_{\theta=0}^{N-1} r(\theta) \exp\left(\frac{-j2\pi n\theta}{N}\right) \quad (46)$$

Po normalizacji wszystkich współczynników dostajemy wektor $\mathbf{a} = \{a_1, a_2, \dots, a_N\}$, który jest używany przy indeksowaniu jako wyróżnik danego kształtu.

Falkowy deskryptor kształtu jest oparty na podobnej zasadzie jak deskryptor Fouriera, z tą jedynie różnicą, że transformacja funkcji odległości $r(\theta)$ jest dokonywana w przestrzeni falkowej, czyli zgodnie ze wzorem (100) otrzymamy następujące współczynniki falkowe:

$$d_j(k) = 2^{\frac{j}{2}} \sum_{-\infty}^{+\infty} r(\theta) \psi_{j,k}(\theta) \quad (47)$$

dla interesującego nas poziomu j analizy wielorozdzielczej.

Kolejnym mechanizmem zaprzęgniętym do opisu kształtu jest **autoregresja**. Metoda ta również używa jednowymiarowej funkcji kształtu, ale jako modelu procesu stochastycznego. Parametry tego procesu są deskryptorami kształtu. Ogólna postać zamkniętego modelu autoregresyjnego jest następująca:

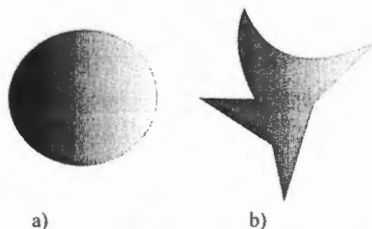
$$r_i = \alpha + \sum_{j=1}^m \sigma_j r_{i-j} = \sqrt{\beta} \omega_i \quad (48)$$

gdzie: $\{\alpha, \beta, \sigma_1, \dots, \sigma_m\}$ są współczynnikami modelu estymowanymi przez $\sqrt{\beta} \omega$, modelującymi biały szum i α jest proporcjonalna do średniej wartości funkcji kształtu. Model autoregresyjny nie jest zawsze wystarczającym deskryptorem szczególnie do opisu kształtów nie wypukłych, jednakże pod względem odporności na zakłócenia wydaje się być odporniejszy od metody współczynników Fouriera [34].

Ostatnią z metod opartą na opisie konturu, a tu wspomnianą jest dopasowanie elastyczne oparte na aktywnych konturach. W tym przypadku aktywny kontur dopasowuje się do obiektu, a deskryptorami są rozwiązania równania (95).

Deskryptory oparte na regionach opisują całą powierzchnię obiektu. Najprostszym parametrem jest **zwartość** rozumiana jako stosunek kwadratu obwodu do powierzchni, co ilustruje rys. 70.

$$zw = \frac{(O)^2}{A} \quad (49)$$



Rysunek 70 a) przykład zwartego kształtu, b) przykład mało zwartego kształtu

Następnym parametrem jest **liczba Eulera** E określająca jednorodność obiektu, czyli ile jest w nim dziur: $E = C - H$, gdzie C jest to liczba obiektów w rejonie, a H liczba dziur.

Kolejne są **centralne momenty bezwładności** obliczane jako suma po wszystkich pikselach: $m_{ij} = \sum_{k=1}^n (x_k - \bar{x})^i (y_k - \bar{y})^j$, gdzie i, j - rząd momentu względem odpowiednio osi x i y . Na podstawie momentów centralnych możemy obliczyć inne momenty bezwładności, które są niezmiennicze względem przesunięcia, skali i orientacji w przestrzeni co zapewnia nam dobrą reprezentację obiektu i łatwą jego rozpoznawalność. Przykładowe trzy liczymy następująco:

$$\begin{aligned} M_1 &= m_{20} + m_{02} \\ M_2 &= (m_{20} - m_{02})^2 + 4m_{11}^2 \\ M_3 &= (m_{30} - 3m_{12})^2 + (3m_{21} - m_{03})^2 \end{aligned} \quad (50)$$

Jako deskryptorów kształtu używa się od trzech do dziewięciu momentów bezwładności.

Następną cechą charakteryzującą obiekt jest **mimośród**, czyli stosunek odległości między ogniskami elipsy do jej większej osi. Na podstawie momentów mimośród można policzyć według wzoru:

$$e = \frac{m_{20} + m_{02} + \sqrt{(m_{20} - m_{02})^2 + 4m_{11}^2}}{m_{20} + m_{02} - \sqrt{(m_{20} - m_{02})^2 + 4m_{11}^2}} \quad (51)$$

Kolejnym parametrem kształtu obiektu jest kierunek nachylenia **osi bezwładności** do poziomu:

$$\theta = \frac{1}{2} \tan^{-1} \left(\frac{2m_{11}}{m_{20} - m_{02}} \right) \quad (52)$$

Natomiast Teague [61] proponuje użycie momentów ortogonalnych do reprezentacji obrazu w oparciu o teorię wielomianów ortogonalnych i wprowadzonych przez Zernika momentów zespolonych. Zapewniają one możliwość konstruowania momentu dowolnie wysokiego rzędu opisanego jako:

$$V_{mn}(x, y) = V_{mn}(\rho \cos \theta, \rho \sin \theta) = R_{mn}(\rho) \exp(jm\theta) \quad (53)$$

gdzie

$$R_{mn}(\rho) = \sum_{s=0}^{\lfloor \frac{n-|m|}{2} \rfloor} (-1)^s \frac{(n-s)!}{s! \left(\frac{n+|m|}{2} - s \right)! \left(\frac{n-|m|}{2} - s \right)!} \rho^{n-2s} \quad (54)$$

gdzie ρ jest promieniem punktu o współrzędnych (x, y) od środka kształtu, θ jest kątem nachylenia promienia do współrzędnej x , a m i n są liczbami całkowitymi takimi, że różnica

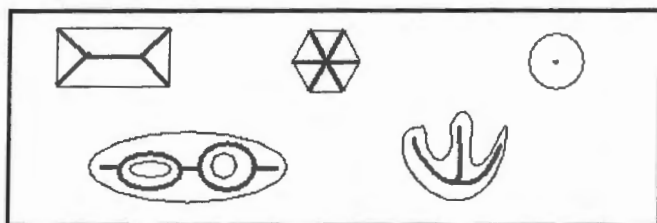
$n-|m| = 2k$, dla $|m| \leq n$. Wielomiany Zernike'a są zbiorem zupełnym ortogonalnych funkcji zespolonych nad dyskiem jednostkowym takim, że $x^2 + y^2 = 1$. Zatem zespolone momenty Zernike'a rzędu n z m powtórzeniami są definiowane jako:

$$A_{nm} = \frac{n+1}{\pi} \sum_x \sum_y f(x, y) \bar{V}_{nm}(x, y), \quad x^2 + y^2 \leq 1 \quad (55)$$

gdzie: \bar{V} - liczba sprzężona.

Ponieważ funkcje Zernike'a przyjmują za swoją dziedzinę dysk jednostkowy, to dysk ten musi być określony zanim zaczniemy liczyć momenty, dlatego standardowo normalizuje się kształt, a środek dysku jednostkowego jest sytuowany w środku ciężkości obiektu. To powoduje, że momenty te są niezmiennicze względem skali i przesunięcia.

Kolejną metodą rozpoznawania kształtu jest szkieleutowanie, które wywodzi się z grupy operacji morfologicznych. Jeśli przeprowadzimy kilka etapów erozji, to po około 10 iteracjach (zależnie od wielkości obiektu) otrzymamy szkielety kształtu mogące służyć jako deskryptor.



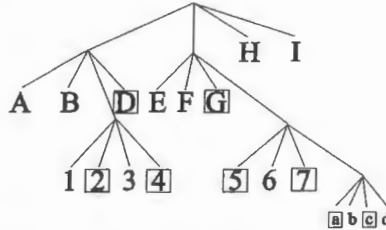
Rysunek 71 Przykłady szkieleutowania dla różnych kształtów

4.2 Przestrzenne modele danych

Przestrzenne modele danych to modele tworzone na potrzeby analizy obrazu na poziomie cech logicznych. Pierwszą z takich cech jest wskazanie położenia obiektu w całym obrazie. Jedną z proponowanych w literaturze metod jest reprezentacja lokalizacji obiektu w postaci drzewa. W metodzie tej obraz dzielimy na połowy, a te połowy jeszcze na połowy i otrzymujemy coraz mniejsze regiony. Każdy podział odpowiada jednemu poziomowi struktury drzewa. Następnie zaznaczamy, który z regionów jest zajęty przez opisywany obiekt. Ilustrację tego procesu widać na rysunku 72.



A	B	E	F
1 2	D	G	5 6
3 4		7	a b c d
H		I	



Rysunek 72 Zaznaczony obiekt (dach) jest odwzorowywany na „mapie” regionów, a następnie jest zapisywany w postaci drzewa z zaznaczonymi przez kwadraty elementami, które wyznaczają położenie obiektu w całym obrazie.

Do opisu położenia obiektów wzajemnie między sobą jest używany wektorowy model danych, zawierający takie elementy jak: punkt, odcinek, łamana i poligon.

Wybór cech charakteryzujących obraz jest szczególnie istotny najpierw przy indeksowaniu bazy danych, a potem przy przeszukiwaniu tej bazy i przy określaniu zapytań użytkownika do bazy. Ważne jest również, żeby zbiór cech opisujących obraz miał swoje odzwierciedlenie w graficznym interfejsie użytkownika służącym do zadawania pytań do bazy danych.

ID – obiektu	
Procent podobieństwa dla każdej cechy (np. 10% podobieństwa obiektu ID – obiektu dla cechy 5)	
Wskaźnik do tab. 6 dla każdego obiektu	

↓

↓

6. Tabela rozpoznanych obiektów	
ID - obiektu	Nazwa obiektu (np. okno)
Wskaźnik do tab. 7 dla każdego obiektu	

↓

↓

7. Tabela istniejących przypadków	
Nazwa obrazu	
ID – obiektu	
Nazwa obiektu	
Wskaźnik do dodatkowych koniecznych tabel	Wskaźnik do tab. 1 dla każdego obiektu żeby znaleźć jego cechy

↓

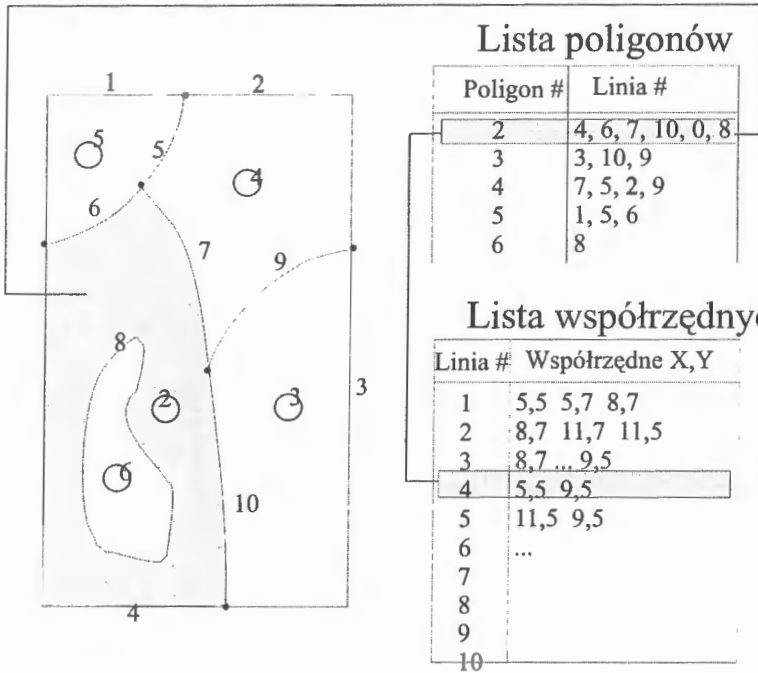
↓

↓

↓

①

Natomiast struktura przestrzenna będzie opisana relacjami przestrzennymi udostępnianymi w systemie bazodanowym Oracle. Przykład takiego opisu pokazuje rys. 73. Każdy obiekt taki jak punkt, linia i poligon ma swój unikalny identyfikator i współrzędne wierzchołków, zapewniające określenie relacji przestrzennych pomiędzy tymi obiektami.



Rysunek 73 Przykładowa organizacja topologii w bazach danych

6. Indeksowanie danych

Indeksowanie danych jest problemem istniejącym w każdej bazie danych i dobór odpowiednich mechanizmów zapewnia efektywny dostęp do danych i przyspiesza ich wyszukiwanie. W przypadku obrazowych baz danych tradycyjne mechanizmy nie znajdują zastosowania, ale metody indeksowania nie są też jednakowe i uniwersalne dla wszystkich obrazowych baz danych. Mechanizmy używane do indeksowania baz obrazowych, lub szerzej rozumianych baz multimedialnych są ściśle zależne od struktur danych przyjętych w momencie tworzenia takich baz. Oznacza to, że sposób opisu obiektów i całych obrazów determinuje już na wstępie metody indeksowania, które mogą być później zastosowane. Podobnie jest też z rodzajami zapytań, które taka baza ma obsłużyć. Sposoby indeksowania muszą być też dostosowane do odpowiedniej realizacji zapytań użytkownika i stworzonego interfejsu.

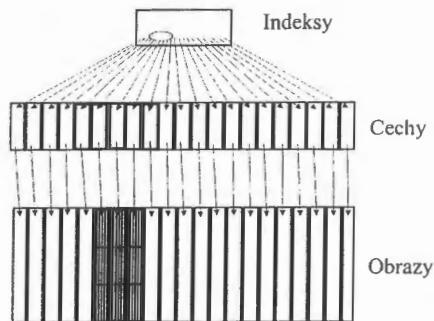
Jak pokazuje to schemat przetwarzania obrazu na rys. 1 wyniki każdego etapu przetwarzania są składowane w bazie danych. Zatem dla danych z każdego etapu muszą być inne struktury indeksowania. Cechy pierwotne są indeksowane na podstawie wektorów cech. Przestrzenne cechy logiczne są indeksowane przy pomocy np. R-drzew, natomiast cechy z najwyższego poziomu abstrakcji są indeksowane według reguł wiedzy.

Weźmy pod uwagę przypadek, w którym zapytanie do bazy dotyczy podobieństwa cech pierwotnych. Takie zapytanie jest definiowane jako odległość w wielowymiarowej przestrzeni cech (patrz rozdz. 8.1.), tak więc indeksowanie musi dotyczyć cech obrazu lub cech poszczególnego obiektu. Najlepiej sprawdzone wydają się dwie metody: filtrowanie [1] i indeksowanie metodą R-drzew [39]. Obiecująco wyglądają też metody oparte na transformacjach: θ -transformacji [37] i transformacji falkowej [65].

6.1. Filtrowanie

W przypadku filtrowania struktura danych jest hierarchiczna i zawiera trzy poziomy:

- cały obraz
- wydzielone cechy obrazu lub obiektu
- indeksy

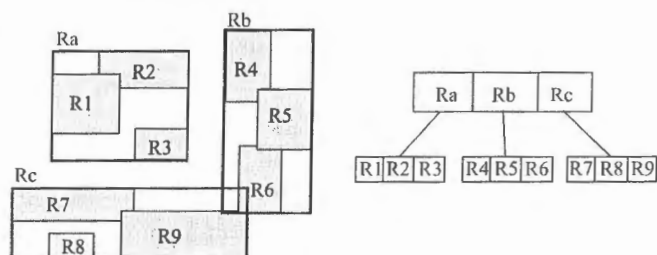


Rysunek 74 **Struktura indeksu obrazowej bazy danych. Szare pola na poziomie cech pokazują obszar poszukiwań, a ciemne pola to wyszukane obrazy.**

Najniższym poziomem jest oryginalny obraz. Służy on do wyboru z niego cech do drugiej warstwy i dalej do interfejsu użytkownika. Druga warstwa reprezentuje cechy obrazu i właśnie w niej przechowywany jest wektor cech, który z natury jest znacznie mniejszy niż obraz. Istnieje też sytuacja, gdy z obrazu wydzielone są obiekty, a dopiero cechy dotyczą samych obiektów. W takiej sytuacji wektory cech związane są z obiektami. Dopiero na bazie warstwy z wektorami cech liczone są dalej miary podobieństwa pomiędzy obrazami. W celu zmniejszenia liczby przeglądanych wektorów cech tworzy się indeksy. One to właśnie dają możliwość eliminacji mało prawdopodobnych dopasowań. Strukturę takiego indeksowania prezentuje rysunek 74.

6.2. R-drzewa

Następną metodą indeksowania i zarządzania danymi przestrzennymi są R-drzewa zaprojektowane przez Guttmana w 1984 r. [20]. Pierwotna propozycja dotyczyła obiektów prostokątnych, ale następnie została rozszerzona na obiekty w przestrzeniach wielowymiarowych. Najprostsza struktura R-drzew przypomina B-drzewa. Mówiąc ogólnie R-drzewa są hierarchiczną strukturą danych służącą do dynamicznej organizacji zbiorów wielowymiarowych obiektów geometrycznych. Obiekty są reprezentowane przez najmniejsze otoczenia prostokątne. Każdy węzeł R-drzewa odpowiada najmniejszemu prostokątowi otaczającemu jego dzieci. Liście drzewa zawierają wskaźniki do bazy danych obiektów, zamiast wskaźników na węzły pochodne. Należy zaznaczyć, że prostokąty otaczające różne węzły mogą na siebie zachodzić, tak jak to jest pokazane na rysunku 75.



Rysunek 75 Przykład struktury R-drzewa i jego reprezentacja przestrzenna (po lewo)

Zasady tworzenia R-drzewa są następujące:

- liście są ustawiane na tym samym poziomie
- każdy liść zawiera informację o parze elementów (R, O) , gdzie R to prostokąt, a O to obiekt objęty tym prostokątem
- każdy węzeł zawiera parę elementów (R, P) , gdzie P jest wskaźnikiem do dziecka, a R jest prostokątem zawierającym prostokąt z tym dzieckiem
- każdy węzeł (z wyjątkiem korzenia) R-drzewa klasy (m, M) zawiera od m do M par, gdzie $m \leq \lceil M/2 \rceil$. Korzeń zawiera co najmniej dwie pary, jeśli to nie są liście.

Rys. 21 przedstawia przykładowe obiekty po lewej stronie i odpowiadające im R-drzewo po prawej. Prostokąty $R1 \div R9$ są przechowywane jako liście, natomiast prostokąty Ra , Rb i Rc są traktowane jako węzły pośrednie.

Zapytania do bazy z indeksami opartymi na R-drzewach są bardzo efektywne dzięki bliskości obiektów w strukturze R-drzewa. Jedną z cech, dla której indeksowanie przy pomocy R-drzew jest popularne jest wszechstronność możliwości zapytań do bazy. Tak więc możemy zażądać odpowiedzi dotyczącej:

- topologii (znajdź wszystkie obiekty pokrywające się lub przecinające z danym obiektem)
- kierunku (znajdź wszystkie obiekty leżące na północ (w górę) o danego obiektu)
- odległości (znajdź wszystkie obiekty leżące w mniejszej odległości od danego obiektu niż...)
- sąsiedztwa (znajdź najbliższego sąsiada danego obiektu)

6.3. Indeksowanie w przestrzeni transformat

W przypadku indeksowania baz danych w przestrzeni transformat każdy obraz poddaje się na wstępie wybranej transformacji, a następnie w indeksie umieszcza się współczynniki tej transformaty, zakładając że identyfikują one jednoznacznie transformowany obraz.

6.3.1. Indeksowanie w oparciu o θ -transformację

Jeżeli do indeksowania wybierzemy θ -transformację, to opieramy się na zapisie dwójkowym poziomu szarości punktów. Niech $N = 2^n$ będzie liczbą pikseli, a p - N -wymiarowym wektorem, takim że $p = (p_0, p_1, \dots, p_{N-1})$. Jeśli poziom szarości piksela oznaczymy jako p_x , gdzie $x \in \{0, 1, \dots, N-1 = 2^n-1\}$, wtedy x możemy zapisać w systemie dwójkowym jako:

$$x = x_n 2^{n-1} + \dots + x_2 2^1 + x_1 2^0 \quad (56)$$

Niektóre ze współczynników mogą być zerami, więc je pomijamy i otrzymujemy zapis:

$$x = x_{i_1} x_{i_2} \dots x_{i_m} = 2^{i_n-1} + \dots + 2^{i_2-1} + 2^{i_1-1} \quad (57)$$

Generuje on następujące przekształcenie: $p_m^{i_1 i_2 \dots i_m} = p_x = p_{2^{i_n-1} + \dots + 2^{i_2-1} + 2^{i_1-1}}$, gdzie $1 \leq i_1 < i_2 < \dots < i_m \leq n$. Jeśli wektor p jest reprezentowany w układzie x -współrzędnych jako $p(x)$, to w układzie θ -współrzędnych może być reprezentowany jako $p(\theta)$. Kiedy wektor opisujący piksel jest transformowany z jednego układu współrzędnych do drugiego, w obu układach reprezentuje ten sam abstrakcyjny wektor.

Funkcja F jest funkcją transformującą dla θ -transformacji, a funkcja G jest funkcją odwrotną do F . Zatem:

$$FG = GF = I \Rightarrow G = F^{-1}$$

θ -transformacja przekształca wektor $p(x) = (p_0, p_1, \dots, p_{N-1})$ we współrzędnych x na wektor $p(\theta) = (\theta_0, \theta_1, \dots, \theta_{N-1})$, lub w zapisie analogicznym do równania (57):

$$p(\theta) = (\theta_0; \theta_1^1, \dots, \theta_1^n; \theta_2^2, \dots, \theta_2^{n-1, n}; \theta_3^{23}, \dots, \theta_n^{2 \dots n}) \text{ we współrzędnych } \theta.$$

Gdy F jest funkcją transformującą, to θ -współczynniki są określone przez wektor:

$$F[p(x)] = \theta_0 + \sum_i \theta_1^{i_1} x_{i_1} + \sum_i \theta_2^{i_1 i_2} x_{i_1} x_{i_2} + \sum_i \theta_3^{i_1 i_2 i_3} x_{i_1} x_{i_2} x_{i_3} + \dots + \theta_n^{2 \dots n} x_1 x_2 \dots x_n \quad (58)$$

Natomiast transformacja odwrotna jest określona przez funkcję G : $p(x) = G\{F[p(x)]\}$.

6.3.2. Indeksowanie w oparciu o transformację falkową

Przy indeksowaniu bazy danych można też używać bezpośrednio dwuwymiarowej transformacji falkowej do obrazów kolorowych w kilku poziomach rozdzielczości j . Liczba przechowywanych poziomów transformaty zależy *de facto* od wielkości bazy danych. Jeśli $j=1,2$ nie identyfikuje nam jednoznacznie obrazów, bo jest ich zbyt dużo, wtedy trzeba wprowadzić i przechowywać kolejne poziomy. W zależności od rodzajów obrazów przechowywanych w bazie można pokusić się o dobranie rozróżniającego te obrazy rodzaju falek. Wang [65] opisuje eksperyment w swojej bazie z falkami Haara i Daubechies dla kolorowych obrazów naturalnych.

7. Przeszukiwanie danych – zapytania do bazy danych

Przeszukiwanie obrazowych baz danych w oczywisty sposób zależy od celu do jakiego baza ta została stworzona. Jest on też uzależniony od przyjętego modelu danych. Jeśli zostało zaimplementowane tylko przeszukiwanie i indeksowanie po cechach pierwotnych tj.: kolor, tekstura, krawędzie itp. to niemożliwa będzie w tej bazie analiza przestrzenna lub badanie relacji między obiektami typu: sąsiedztwo. Natomiast jeśli użyto struktury typu R-drzewa wówczas będzie można analizować wzajemną relację obiektów na obrazie.

Bazy obrazowe nie mogą istnieć w oderwaniu od innych informacji kontekstowych dotyczących bytów przedstawionych na obrazach. Tak na przykład, baza obrazów medycznych nie może istnieć niezależnie od szerokiej informacji o pacjencie i jego chorobach i przeprowadzonym leczeniu itp. [19] Zatem poza przeszukiwaniem bazy pod kątem obrazów, musi istnieć możliwość wyszukiwania tekstowego w klasycznej jego formie. Można też wprowadzać meta-dane, które łączą w sobie informację o tekście i obrazie.

Taką strukturę zastosowano w przypadku bazy CHABOT zaprojektowanej na uniwersytecie w Berkeley [47]. Jest ona obiektowa, o złożonych typach danych. Indeksy mogą być definiowane przez użytkownika ponieważ w bazie zaimplementowano dane o strukturze B-drzew i R-drzew. Dla każdego obrazu jest też zapisany na dysku jego kolorowy histogram.

Dostępny jest też przyjazny interfejs użytkownika, gdyż na ekranie pojawiają się zarówno obrazy w wybranej przez użytkownika rozdzielczości, jak i informacja tekstowa dotycząca tego obrazu. Zapewniono też szybkie przeszukiwanie obrazów w niskiej rozdzielczości i szybki dostęp do wybranego obrazu w wysokiej rozdzielczości.

Jeśli przyjrzymy się rysunkowi 1 to okazuje się, że „Graficzny interfejs użytkownika” jest podstawowym narzędziem do zadawania zapytań do bazy. Musi mieć on zatem możliwość łatwego pokazania użytkownikowi o jakie cechy może zapytać. Zatem musi istnieć możliwość wyboru (np. przesunięcia suwakiem) koloru, kształtu, tekstury, lub typu szukanego elementu, jego atrybutów, itp. W niektórych bazach istnieje nawet możliwość zadawania pytań graficznie, czyli naszkicowania interesującego użytkownika kształtu lub podania kształtu przykładowego [17]. Można też podać obraz o interesującym użytkownika histogramie kolorów i wtedy baza będzie poszukiwać obrazów o analogicznym rozkładzie kolorów, itd.

Jako odpowiedź na zapytanie użytkownika otrzymuje się wynik przeszukiwania bazy, czyli konkretny obraz, lub zbiór obrazów. To, na ile ten obraz spełnia oczekiwania użytkownika zależy już od następnego modułu, wyspecyfikowanego na rys. 1, jako „Modułu dopasowania obrazów”.

Miary podobieństwa występujące w literaturze można podzielić na trzy kategorie:

- miary oparte na metrykach
- miary oparte na zbiorach rozmytych
- miary oparte na łącznym rozkładzie prawdopodobieństwa

7.1 Miary oparte na metrykach

Takie najprostsze zapytanie o kolor opiera się na porównywaniu k -wymiarowych (64-256) kolorowych histogramów. Jeśli histogram q jest wskazany przez użytkownika, to jest mierzona jego odległość w przestrzeni cech (w tym wypadku kolorów) do wszystkich histogramów r w bazie danych, jako miara euklidesowa:

$$d = \sqrt{\sum_{i=0}^k (r_i - q_i)^2} \quad (59)$$

gdzie: r_i i q_i są to k -elementowe wektory, których i -ty element jest numerem koloru [59]. Do odpowiedzi system wybiera z bazy obraz o minimalnej odległości d .

Równie dobrze możemy oprzeć się na metryce miasta

$$D_4((i, j), (h, k)) = |i - h| + |j - k| \quad (60)$$

lub na metryce szachownicy:

$$D_8((i, j), (h, k)) = \max\{|i - h| + |j - k|\} \quad (61)$$

W trochę bardziej zaawansowanym porównywaniu [7] uwzględnia się nieliniowość modelu barw RGB wprowadzając symetryczną macierz wag $A = [a_{ij}]$ z przedziału $[0,1]$ opartą na L_2 -normie. Stąd odległość między podobnymi obrazami liczymy według następującego wzoru [22]:

$$d_{hist}^2(r, q) = (r - q)^T A (r - q) \quad (47)$$

W większości przypadków użytkownicy są zainteresowani zapytaniem o indywidualne obiekty, które trzeba wyselekcjonować z obrazu (patrz rozdz. 3.4) i być w stanie porównać je z podobnymi obiektami na innych obrazach [1]. Porównania te przeprowadza się w przestrzeni cech. Dla każdego obiektu O każda cecha F_i (tj. kolor, tekstura, deskryptor kształtu, itp. (patrz rozdz. 4.1)) ma określoną wartość liczbową, którą oblicza się wcześniej i

przechowuje w bazie. Rozpatrujemy więc N -wymiarową przestrzeń cech $F_1 \times \dots \times F_i \times \dots \times F_N \in \mathbf{R}^n$, w której mamy znaleźć podobieństwo obiektów O_i i O_j . Określamy metrykę ϱ jako funkcję $\varrho: O_i \times O_i \rightarrow \mathbf{R}^+$ spełniającą następujące warunki:

1. $\varrho(O_i, O_i) = 0$
2. $\varrho(O_1, O_2) = \varrho(O_2, O_1)$
3. $\varrho(O_1, O_3) \leq \varrho(O_1, O_2) + \varrho(O_2, O_3)$

Moglibyśmy teraz określać odległość w przestrzeni z tak zdefiniowaną metryką, gdyby nie fakt, że każdy wektor reprezentuje zupełnie inną cechę obrazu. W najprostszym podejściu można tę trudność ominąć definiując macierz wag $D = [d_{ij}]$, która określi wzajemną ważność pomiędzy cechami f_i i f_j . Poszukujemy zatem najmniejszej wartości S , którą obliczamy jako:

$$S_{\min}(i, j) = \frac{\sum_{i,j=1}^N d_{i,j} (f_i - f_j)^2}{\sum_{i,j=1}^N (f_i - f_j)^2} \quad (48)$$

gdzie: f_i i f_j - odpowiednio cechy obrazów O_i i O_j .

Uogólnioną funkcję odległości (podobieństwa) można oprzeć na r -metryce Minkowskiego [43]:

$$d(x, y) = \sqrt[r]{\sum_{i=0}^k (x_i - y_i)^r} \quad \text{i} \quad d_{\infty}(x, y) = \max_i |x_i - y_i| \quad (49)$$

gdzie: $r \geq 1$, a x_i i y_i są to k -elementowe wektory. Oczywiście dla $r = 2$ uzyskujemy odległość euklidesową [29].

Kolejną miarą podobieństwa opartą na wektorach cech jest cosinus wektorów:

$$\text{Sim}(O_i, O_j) = \frac{\sum_{i=1}^N x_i y_i}{\sum_{i=1}^N x_i^2 \sum_{i=1}^N y_i^2} \quad (50)$$

gdzie: x_i i y_i - odpowiednio cechy obrazów O_i i O_j .

7.2 Rozmyte miary podobieństwa

Zapytania w bazie CHABOT można formułować w sposób rozmyty, np. *trochę czerwonego, większość żółtego*, itp. Przy takich zapytaniach w „Module dopasowania” jest zliczana liczba pikseli uważanych jako czerwone lub żółte w stosunku do całkowitej liczby

pikseli w obrazie. Np. aby uznać obraz za w większości żółty 50% pikseli musi być żółtych przy czym 20 kolorów w histogramie jest klasyfikowanych jako kolor żółty.

Porównywanie obrazów kolorowych w podanym powyżej przykładzie bazy CHABOT jest tylko najprostszą implementacją i nie daje dokładniejszej informacji o obrazie. Jeśli np. użytkownik chce znaleźć zachód słońca, to poza czerwonymi zachodami słońca baza znajdzie również łąki z czerwonymi kwiatami. Natomiast jeśli przyjrzeć się porównaniom w większej przestrzeni cech to rezultaty mogą być znacznie bardziej obiecujące.

Zastanówmy się więc nad zapisem formalnym miary podobieństwa opartej na zbiorach rozmytych [48].

Definicja 3

Niech zbiór U ma rozmyte podzbiory A , B i C . Rodzinę funkcji odległości pomiędzy A i B można zdefiniować następująco:

$$d(A, B) = r \sqrt{\sum_{i=0}^k [\mu_A(x_i) - \mu_B(x_i)]^r} \quad \text{ i } \quad d_\infty(A, B) = \sup_i |\mu_A(x_i) - \mu_B(x_i)| \quad (62)$$

gdzie: $r \geq 1$, a $\mu_{(\cdot)}(\cdot)$ są funkcjami przynależności.

Relacja podobieństwa oparta na zbiorach rozmytych $U \times U \rightarrow [0,1]$ ma następujące właściwości:

1. $\text{Sim}(A, A) = 1$
2. $\text{Sim}(A, B) = \text{Sim}(B, A) > 0$
3. $\text{Sim}(A, B) = 0$ jeśli A i B są rozłączne
4. Jeśli $A(u) \leq B(u) \leq C(u)$ lub $A(u) \geq B(u) \geq C(u)$ i $\forall u \in U$ wtedy $\text{Sim}(A, C) \leq \text{Sim}(B, C)$

Przykładową relację podobieństwa można zapisać następująco:

$$\text{Sim}(A, B) = \frac{\max_{u \in U} (A \cap B)}{\max_{u \in U} (A \cup B)} \quad (63)$$

W przypadku obrazów obiecujący jest model Czernskiego [13], w którym podobieństwa pary obrazów wyraża się liniową kombinacją miary wspólnej i rozłącznej cech. Jeśli zatem A i B oznaczają zbiory cech odpowiednio obrazów a i b , to można zaproponować następującą miarę podobieństwa:

$$S(a, b) = \theta f(A \cap B) - \alpha f(A - B) - \beta f(B - A) \quad (64)$$

gdzie: $f(\cdot)$ jest zwykłą licznością zbiorów; $\alpha, \beta, \theta \geq 0$ są wagami.

Można też posłużyć się modelem znormalizowanym do przedziału $[0,1]$:

$$S(a,b) = \frac{f(A \cap B)}{f(A \cap B) + \alpha f(A - B) + \beta f(B - A)} \quad (65)$$

gdzie: $0 \leq \alpha \leq 1$ i $0 \leq \beta \leq 1$.

7.3 Miary oparte na łącznym rozkładzie prawdopodobieństwa

Do poszukiwania podobieństwa można też stosować podejście probabilistyczne zaproponowane przez Jaccarda [27]. Przyjmujemy, że mamy obiekty na obrazie posiadające oddzielnie cechy A , S oraz obie cechy A i S na raz tak jak to pokazuje rys. 76.



Rysunek 76 Obiekty w obrazie mogą mieć cechy A (zielone), cechy S (pomarańczowe) lub obie te cechy na raz (pomarańczowe z zieloną ramką)

W takim przypadku miara podobieństwa będzie liczona na podstawie łącznego rozkładu prawdopodobieństwa wystąpienia cech:

$$Sim(A,S) = \frac{P(A \cap S)}{P(A \cup S)} \quad (66)$$

Można tę metodę zastosować też do obrazów nie poddanych jeszcze obróbce wstępnej, w których nie są wydzielone obiekty i ich cechy *explicite*. Wtedy zbiory A i S będą zbiorami pikseli i będzie można porównywać na przykład podobieństwo kolorów.

8. Instrukcja obsługi użytkownika

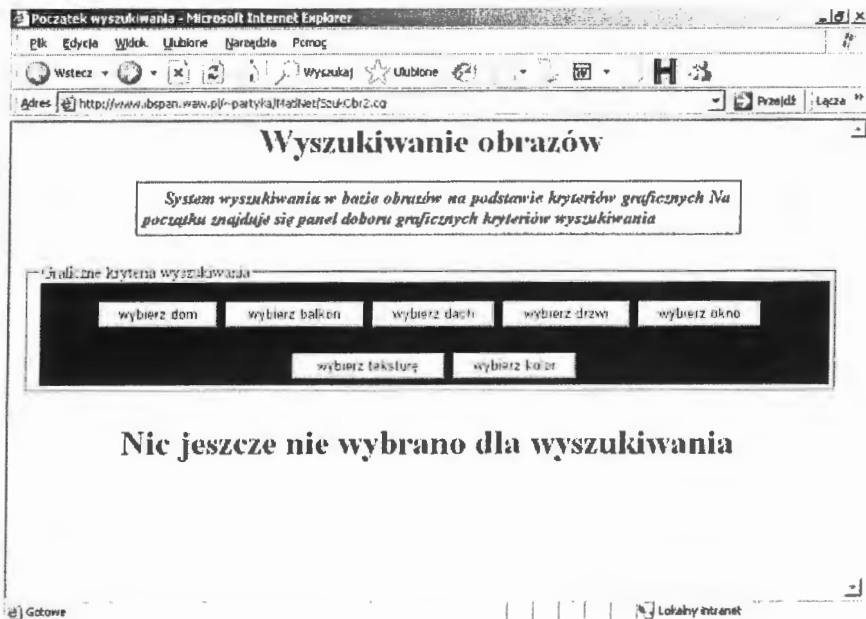
Obrazowa baza danych jest dostępna na instytucyjnej stronie internetowej. Dla rozpoczęcia pracy należy wpisać w polu adresu wyszukiwarki adres:

<http://www.ibspan.waw.pl/~partyka/MatNet/SzukObr2.cgi>

Jeżeli użytkownik chce ominąć korzystanie z apletu Javy przy ustalaniu koloru, powinien wpisać:

<http://www.ibspan.waw.pl/~partyka/MatNet/SzukObr2.cgi/BezJawy>

Powinien wtedy ukazać się następujący ekran:



Dla wybrania określonego elementu graficznego należy kliknąć na odpowiedni przycisk. Jeżeli nie jest to przycisk „wybierz kolor, to pokaże się ekran z obrazami, z których należy wybrać jeden. Wygląd takiego ekranu przedstawiony jest poniżej



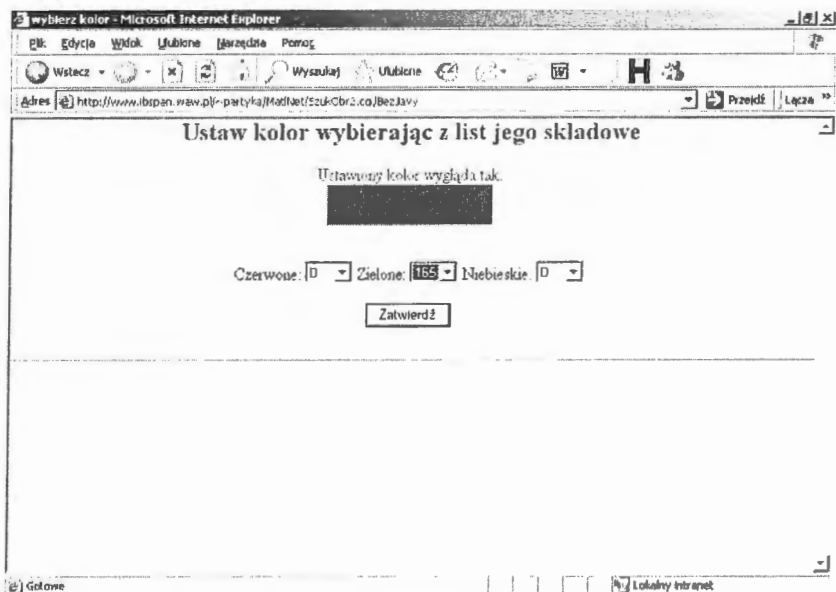
Dla wybrania obrazu należy zaznaczyć odpowiedni przycisk radiowy pod obrazem, a następnie zatwierdzić to przez kliknięcie przycisku „zatwierdź wybór”. Jeżeli użytkownik chce dokładniej obejrzeć dany obraz należy kliknąć na niego dwa razy i wtedy obraz pokaże się w oryginalnej wielkości na oddzielnym ekranie.

Jeżeli przycisk zatwierdzający zostanie kliknięty *bez uprzedniego zaznaczenia obrazu*, to ewentualny poprzedni wybór elementu z tej kategorii zostanie skasowany.

Jeżeli użytkownik wybierze ustawianie koloru, a jako sposób wyboru jest określone wykorzystanie apletu Javy, to pokaże się ekran apletu przedstawiony poniżej:



Jeżeli korzystania z apletu zabroniono, to ekran ustawiania koloru będzie wyglądał tak, jak poniżej



Powyższe kroki użytkownik może powtarzać dowolną ilość razy. Po każdym zatwierdzeniu pokazywane są elementy graficzne dotychczas wybrane. Ekran pokazujący dotychczasowe wybory pokazany jest poniżej



Gdy użytkownik stwierdzi, że zakończył ustalanie kryteriów wyszukiwania, powinien kliknąć przycisk „Wyszukaj w bazie”. Wtedy następuje wywołanie programu w MatLabie, który przyjmuje wybory użytkownika jako parametry wstępne.

9. Matematyczne podstawy przetwarzania obrazu – metody stosowane w tym opracowaniu

9.1 Analiza skupień

Analizy skupień jest to podział zbioru X na skupienia C_1, \dots, C_j . W przypadku zastosowania analizy skupień do segmentacji obrazu każdy punkt obrazu $x_{mn} \in X$ o wymiarach $M \times N$ i L stopniach szarości przyporządkowuje się do jednego ze skupień C_j . Dla formalności należy jeszcze zastrzec rozłączność skupień i określić przynależność wszystkich punktów według następującej definicji:

Definicja 1

Niech X będzie zbiorem danych (np. obrazem rastrowym) i $x_{mn} \in X$. Podziałem zbioru X nazwiemy taką klasę jego podzbiorów (skupień) $P = \{C_1, C_2, \dots, C_l\}$, która spełnia następujące warunki:

1. $\bigwedge_{x_{mn} \in X} \bigvee_{C_j \in P} x_{mn} \in C_j$;
2. $\bigwedge_{x_{mn} \in X} x_{mn} \in C_j \Rightarrow x_{mn} \notin C_i$ gdzie $i \neq j, C_i, C_j \in P$.

Mówimy, że zbiór skupień jest dobrze określony jeśli każde dwa punkty wewnątrz skupienia leżą bliżej siebie niż najbliższe dwa punkty z sąsiadujących ze sobą skupień. W najprostszym przypadku „bliskość” określamy jako odległość euklidesową. Zatem, aby skupienia były dobrze od siebie oddzielone musimy znaleźć środek każdego z nich zwany centroidem, a następnie musimy przyporządkować każdemu punktowi obrazu odpowiednie skupienie. Aby to osiągnąć trzeba przyjąć, że problem analizy skupień jest problemem minimalizacji wartości następującej funkcji:

$$f(P, V) = \sum_{j=1}^c \sum_{m=1}^M \sum_{n=1}^N \|x_{mn} - v_{C_j}\|^2 \quad (67)$$

gdzie: v_{C_j} - są środkami klastrow C_j dla $j = 1, \dots, c$,

$V = \{v_{C_j}\}$ – zbiór środków klastrow.

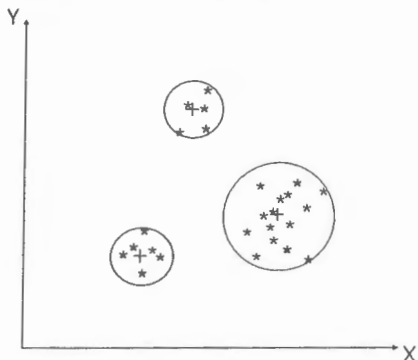
Jako metrykę przyjmujemy metrykę euklidesową. Próbuje się zatem znaleźć takie centroidy v_{C_j} , które minimalizują wartość funkcji (67). Postępowanie jest iteracyjne i odbywa się w dwóch krokach:

1. Obliczamy podział w oparciu o takie centroidy jakie już mamy;

2. Modyfikujemy istniejące skupienia przesuwając ich centroidy. Przy tej modyfikacji używamy metody malejących gradientów do minimalizacji funkcji $f(P, V)$.

Kończymy iteracje, gdy różnica położenia centroidów pomiędzy dwoma kolejnymi krokami iteracji jest mniejsza niż przyjęty próg (np. rozdzielczość obrazu).

Algorytm analizy skupień jest zatem zbieżny do lokalnych minimów funkcji $f(P, V)$. Poglądowo skupienia zostały zobrazowane na rys. 77.



Rysunek 77 Przykłady skupień z zaznaczonymi centroidami „+” w przypadku dwuwymiarowym

9.2 Segmentacja oparty na progowaniu rozmytym

Najprostsza z wymienionych segmentacji oparta na progowaniu rozmytym pozwala na wydzielenie obiektów z tła. Progowanie bazuje na założeniu, że regiony odpowiadające różnym obiektom mogą być rozróżniane przy użyciu funkcji opartej na wartości intensywności pikseli w obrazie [3]. Algorytm segmentacji opartej na progowaniu przy zastosowaniu zbiorów rozmytych wygląda następująco:

Zbiór rozmyty A w przestrzeni U reprezentowany jest przez swoją funkcję przynależności $\mu_A \sim [0,1]$. Dla danego p (może to być nasz przykładowy piksel) $\mu_A(p)$ nazywamy stopniem przynależności p do A . Obraz X o wymiarach $M \times N$ i L stopniach szarości jest traktowany jako tablica rozmytych singletonów, z których każdy ma taki stopień przynależności do obrazu, jaki wynika z jego poziomu wybranej cechy (np. jasności l , gdzie $l = 0, 1, \dots, L-1$). Obraz X może być zatem opisany jako:

$$X = \{\mu_x / x_{mn}; \quad m = 1, \dots, M, n = 1, \dots, N\} \quad (68)$$

gdzie: $0 \leq \mu_x(x_{mn}) \leq 1$ oznacza, że piksel x_{mn} posiada w odpowiednim stopniu pewną cechę (np. jasność, gładkość, bliskość krawędzi, itp).

Przy definiowaniu funkcji przynależności brane są pod uwagę zarówno informacje globalne o danym obrazie jak i lokalne. Tak więc wartość funkcji przynależności może zależeć od takich cech pikseli jak: intensywność, barwa, odległość od krawędzi, położenie w obrazie. Przykładowo, funkcja przynależności może mieć postać:

- S funkcji
- funkcji trapezoidalnej
- trójkąta
- funkcji Gaussa

W oparciu o funkcję przynależności można zdefiniować dla danego obrazu różne miary rozmytości np.:

- indeksy rozmytości
- rozmytą entropię
- zwartość zbioru rozmytego
- stopień pokrycia zbioru rozmytego

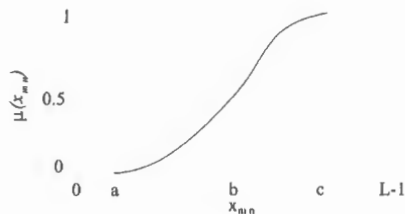
które z kolei są kryterium do określenia progów dla segmentowanego obrazu. Z całej tej gamy możliwości przyjrzymy się najprostszemu i najbardziej typowemu podejściu do segmentacji w oparciu o progowanie rozmyte, który przedstawia poniższy algorytm.

Krok 1

Konstruujemy funkcję przynależności μ_{mn} standardowego typu S taką, że:

$$\mu_x(x_{mn}) = S(x_{mn}; a, b, c) = \begin{cases} 0 & \text{if } x_{mn} \leq a, \\ 2(x_{mn} - a)^2 / (c - a)^2 & \text{if } a \leq x_{mn} \leq b, \\ 1 - 2(x_{mn} - c)^2 / (c - a)^2 & \text{if } b \leq x_{mn} \leq c, \\ 1 & \text{if } x_{mn} \geq c. \end{cases} \quad (69)$$

gdzie: $b = (a+c)/2$ jest punktem przecięcia z $S(b; a, b, c) = 0.5$ dla okna o wymiarze $w = c - a$.
 Funkcja typu S jest pokazana na rysunku 78.



Rysunek 78 Standardowa S funkcja

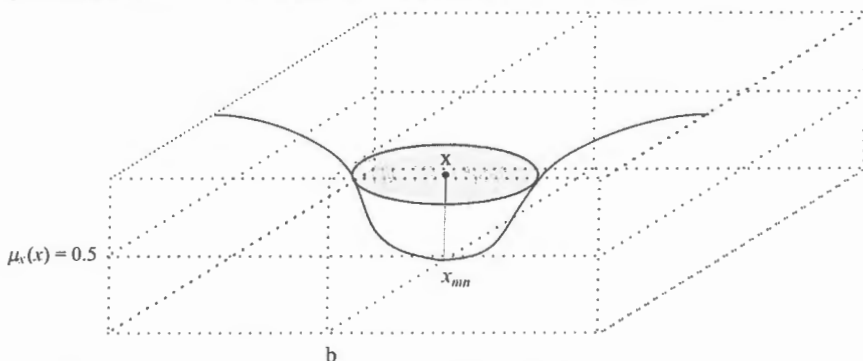
Krok II

Obliczamy: powierzchnię, długość, szerokość i stopień rozmycia zbioru rozmytego według następujących definicji:

- licznosc traktowana jako powierzchnia obrazu $a(\mu_x) = \sum_m \sum_n \mu_x(x_{mn})$;
- dlugosc $l(\mu_x) = \max_m (\sum_n \mu_x(x_{mn}))$;
- szerokosc $b(\mu_x) = \max_n (\sum_m \mu_x(x_{mn}))$;
- stopien pokrycia $IOAC(\mu_x) = \frac{a(\mu_x)}{l(\mu_x) \times b(\mu_x)}$.

Krok III

Zmieniamy b od 0 do $L-1$ i wybieramy takie płaszczyzny, dla których stopień pokrycia $IOAC(\mu_x)$ ma lokalne minima. Wśród lokalnych minimów globalne minimum ma punkt przecięcia x taki, że $\mu_x(x) = 0.5$ i $b = x$. Płaszczyzna $\mu_x(x_{mn})$ odpowiadająca punktowi przecięcia x może być widziana jako rozmyty segment obrazu X (rys. 79).



Rysunek 79 Segmentacja obrazu oparta na funkcji przynależności $\mu_x(x_{mn})$ typu S

9.3 Rozmyta analiza skupień

Natomiast rozmyta analiza skupień (*fuzzy c-means clustering*) [42] wymaga, żeby w przypadku segmentacji obrazu każdy punkt $x_{mn} \in X$ obrazu o wymiarach $M \times N$ i L stopniach szarości należał do skupienia C_j w stopniu określonym przez funkcję przynależności $0 \leq \mu_{C_j}(x_{mn}) \leq 1$, gdzie $j = 1, \dots, c$; $m = 1, \dots, M$ i $n = 1, \dots, N$. Funkcja przynależności powinna być tak skonstruowana, aby spełniała następujące warunki:

$$\bigwedge_{x_{mn} \in X} \sum_j \mu_{C_j}(x_{mn}) = 1 \quad (70)$$

(czyli jest $M \times N$ takich sum) oraz

$$\bigwedge_{x_{mn} \in X} \bigvee_{C_j \in P} \mu_{C_j}(x_{mn}) > 0 \quad (71)$$

W tym wypadku funkcja minimalizacji powinna uwzględniać dodatkowo funkcję przynależności punktów do skupień i współczynnik rozmytości skupień. Otrzymujemy zatem następującą funkcję:

$$f_X(P, U) = \sum_{j=1}^c \sum_{m=1}^M \sum_{n=1}^N [\mu_{C_j}(x_{mn})]^q (x_{mn} - v_{C_j})^2 \quad (72)$$

gdzie:

$U = \{ v_{C_j} \}$ - zbiór centroidów, gdzie $j = 1, \dots, c$,

P - zbiór rozmytych podziałów punktów x_{mn} pomiędzy skupienia C_j ,

$q > 1$ - współczynnik rozmytości skupień.

Jako metrykę przyjmujemy metrykę euklidesową. Funkcja (72) przyjmuje wartości niezerowe. Natomiast gdyby nie było warunku (70) minimalizacja sprowadziłaby się tylko do przyjęcia $\mu_{C_j} = 0$. Podobnie jak w przypadku klasycznego algorytmu c-środków szuka się takiego podziału $\{C_j\}$, żeby zminimalizować funkcję $f_X(P, U)$, ale algorytm rozmytych c-środków musi jeszcze brać pod uwagę funkcje przynależności μ_{C_j} , dla których funkcja $f_X(P, U)$ jest minimalna. Żeby osiągnąć oba te wymagania musi zostać narzucony warunek na lokalne minima $f_X(P, U)$. Przede wszystkim pochodne cząstkowe $f_X(P, U)$ muszą być jednocześnie równe 0.

Zatem:

$$\frac{\partial f_x(P,U)}{\partial \mu_{C_j}} = q[\mu_{C_j}(x_{mn})]^{q-1} \sum_{m=1}^M \sum_{n=1}^N (x_{mn} - v_{C_j})^2 = 0 \quad \text{dla każdego } j \quad (73)$$

$$\frac{\partial f_x(P,U)}{\partial v_{C_j}} = \sum_{j=1}^c \sum_{m=1}^M \sum_{n=1}^N [\mu_{C_j}(x_{mn})]^{q-1} 2(x_{mn} - v_{C_j}) = 0 \quad (74)$$

Stąd obliczamy v_{C_j} dla danego j :

$$v_{C_j} = \frac{\sum_{m=1}^M \sum_{n=1}^N \mu_{C_j}(x_{mn})^q x_{mn}}{\sum_{m=1}^M \sum_{n=1}^N \mu_{C_j}(x_{mn})^q} \quad (75)$$

Natomiast przyrównując do siebie dwie pochodne (73) dla $j \neq i$ otrzymujemy:

$$q[\mu_{C_j}(x_{mn})]^{q-1} \sum_{m=1}^M \sum_{n=1}^N (x_{mn} - v_{C_j})^2 = q[\mu_{C_i}(x_{mn})]^{q-1} \sum_{m=1}^M \sum_{n=1}^N (x_{mn} - v_{C_i})^2 \quad (76)$$

$$\mu_{C_j}(x_{mn}) = \mu_{C_i}(x_{mn}) \left[\frac{\sum_{m=1}^M \sum_{n=1}^N (x_{mn} - v_{C_j})^2}{\sum_{m=1}^M \sum_{n=1}^N (x_{mn} - v_{C_i})^2} \right]^{\frac{1}{q-1}} \quad (77)$$

Z własności (70) otrzymujemy:

$$\mu_{C_j}(x_{mn}) = \frac{1}{\sum_{j=1}^c \left[\frac{\sum_{m=1}^M \sum_{n=1}^N (x_{mn} - v_{C_j})^2}{\sum_{m=1}^M \sum_{n=1}^N (x_{mn} - v_{C_j})^2} \right]^{\frac{1}{q-1}}} \quad (78)$$

Na podstawie równań (75) i (78) można napisać algorytm analizy skupień z iteracyjnym liczeniem funkcji przynależności. W 1990 Bezdek udowodnił jego zbieżność dla $q > 1$ [4]:

Krok 1: Wybieramy losowo c skupienia o środkach $V = \{v_{C_1}, v_{C_2}, \dots, v_{C_c}\}$.

Krok 2: Przyjmujemy V jako stare V^{st} .

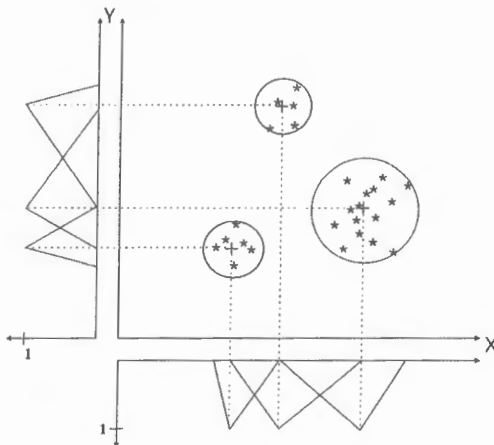
Krok 3: Liczymy funkcję przynależności według wzoru (78).

Krok 4: Uaktualniamy wstępne centroidy v_{C_j} w V według wzoru (75).

Krok 5: Obliczamy odległość starego centroidu od nowego $E = \sum_{i=1}^c (v_{C_i}^{st} - v_{C_i})^2$.

Krok 6: Jeśli $E > \varepsilon$ to idź do kroku 2.

Krok 7: Jeśli $E \leq \varepsilon$ to wyświetl rezultat końcowy.



Rysunek 80 Przykładowe skupienia z zaznaczonymi centroidami i zaznaczonymi funkcjami przynależności dla punktów najbliższych tym centroidom w danych skupieniach.

W powyższym algorytmie ε jest wartością progową np. rozdzielczością obrazka. Należy zauważyć, że wyniki algorytmu zależą nie tylko od wyboru parametrów q i c , ale również od wyboru początkowych centroidów.

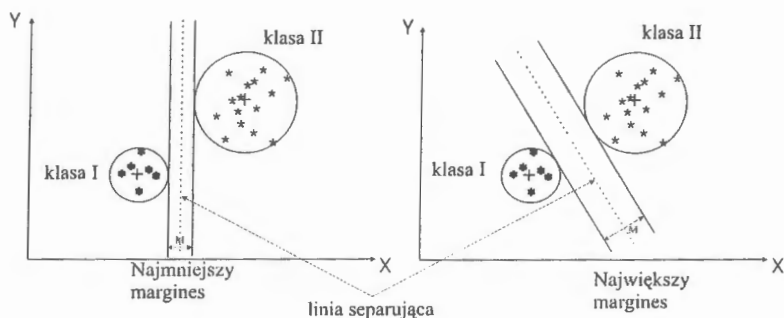
Na rysunku 80 pokazane są funkcje przynależności dla przykładowych punktów leżących najbliższymi centroidami w przykładowych segmentach.

Istnieją oczywiście pewne wady tego algorytmu, do których należy zaliczyć:

1. Zmienność wyników końcowych w zależności od wyboru parametrów początkowych;
2. Nieznana najlepsza liczba klastrów c ;
3. Brak danych na ile ten algorytm pokrywa naturalne struktury obrazu.

9.4 Metoda wektorów wspierających SVM

metodą wektorów wspierających (SVMs wprowadzonych po raz pierwszy przez Vapnika w 1995r. [11]) w celu znalezienia optymalnej hiperpłaszczyzny separującej (OSH), czyli takiej hiperpłaszczyzny, która znajduje maksymalny przedział/ margines M pomiędzy szukanymi klasami [58].



Rysunek 81 Linie separujące dla dwóch klas (w przypadku wielowymiarowym hiperpłaszczyzny separującej). Po lewo linia z małym marginesem M i po prawo linia z dużym marginesem M , który docelowo chcemy uzyskać.

Jeśli dla łatwiejszego zrozumienia uprościmy sobie problem do dwóch klas $x \in \mathbf{R}^2$ to możemy go przedstawić graficznie [66] jak na rysunku 81. Widzimy na nim płaszczyznę separującą opisaną równaniem $w_1 x_1 + w_2 x_2 + b = 0$, która jest też granicą decyzyjną. Interesuje nas zatem, jak dobrać parametry w_1 , w_2 i b , aby uzyskać największy margines M pomiędzy klasami, czyli jak znaleźć optymalną funkcję decyzyjną $d(x, w, b)$.

$$d(x, w, b) = w^T x + b = \sum_{i=1}^n w_i x_i + b \quad (79)$$

gdzie: x , $w \in \mathbf{R}^n$ i b jest skalarą zwaną błędem.

Okazuje się, że uzyskujemy tę funkcję dobierając odpowiednio przykłady uczące $w = [w_1 \ w_2 \ \dots \ w_n]^T$ w kolejnych etapach uczenia. Linia przerywana na rys. 81 jest opisana jako $d(x, w, b) = 0$. W procesie uczenia korzystamy z przykładowych punktów x_p i z funkcji wskaźnikowej (indicator function) danej wzorem:

$$i_F = o = \text{sign}(d(x, w, b)) \quad (80)$$

gdzie: o jest standardową notacją dla „wyjścia”. Możemy zatem zapisać regułę decyzyjną jako:

jeśli $d(x, w, b) > 0$ to przykładowe punkty x_p należą do klasy I, czyli ($o = y_1 = +1$)

i

jeśli $d(x, w, b) < 0$ to przykładowe punkty x_p należą do klasy II, czyli ($o = y_2 = -1$)

Należy zauważyć, że płaszczyzna decyzyjna $d(x, w, b)$ jest również płaszczyzną kanoniczną, a zatem hiperpłaszczyzna przyjmuje formę kanoniczną ze względu na dane uczące $x \in \mathbf{X}$, jeśli

$$\min_{x \in X} |\mathbf{w}^T \mathbf{x} + b| = 1 \quad (81)$$

i jeśli jednocześnie ta hiperpłaszczyzna kanoniczna ma maksymalny margines M i leży w jego środku. Zatem z zależności geometrycznych (odległości między elementami dwóch klas) widać, że:

$$M = \frac{2}{\|\mathbf{w}\|} \quad (82)$$

gdzie: $\|\mathbf{w}\| = \sqrt{\mathbf{w}^T \mathbf{w}} = \sqrt{\sum_i w_i^2}$. Jeśli $\|\mathbf{w}\|$ będzie minimalne, to M będzie maksymalne.

Żeby to uzyskać trzeba przyjąć następujące ograniczenie:

$$y_i[\mathbf{w}^T \mathbf{x} + b] \geq 1 \quad \text{dla } i = 1, \dots, l \quad (83)$$

gdzie: l oznacza liczbę punktów uczących/ trenujących, a \mathbf{w} spełniające to ograniczenie nazywamy wektorem wspomagającym.

Znalezienie minimum $\|\mathbf{w}\|^2$ jest klasycznym problemem optymalizacji przy nierównych ograniczeniach i jest rozwiązywany przy użyciu funkcji Lagrange'a

$$L(\mathbf{w}, b, \boldsymbol{\alpha}) = \frac{1}{2} \mathbf{w}^T \mathbf{w} - \sum_{i=1}^l \alpha_i \{y_i[\mathbf{w}^T \mathbf{x}_i + b] - 1\} \quad (84)$$

gdzie: α_i są to mnożniki Lagrange'a, które w tym przypadku są wektorami nośnymi dla hiperpłaszczyzny separującej. Poszukujemy punktu siodłowego $(\mathbf{w}_0, b_0, \boldsymbol{\alpha}_0)$, bo lagrangian L musi być minimalizowany ze względu na \mathbf{w} i b_0 i niezerowe α_i (czyli poszukujemy takich $\alpha_i \geq 0$). Tak więc pochodne po głównych zmiennych są równe zero, a zatem otrzymujemy:

$$\frac{\partial L}{\partial \mathbf{w}_0} = 0 \quad \text{stad } \mathbf{w}_0 = \sum_{i=1}^l \alpha_i y_i \mathbf{x}_i \quad (85)$$

$$\frac{\partial L}{\partial b_0} = 0 \quad \text{stad } \sum_{i=1}^l \alpha_i y_i = 0 \quad (86)$$

Dodatkowo musi być spełniony warunek Karush-Kuhn-Tuckera (KKT)

$$\alpha_i \{y_i[\mathbf{w}^T \mathbf{x}_i + b] - 1\} = 0, \quad i = 1, \dots, l \quad (87)$$

Podstawiając równania (85) i (86) do równania (84) zmieniamy lagrangian $L(\mathbf{w}, b, \boldsymbol{\alpha})$ jednej zmiennej na lagrangian $L_d(\boldsymbol{\alpha})$ dualny i otrzymujemy:

W celu znalezienia optymalnej hiperpłaszczyzny dualny lagrangian maksymalizujemy ze względu na niezerowe mnożniki $\alpha_i \geq 0$ przy założeniu ograniczenia $\sum_{i=1}^l \alpha_i y_i = 0$.

$$L_d(\boldsymbol{\alpha}) = \sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i,j=1}^l \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j \quad (88)$$

Obliczenie hiperpłaszczyzn umożliwia łatwiejszy podział w naszym przypadku punktów obrazu na odpowiednie segmenty, szczególnie jeśli segmentacja dotyczy obrazów kolorowych, gdzie kolory są ze sobą skorelowane np. w modelu RGB.

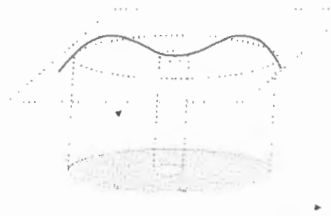
9.5 Metoda aktywnych konturów

Kontur może być definiowany jako krzywa parametryczna, wielokąt, bądź krzywa sklejana (B-spline), jednakże występuje wtedy problemy z opisaniem obiektów niejednorodnych topologicznie (patrz rysunek 82). Dlatego też powstał model geodezyjny konturów aktywnych [8], w których nie występuje ten problem. Formalna definicję konturów możemy przedstawić jako:

Definicja 2

Niech ϱ oznacza metrykę (np. metrykę euklidesową) w \mathbf{R}^2 i niech $K(\mathbf{x}_0, \varepsilon) = \{\mathbf{x} \in \mathbf{R}^2: \varrho(\mathbf{x}_0, \mathbf{x}) < \varepsilon\}$ oznacza kulę o środku $\mathbf{x}_0 \in \mathbf{R}^2$ i promieniu $\varepsilon > 0$. Zbiór $c \subset \mathbf{R}^2$ nazywamy konturem wtedy i tylko wtedy, gdy istnieje taka funkcja $f: \mathbf{R}^2 \rightarrow \mathbf{R}$, że:

$$c = \{x \in \mathbf{R}^2: \bigvee_{\varepsilon > 0} \bigwedge_{x_1, x_2 \in K(x, \varepsilon)} f(x_1) \geq 0 \wedge f(x_2) < 0\}$$



Rysunek 82 Przykład konturów dla obiektu niejednorodnego topologicznie

Parametryczny model aktywnego konturu zwany też algorytmem węża, odchodzi od tradycyjnego modelu wyznaczania cech w obrazie opartego jedynie na intensywności pikseli. Algorytm węża zmienia problem wyznaczania granic obiektu w problem minimalizacji energii. Tradycyjny „wąż” jest krzywą parametryczną $C(p) = \begin{bmatrix} x(p) \\ y(p) \end{bmatrix}$ dla $p \in [0,1]$, która porusza się w dziedzinie przestrzennej Ω obrazu $I(x,y)$. Zagadnienie polega na minimalizacji energii opisanej jako suma energii wewnętrznej E_{int} i zewnętrznej E_{ext} . Kształt modelu obiektu jest kontrolowany przez energię wewnętrzną, podczas gdy energia

$$J(C) = E_{int}(C) + E_{ext}(C) \quad (89)$$

zewnętrzna jest potrzebna do opisu zewnątrz obiektu. Energia zewnętrzna jest obliczana przez całkowanie funkcji energii wzdłuż konturu czyli:

$$E_{ext} = \int_0^1 P(C(p)) dp = -\nabla P(C(p)) \quad (90)$$

gdzie $P(x,y)$ jest liczona jako splot obrazu $I(x,y)$ z dwuwymiarową funkcją Gaussa $G_\sigma(x,y)$ o wariancji σ^2 , czyli:

$$P(x,y) = -|\nabla G_\sigma(x,y) * I(x,y)|^2 \quad (91)$$

Tak liczona funkcja energii potencjalnej zwana jest mapą krawędzi obrazu.

Energia wewnętrzna zapewnia nam zatem gładkość i ciągłość konturu, czyli istnieje pierwsza i druga pochodna z C po p . Pierwsza pochodna $C'(p)$ jest odpowiedzialna za rozciąganie i elastyczne zachowanie konturu, natomiast druga pochodna $C''(p)$ odpowiada za zginanie się konturu i jego odpowiednią sztywność. Wagi α i β są odpowiednio parametrami używanymi do sterowanie elastycznością i sztywnością. Zatem energia wewnętrzna jest opisana jako:

$$E_{int} = \frac{1}{2} \int_0^1 \alpha |C'(p)|^2 + \beta |C''(p)|^2 dp \quad (92)$$

Aby zatem znaleźć krzywą $C(p)$, która minimalizuje funkcję $J(C(p))$ wykorzystujemy równanie Eulera-Lagrange'a:

$$\alpha C''(p) - \beta C''''(p) - \nabla P(C(p)) = 0 \quad (93)$$

W tym miejscu należy zauważyć (co zrobili Xu i Prince [68]), że (93) jest to równanie sił, a mapę krawędzi można traktować jako pole siły potencjalnej. Taki przykładowy rozkład sił na aktywnym konturze prezentuje rysunek 83. Zatem równanie sił możemy zapisać jako:

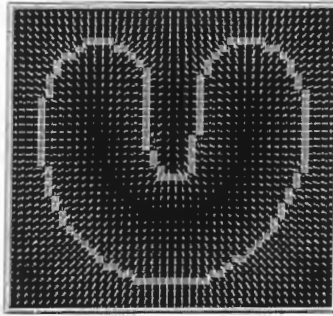
$$F_{int} + F_{ext}^{(p)} = 0 \quad (94)$$

gdzie: $F_{int} = \alpha C''(p) - \beta C''''(p)$ i $F_{ext}^{(p)} = -\nabla E_{ext}$.

Aby rozwiązać (95) trzeba potraktować „węże” jako krzywą dynamiczną, czyli zależną od czasu $C(p,t)$. Wtedy pochodna z C po czasie jest równa lewej stronie równania (95):

$$\frac{\partial C}{\partial t} = \alpha C''(p) - \beta C''''(p) - \nabla P(C(p)) \quad (95)$$

Kiedy rozwiązanie $C(p,t)$ jest stabilne, człon $\frac{\partial C(p,t)}{\partial t}$ znika i otrzymujemy rozwiązanie równania (95).



Rysunek 83 Przykład rozkładu gradientów sił na aktywnym konturze

9.6. Transformacja falkowa

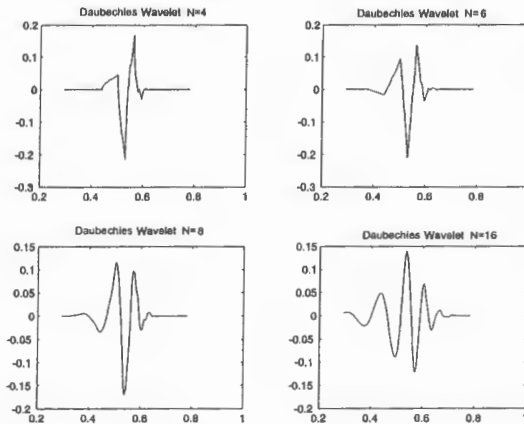
9.6.1. Jednowymiarowa transformacja falkowa

Jednowymiarowa funkcja falkowa określana wzorem [14]:

$$\psi_{su}(x) = \frac{1}{\sqrt{s}} \psi\left(\frac{x-u}{s}\right) \quad (96)$$

jest to funkcja zmiennej x o dwóch parametrach: $s > 0$ – skali i $u \in R$ – przesunięciu.

Przykład funkcji falkowej pokazano na rys. 84.



Rysunek 84 Przykład falek Daubechies

Jednowymiarowa transformata falkowa $[W_{\psi}f](su)$ jest rzutem ortogonalnym sygnału f z przestrzeni $L^2(R)$ na rodzinę funkcji $\{\psi_{su}, s > 0, u \in R\}$ generowanych przez przesunięcie i skalowanie funkcji ψ .

$$[W_{\psi} f](s, u) = \langle \psi_{su}, f \rangle = \frac{1}{\sqrt{s}} \int_{-\infty}^{+\infty} \psi \left(\frac{x-u}{s} \right) f(x) dx \quad (97)$$

„Nadmiarowość” ciągłej transformacji falkowej może być usunięta przez dwójkową (diadyczną) dyskretyzację współczynnika skali $s = 2^j$ i przesunięcia $u = k$, gdzie $j, k \in \mathbb{Z}$.

$$\psi_{j,k}(x) = 2^{-\frac{j}{2}} \psi(2^{-j}x - k) \quad (98)$$

Transformacja falkowa w układzie falek diadycznych jest następująca:

$$[W_{\psi} f](k, j) = 2^{-\frac{j}{2}} \int_{-\infty}^{+\infty} \psi(2^{-j}x - k) f(x) dx \quad (99)$$

Jeśli teraz przyjmiemy dyskretny argument x to otrzymamy dokładnie taką transformację falkową jaką stasuje się do sygnałów dyskretnych.

$$d_j(k) = 2^{-\frac{j}{2}} \sum_{-\infty}^{+\infty} f(x) \psi_{j,k}(x) \quad (100)$$

9.6.2. Dwuwymiarowa transformacja falkowa

W przypadku obrazu stosuje się dwuwymiarową (2D) transformację falkową [38], która sprowadza się do rzutowania obrazu na falki ustalonego ortogonalnego układu falek. Dwuwymiarowy separowalny układ falek ortogonalnych utworzony konwencjonalnie na podstawie układu jednowymiarowego ma postać: $\{\psi_{k,l}(x) \psi_{l,m}(y)\}_{(j,k,l,m) \in \mathbb{Z}^4}$, gdzie $\psi_{k,l}(x)$ – falka „w kierunku x ”, a $\psi_{l,m}(y)$ – falka „w kierunku y ”, a j to poziom analizy wielorozdzielczej.

Wykorzystując założenia analizy wielorozdzielczej [14] można pokazać, że zaproponowany układ jest nadmiarowy i można go zredukować do:

$$\{\psi_{j,k,m}^1(x, y), \psi_{j,k,m}^2(x, y), \psi_{j,k,m}^3(x, y)\}_{(j,k,m) \in \mathbb{Z}^3} \quad (101)$$

gdzie:

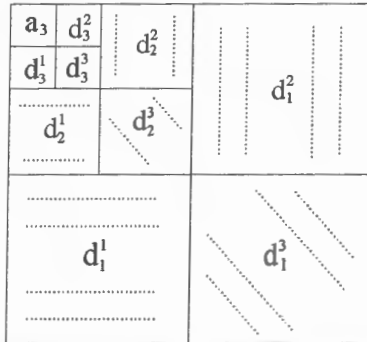
$$\psi^1(x, y) = \phi(x) \psi(y)$$

$$\psi^2(x, y) = \psi(x) \phi(y)$$

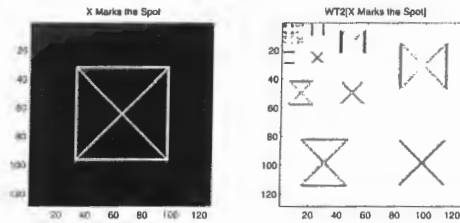
$$\psi^3(x, y) = \psi(x) \psi(y)$$

Podzbiory falek $\psi^1(x, y)$, $\psi^2(x, y)$, $\psi^3(x, y)$ nazwane są odpowiednio dwuwymiarowymi falkami horyzontalnymi, wertykalnymi i diagonalnymi. Podobnie rzuty obrazu na te falki nazywane są horyzontalnymi d_j^1 , wertykalnymi d_j^2 i diagonalnymi d_j^3 detalami obrazu. Na rysunku 23 pokazane są układy tych detali w konwencjonalnym układzie przyjmowanym do wizualizacji 2D transformacji falkowej, natomiast na rys. 24 pokazany jest specjalnie

dobrany obraz pokazujący kierunkowość detali i jego 2D transformacja falkowa ilustrująca układ współczynników z rys. 85.



Rysunek 85 Konwencjonalny układ detali horyzontalnych d_j^1 , wertykalnych d_j^2 i diagonalnych d_j^3 przy wizualizacji dwuwymiarowej transformaty falkowej (a_3 oznacza aproksymację obrazu)



Rysunek 86 Przykładowy obraz (po lewej) i jego transformacja falkowa w układzie współczynników jak na rys. 85 (po prawej)

10 Dokumentacja techniczna programu

10.1 Potrzebne pliki i ich lokalizacja

Interfejs oprogramowania jest umieszczony na serwerze Instytutu. W jego skład wchodzi skrypty CGI napisane w języku Perl, skrypty pomocnicze (również w języku Perl), pliki definiujące arkusze stylów, aplet Javy, a także pliki zawierające obrazy, które wchodzi do kryteriów wyszukiwania.

Wszystkie pliki oprócz plików z obrazami muszą być zawarte w jednym katalogu. Aktualnie jest to katalog: /home3/partyka/public.htm/MatlNet. Natomiast MatLab wersja 7.1 umieszczony jest serwerze ibs5.

W skład oprogramowania wchodzi następujące pliki:

SzukObr2.cgi
Przetwarzaj.cgi
SzukObr.css
Globalne.pl
PokazUrobku.pl
DialogLayout.class
UstawDialog.class
UstawKolor.class

oraz pliki w Matlabie:

domki.m – przeprowadzający analizę skupień obrazu podanego jako parametr
zapis_obrazu.m – zapisuje obraz po analizie w formacie JPEG

Wszystkie pliki zawierające obrazy wchodzące do kryteriów wyszukiwania muszą być umieszczone w podkatalogach określonego katalogu. Nazwa tego katalogu jest podana w pliku Globalne.pl. Aktualnie jest to katalog: /home3/partyka/public.htm. Zmiana tego katalogu wymagałaby zmiany pliku Globalne.pl

Istnieje 6 podkatalogów o nazwach:

BalkonyImg	dla obrazów <i>balkonów</i>
DachyImg	dla obrazów <i>dachów</i>
DrzwiImg	dla obrazów <i>drzwi</i>
OknaImg	dla obrazów <i>okien</i>

TeksturyImg	dla obrazów <i>tekstur</i>
DomyImg	dla obrazów <i>domów</i>

Obrazy w tych podkatalogach powinny być zapisane w formacie GIF lub JPEG.

10.2 Sposób wybierania koloru

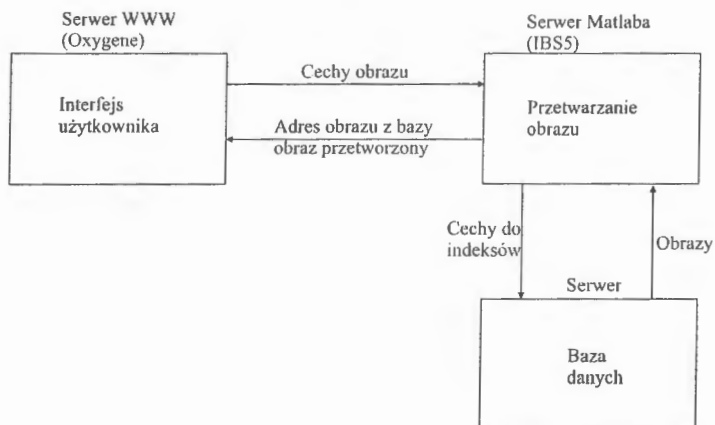
Domyślnym zadawaniem koloru steruje aplet Javy, zawarty w plikach z rozszerzeniem *.class*. Użytkownik ustawia trzema suwakami trzy składowe koloru (czerwoną, zieloną i niebieską) przy czym natychmiast widzi skutki swoich decyzji. Wymaga to jednak, żeby na komputerze użytkownika mogły być realizowane aplety Javy w wersji co najmniej 1.5.0. Ten warunek nie jest zawsze spełniony.

Dla umożliwienia maksymalnie łatwego wyboru koloru dla tych użytkowników, których komputery nie współpracują z apletami Javy, została opracowana metoda alternatywna. W tej metodzie użytkownik ustawia składowe koloru w trzech listach selekcyjnych.

Wyboru metody alternatywnej może użytkownik dokonać przy rozpoczęciu pracy. Alternatywna metoda zostanie też zaproponowana przez oprogramowanie po wykryciu niemożliwości odwołania się do apletów Javy.

10.3 Konfiguracja oprogramowania

Pliki dotyczące strony WWW wykonywane są przez serwer *Oxygene* i są dostępne dla zewnętrznego użytkownika na stronie internetowej Instytutu. Serwer ten komunikuje się z Matlabem przesyłając jako informacje wejściowe parametry (cechy obiektu) wybrane przez użytkownika używającego interfejsu poprzez skrypt *wejściowy* i przyjmuje od Matlaba plik wyjściowy np. *matlab.wy*, zawierający adres do obrazu wybranego przez Matlaba z bazy danych obrazów, bądź obraz przetworzony przez Matlaba. Sama baza danych obrazów może znajdować się na dowolnym serwerze widzianym przez Matlaba. Schemat komunikacji pomiędzy serwerami jest pokazany na rysunku 87.



Rysunek 87 Ideowy schemat konfiguracji pomiędzy serwerami obsługującymi obrazową bazę danych

11. Podsumowanie

Przedstawione opracowanie w szkieletowy sposób przedstawia zagadnienia związane z budową obrazowej bazy danych. Wstępna obróbka obrazu, a zwłaszcza segmentacja zostały rozbudowane ze względu na ich znaczenie dla dalszego funkcjonowania całej bazy. Spośród wielu istniejących metod zostały wybrane te, które wydają się być najbardziej obiecujące przy tworzeniu bazy danych nieruchomości, a więc odnoszą się do pojedynczych obrazów (nie wspomniano metod stosowanych przy bazach zawierających filmy wideo).

Do wydzielenia obiektów z tła została zaprezentowana metoda pozwalająca na analizę obiektów topologicznie niejednorodnych. Dalsza część opracowania obejmuje zagadnienia bazodanowe takie, jak indeksowanie danych i przeszukiwanie danych, również w odniesieniu do bazy obrazowej.

Do opracowania pozostaje jeszcze rozwinięcie interfejsu graficznego użytkownika dające możliwość wyboru nie tylko elementu obrazu, ale też definiowania „ręcznego” parametrów takich jak: kształt, położenia elementu w obrazie, itp.

W miarę jak baza będzie się powiększać będzie też wybierana najbardziej efektywna metoda przeszukiwania danych i ich porównywania ze sobą w celu zapewnienia sprawnej obsługi użytkownika.

Dopiero po wykonaniu całej bazy będzie można pokusić się o wprowadzenie modułu interpretującego zawartość obrazu i „nazywającego” poszczególne obiekty. Taki moduł ekspercki oparty na algorytmach uczenia maszynowego na poszerzenie możliwości eksploatacji danych zawartych w bazie.

Bibliografia

- [1] Alexandrov A. D., Ma W. Y., i in.: *Adaptive Filtering and Indexing for Image Databases*, SPIE Vol. 2420, 1995, str. 12-23
- [2] Balmelli L., Mojsilović A.: *Wavelet domain features for texture/pattern description, classification and replicability analysis*. W: A. Petrosian i F. Meyer, red. ,*Wavelets in Signal and Image Analysis*. Kluwer Academic Publishers, Londyn, rozdz. 7, 2001. str. 193-212.
- [3] Basir O., Zhu H., Karray F.: *Fuzzy Based Image Segmentation*, w *Fuzzy filters for Image Processing*, Springer, Berlin, 2003, str. 101-105
- [4] Bezdek J. C., Kim T., Hathaway R. J.: Reply to *Comments on: optimality test for fixed points by Kim et al.*. *Pattern Recognition*, Vol 23 (11), 1990, str. 1309-1310
- [5] Bracewell R.: *The Fourier Transform and its Application*, McGraw-Hill, NY, 1978
- [6] Canny J.: *A Computational Approach to Edge Detection*, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-8, No. 6, 1986, str. 679-698.
- [7] Carson Ch., i in.: *Blobworld: Image Segmentation Using Expectation-Maximization and Its Application to Image Querying*, *IEEE PAMI* Vol. 24, Nr 8, sierpień 2002, str. 1026-1038
- [8] Caselles V., Kimmel R., Sapiro G.: *Geodesic Active Contours*, *International Journal of Computer Vision* Vol. 22, Nr.1, 1997, str. 61-79
- [9] Castelli V., Bergman I. D., Kontoyiannis I., et al.: *Progressive search and retrieval in large image archives*, *IBM Journal Research Development*, Vol. 42, No. 2, 1998, str. 253-268.
- [10] Castleman Kenneth R.: *Digital Image Processing*, Prentice Hall, 1996
- [11] Cortes C., Vapnik V.: *Support vector networks*. *Machine Learning*, Nr. 20, 1995, str. 273-297.
- [12] Cox B. L., Wang J. S.: *Fractal Surfaces: Measurement and Applications in Earth Sciences*. *Fractals*, 1993, str. 1-87.
- [13] Czerski A. *Features of similarity*. *Psychological Review*, Vol 84, czerwiec 1977, str. 327-352
- [14] Daubechies I.: *Ten lectures on wavelets*, Society for Industrial and Applied Math., Filadelfia, 1992

- [15] Delibassis K., Undrill P. E., Cameron G. G., *Designing Texture Filters with Genetic Algorithms : an application to Medical Images*, Signal Processing, Vol. 57, Nr. 1, 1997, str.19-33.
- [16] Dudek G, Czoczoz J., K.: *Shape representation and recognition from multiscale curvature*, Computer Vision and Image Understanding, Nr. 68 (2), 1997, str. 170-189.
- [17] Flickner M., Sawhney H., i inn.: *Query by Image and Video Content: The QBIC System*, IEEE Computer, Vol. 28, Nr. 9, sierpień 1995, str. 23-32
- [18] Grosky W., Mehrotra R.: *Special Issue on Image Database Management*, Computer, Vol. 22 Nr. 12, grudzień 1989
- [19] Gudivada Venkar N.: *Content-Base Image Retrieval Systems*, IEEE Computer, Vol. 28, Nr. 9, sierpień 1995, str. 18-22.
- [20] Guttman A., *R-trees: a Dynamic Index Structure for Spatial Searching*, Proceedings ACM SIGMOD Conference, Boston, 1984, str. 47-57
- [21] Haar A.: *Zur theorie der orthogonalen funktionensysteme*. Math. Annal., Nr. 69, 1910, str. 331-371
- [22] Hafner J., i in.: *Efficient Colour Histogram Indexing for Quadratic Form Distance Functions*, IEEE PAMI Vol. 17, Nr. 7, lipiec 1995, str. 729-736
- [23] Haralick R., M.: *Statistical and Structure Approaches to Texture*, Proc. of IEEE, 67, 1979, str. 786-804.
- [24] Hong D. Z., Wu J. K., Singh S. S.: *Refining image retrieval based on context-driven method*, SPIE Proceedings on Storage and Retrieval for Image and Video Database VII, Vol. 3656, 1999, str. 581-593.
- [25] Hsu W., Chua T. S., Pung H. K.: *Approximation Content-based Object-Level Image Retrieval*, Multimedia Tools and Applications, Vol. 12, Springer Netherlands, 2000, str. 59-79.
- [26] Jain R.: *NSF Workshop on Visual Information Management Systems*, SIGmod Record, Vol. 22, Nr. 3, grudzień 1993, str 57-75
- [27] Jaccard P.: *Nouvelles recherches sur la distribution florale*, Bulletin de la Societe Vaudoise des Sciences Naturelles, Vol. 44, str.223-270, 1908.
- [28] Jaimes A., Chang S. F.: *Model-based classification of visual information for content-based retrieval*, SPIE Proceedings on Storage and Retrieval for Image and Video Database VII, Vol. 3656, 1999, str. 402-414.
- [29] Jain R., Murthy J., Chen P.: *Similarity Measures for Image Databases*, SPIE, Vol. 2420, str. 58-62

- [30] Jaworska T.: *Przygotowanie wstępne obrazu do obrazowej bazy danych*, W: Inżynieria Wiedzy i Systemy Ekspertowe, red. A. Grzech, Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław 2006, str. 125-134.
- [31] Jaworska T.: *Wyznaczanie faktur we wstępnym przygotowaniu obrazów dla celów obrazowej bazy danych*, W: Badania operacyjne i systemowe 2006, Metody i techniki, red. J. Kacprzyk, R. Budziński, AOW EXIT, Warszawa 2006, str. 397-408.
- [32] Jaworska T.: *Ekstrakcja obiektów z obrazów kolorowych jako pierwszy etap w tworzeniu obrazowej bazy danych*, W: Materiały z V Sympozjum Naukowego Technik Przetwarzania Obrazu, Serock 2006 [w druku].
- [33] Julesz B.: *Inability of Humans to discriminate Between Visual Texture. That Agree in Second-order Statistics – Revisited.*, Perception Nr. 2, 1973 str. 391-405
- [34] Kauppinen H., Seppanen T., Pietikainen M.: *An experimental comparison of autoregressive and Fourier-based descriptors in 2D shape classification*, IEEE Trans PAMI, Vol. 17, Nr. 2, 1995, str. 201-207.
- [35] Ledley R. S., Buas M., Golab T. J.: *Fundamentals of True Color Image Processing*, Proc. 10th Int. Conf. Pattern Recognition IEEE, Los Alamos, Nr. 1, 1990, str. 791-795
- [36] Lewis P., Fauzi M.: *Automatic texture segmentation for content-based image retrieval application*, Pattern Analysis and Applications, 2006, Springer-Verlag, London, (w druku).
- [37] Liu Y.: *θ -index of Imagebase*, SPIE, Vol. 2420, 1995, str. 68-81
- [38] Mallat S.: *A wavelet tour of signal processing*, Academic Press, Londyn, 1998
- [39] Manolopoulos Y. I in.: *R-trees Have Grown Everywhere*, ACM Computer Surveys, 2003, str. 1-62
- [40] Matheron G., *Random Sets and Integral Geometry*, Willey, NY, 1975
- [41] Mehrotra R., Gary J. E.: *Similar-shape retrieval In shape data management*, IEEE Computer, Vol.28, Nr9, 1995, str. 57-65.
- [42] Melin P., Castillo O.: *Hybrid Intelligent Systems for pattern Recognition Using Soft Computing*, Springer, Berlin, 2005, str. 169-184.
- [43] Minkowski H.: *Volumen und Oberflache*, Math. Ann. Vol. 57. 1903, str. 447-459
- [44] Mokhtarian F., Abbasi S., Kittler J.: *Robust and Efficient Shape Indexing through Curvature Scale Space*, Proc. British Machine Vision Conference, 1996, str. 53-62.
- [45] Munsell A. H.: *A Grammar of Color*, Van Nostrand-Reinhold, NY, 1969.
- [46] Niblack W., Flickner M. i inn.: *The QBIC Project: Querying Images by Content Using Colour, Texture and Shape*, SPIE, Vol. 1908, 1993, str. 173-187.

- [47] Ogle V., Stonebraker M.: *CHABOT: Retrieval from a Relational Database of Images*, IEEE Computer, Vol. 28, Nr. 9, sierpień 1995, str. 40-48
- [48] Omri N., Chouigui M-N.: *Linguistic Variables Definition by Membership Function and Measure of Similarity*, Proceedings of the 14th International Conference on System Science, Vol. 2, Wrocław, wrzesień 2001, str. 264-273
- [49] Park G., Baek Y., Lee H-K.: *Web image retrieval using majority-based ranking approach*, Multimedia Tools and Applications, t. 31, Nr. 2 Springer Netherlands, 2006. (w druku).
- [50] Pea S. C., Lin C. N.: *The detection of dominant points on digital curves by scale-space filtering*, Pattern Recognition, Vol. 25, Nr. 11, 1992, 1307-1314.
- [51] Pratt W. K.: *Digital Image Processing*, Wiley, 1991, str. 276-284
- [52] Prewitt J.: Object Enhancement and Extraction, in Lipkin B. and Rosenfeld A. eds, *Picture Processing and Psychopictorics*, Academic Press, NY, 1970.
- [53] Rege M., Dong M., Fotouhi F.: *Building a user-centered semantic hierarchy in image databases*, Multimedia Systems, Vol. 13, 2006, Springer-Verlag, (in printing).
- [54] Russ J. C.: *The image processing. Handbook*, CRC, Londyn, 1995.
- [55] Ruttimann U., Webber R.,L., Hazelrig J.,B.: Fractal Dimension from Radiographs of Peridental Alveolar Bone. A Possible Diagnostic Indicator of Osteoporosis. *Oral Surg Oral Med Oral Pathol*, 1992, str. 74-98.
- [56] Soares L. D., Pereira F.: *MPEG-7 Based Spatial Shape Concealment Difficulty*, Instituto de Telecomunicações, Instituto Superior Técnico, Lisboa, Portugalia
- [57] Sobel G., *Multivariate Observations*, Wiley, 1984.
- [58] Stapor K., Brueckner A.: *Mean Shift Segmentation, Genetic Algorithms and Support Vector Machines for Identification of Glaucoma in Fundus Eye Images*, Computer Recognition Systems, Proceedings of the 4th International Conference on Computer Recognition Systems CORES'05, 2005, str 679-685
- [59] Stricker M., Orengo M.: *Similarity of Colour Images*, SPIE Vol. 2420, str.381-392
- [60] Tan K-L., Ooi B. Ch., Yee Ch. Y.: *An Evaluation of Color-Spatial Retrieval Techniques for Large Image Databases*, Multimedia Tools and Applications, Vol. 14, Springer Netherlands, 2001, str. 55-78.
- [61] Teague M., R.: *Image analysis via the general theory of moments*, Journal of Optic Soc. Am., Vol 70, Nr. 8, 1980, str. 920-930.

- [62] Tollari S., Glotin H., Mairte J.: *Enhancement of Textual Images Classification Using Segmented Visual Contents for Image Search Engine*, Multimedia Tools and Applications, Vol. 25, No 3, Springer Netherlands, 2005, str. 405-417.
- [63] Undrill P. E., Delibassis K. i Cameron G. G., *Using genetic algorithms to design 2D filters for texture interpretation and image restoration in the presence of noise*. In: Proc. IEE Colloquium on 'Pattern Recognition', Digest 1997/018, IEE Press, London, 1997, str. 4/1 - 4/6
- [64] Wald G.: *The receptors for Human Color Vision*, Science, Nr. 145, 1964, str. 1007-1017
- [65] Wang J. Z., Wiederhold G. i in.: *Content-based image indexing and searching using Daubachies' wavelets*, International Journal on Digital Libraries, Vol. 1, Springer Verlag 1997, str. 311-328
- [66] Wang L.: *Support Vector Machines: Theory and Applications*, Springer, Berlin, 2005
- [67] Weszka J., S., Dyer C., R., Rosenfeld A.: *A Comparative study of Texture Measures for Terrain Classification*, IEEE Trans System s Man Cybernet, SMC, str. 6-269
- [68] Xu Ch., Prince J.: *Snakes, Shapes, and Gradient Vector Flow*, IEEE Transactions on Image Processing, Vol. 7, Nr. 3 Marzec 1998, str. 359-369.
- [69] Zauhar R. J., Mayona G., Welsh W. J.: *"Surface Signatures" – a New Approach for Fast Database Screening and Docking*, Wyd. Chemii i Biochemii Uniwersytety w Filadelfii, 2002.
- [70] Zhang D. S., Lu G.: *A comparative study of Fourier descriptors for shape representation and retrieval*, Proc. Fifth Asian Conference on Computer Vision (ACCV02), Melbourne, Australia, 2002, str. 646-651
- [71] Zhang H., Chen Z., et al.: *Relevance Feedback and Learning in Content-Based Image Search*, WWW: Internet and Web Information Systems, Vol. 6, 2003, str. 131-155.

13. Spis ilustracji

Rysunek 1	Ogólny schemat obrazowej bazy danych	8
Rysunek 2	Obraz rastrowy	11
Rysunek 3	Zmiana jasności obrazów przez „przesunięcie” histogramów	12
Rysunek 4	Przykład obrazu domu jednorodzinnego przed (po lewej) i po zmianie histogramu (po prawej). Wartości kolorów zostały obliczone według wzoru (4), a następnie uśredniono każde trzy sąsiednie poziomy.	13
Rysunek 5	Wykres rodzaju krawędzi (u góry), pierwszej pochodnej (na środku) i drugiej pochodnej (u dołu).....	14
Rysunek 6	Wyznaczenie krawędzi w sposób zaproponowany przez Sobela	15
Rysunek 7	Krawędzie wyznaczone algorytmem Canny.....	16
Rysunek 8	a) 4-sąsiedztwo względem punktu x_0 , b) 8-sąsiedztwo względem punktu x_0	17
Rysunek 9	Obraz binarny przed i po dylatacji elementem strukturalizującym 3×3	18
Rysunek 10	Obraz binarny przed i po erozji elementem strukturalizującym 3×3	18
Rysunek 11	Obraz binarny przed i po operacji domknięcia elementem strukturalizującym 3×3	19
Rysunek 12	Obraz binarny przed i po operacji otwarcia elementem strukturalizującym 3×3	19
Rysunek 13	Przykład obrazu o bimodalnym histogramie. Ciemne tło tworzy jeden pik na takim histogramie, a jasne komórki drugi.	21
Rysunek 14	Bimodalny histogram z zaznaczonym progiem T . Punkty o mniejszym poziomie szarości uznajemy jako tło, a o większym jako punkty należące do poszukiwanych obiektów.	21
Rysunek 15	Przestrzeń kolorów RGB (traktowana jako trójwymiarowy histogram).....	22
Rysunek 16	Obraz oryginalny narożnego okna.....	22
Rysunek 17	Efekt segmentacji algorytmem c-środków dla $c = 6$ z nałożonymi krawędziami dla lepszego podkreślenia segmentów. Każdy poziom szarości reprezentuje jeden segment.	23
Rysunek 18	Efekt segmentacji algorytmem c-środków dla $c = 12$	23
Rysunek 19	Efekt segmentacji algorytmem c-środków dla $c=5$. Segmentacja jest wykonywana dla poszczególnych składowych RGB	24
Rysunek 20	Efekt segmentacji algorytmem c-środków dla $c=5$. Przed segmentacją wartości RGB dla każdego piksela zostały zsumowane.....	24
Rysunek 21	Efekt segmentacji algorytmem c-środków dla $c=7$. Segmentacja jest wykonywana dla poszczególnych składowych RGB.	25
Rysunek 22	Efekt segmentacji algorytmem c-środków dla $c=7$. Przed segmentacją wartości RGB dla każdego piksela zostały zsumowane.....	25
Rysunek 23	Rozkład punktów z rys. 4 w przestrzeni kolorów RGB.	26
Rysunek 24	Rozkład punktów z rys. 16 w przestrzeni kolorów RGB.	26
Rysunek 25	Sposób podziału punktów na segmenty. Obszary I, II, III pokazują jasność punktu, natomiast największa wartość z trójki (R,G,B) określa kolor segmentu, do którego przyporządkowujemy dany punkt.	27
Rysunek 26	Segmentacja oparta na kolorach dla obrazu domu jednorodzinnego.....	28
Rysunek 27	Segmentacja oparta na kolorach dla obrazu narożnego okna	28
Rysunek 28	Oryginalny obraz domu z garażem.....	28
Rysunek 29	Segmentacja oparta na kolorach dla obrazu domu z garażem.	28
Rysunek 30	Oryginalny obraz rezydencji	29
Rysunek 31	Segmentacja oparta na kolorach dla obrazu rezydencji.....	29

Rysunek 32	Segment w kolorze średnim niebieskim dla rezydencji, na podstawie którego są wydzielane poszczególne obiekty	31
Rysunek 33	Dach i dwa górne okna rezydencji z okiennicami wydzielone z niebieskiego segmentu i przedstawione w kolorach oryginalnych.....	32
Rysunek 34	Okno górne z okiennicami wydzielone z warstwy niebieskiej rezydencji i przedstawione w kolorach oryginalnych	32
Rysunek 35	Okno dolne z okiennicami wydzielone z warstwy niebieskiej rezydencji i przedstawione w kolorach oryginalnych	33
Rysunek 36	Niebo nad domem jednorodzinny wydzielone z warstwy średniej niebieskiej i przedstawione w kolorach oryginalnych.....	33
Rysunek 37	Komin domu jednorodzinny wydzielony z warstwy średniej czerwonej i przedstawiony w kolorach oryginalnych	33
Rysunek 38	Balustrada balkonu domu jednorodzinny wydzielona z warstwy białej i przedstawiona w kolorach oryginalnych	34
Rysunek 39	Ściany domu jednorodzinny wydzielone z warstwy średniej czerwonej i przedstawione w kolorach oryginalnych.....	34
Rysunek 40	Ceglane ściany wokół narożnego okna wydzielone z warstwy czerwonej i przedstawione w kolorach oryginalnych	34
Rysunek 41	Blacha nad narożnym oknem wydzielona z warstwy niebieskiej i przedstawiona w kolorach oryginalnych.....	34
Rysunek 42	Obiekty wydzielone z domu jednorodzinny i przedstawione w kolorach średnich.....	36
Rysunek 43	Obiekty wydzielone z domu z garażem i przedstawione w kolorach średnich 36	
Rysunek 44	Obiekty wydzielone z narożnego okna i przedstawione w kolorach średnich bez refleksów w szybach	36
Rysunek 45	Obiekty wydzielone z rezydencji i przedstawione w kolorach średnich.....	37
Rysunek 46	Przykładowe tekstury naturalne a) tkanina, b) kamień, c) marmur i d) korek 38	
Rysunek 47	a) oryginalny obraz faktury kolorowej, b) rozkład energii widma dla tego obrazu. Kierunkowa natura tej tekstury znajduje odbicie w kierunkowym rozkładzie energii widma.	40
Rysunek 48	a) Filtr kołowy dzielący dziedzinę częstotliwościową na okręgi, według których wyznaczano gładkość lub chropowatość tekstury; b) filtr kątowy, który wskazywał kierunkowość tekstury.	41
Rysunek 49	Schemat rozpoznawania tekstur przy użyciu transformacji Fouriera i algorytmów genetycznych.....	43
Rysunek 50	Obraz oryginalny. Dach traktujemy jako obiekt pokryty teksturą	44
Rysunek 51	Segmentacja oparta na kolorach	45
Rysunek 52	Segment w kolorze jasnym czerwonym, z którego wydzielono dach.....	45
Rysunek 53	Przekrój przez detale horyzontalne dla 30-tej kolumny (falka Haara, $j=1$) dla ceglanej ściany narożnego okna pokazanej na rys. 40.	46
Rysunek 54	Przekrój przez detale horyzontalne dla 100-tej kolumny (falka Haara, $j=1$) dla dachu pokazanego na rys. 50.	47
Rysunek 55	Odległości wyznaczone dla progu odcięcia 2% wartości maksymalnej dla całej macierzy detali dla wielkości cegieł dla ściany z rys. 40.....	47
Rysunek 56	Odległości wyznaczone dla progu odcięcia 2% wartości maksymalnej dla całej macierzy detali dla wielkości dachówki z rys. 52.....	48
Rysunek 57	Odległości wyznaczone dla progu odcięcia 2% wartości minimalnej dla całej macierzy detali dla wielkości dachówki z rys. 52.....	48

Rysunek 58	Mapa odległości górnych, czyli policzonych na podstawie maksymalnych wartości detali horyzontalnych.....	49
Rysunek 59	Mapa odległości dolnych, czyli policzonych na podstawie minimalnych wartości detali horyzontalnych.....	49
Rysunek 60	Mapa odległości górnych czyli policzonych na podstawie maksymalnych wartości detali horyzontalnych.....	49
Rysunek 61	Mapa odległości dolnych, czyli policzonych na podstawie minimalnych wartości detali horyzontalnych.....	50
Rysunek 62	Mapa odległości górnych czyli policzonych na podstawie maksymalnych wartości detali horyzontalnych.....	50
Rysunek 63	Mapa odległości dolnych, czyli policzonych na podstawie minimalnych wartości detali horyzontalnych.....	51
Rysunek 64	Obiekty wydzielone z warstwy czerwonej i przedstawione w kolorach średnich.....	51
Rysunek 65	Reprezentacja kształtu w oparciu o "róże wiatrów".....	53
Rysunek 66	Idea wyznaczania sygnatury kształtu.....	54
Rysunek 67	a) obraz oryginalny, b), d), f), h), j) iteracyjne wygładzanie kształtu przy zmianach odchylenia standardowego w operatorze Gaussa odpowiednio dla $\sigma = 1, 4, 7, 10, 12, 14, c), e), g), i), k)$ poszczególne etapy tworzenia mapy konturu. [56].....	56
Rysunek 68	Mapa konturu (po lewej) i wyekstrahowane z niej piki (po prawej).....	56
Rysunek 69	Przykładowy kształt i jego rozwinięcie w funkcję odległości konturu od środka tego obiektu.....	57
Rysunek 70	a) przykład zwartego kształtu, b) przykład mało zwartego kształtu.....	58
Rysunek 71	Przykłady szkielegowania dla różnych kształtów.....	60
Rysunek 72	Zaznaczony obiekt (dach) jest odwzorowywany na „mapie” regionów, a następnie jest zapisywany w postaci drzewa z zaznaczonymi przez kwadraty elementami, które wyznaczają położenie obiektu w całym obrazie.....	61
Rysunek 73	Przykładowa organizacja topologii w bazach danych.....	65
Rysunek 74	Struktura indeksu obrazowej bazy danych. Szare pola na poziomie cech pokazują obszar poszukiwań, a ciemne pola to wyszukane obrazy.....	67
Rysunek 75	Przykład struktury R-drzewa i jego reprezentacja przestrzenna (po lewo).....	68
Rysunek 76	Obiekty w obrazie mogą mieć cechy A (zielone), cechy S (pomarańczowe) lub obie te cechy na raz (pomarańczowe z zieloną ramką).....	75
Rysunek 77	Przykłady skupień z zaznaczonymi centroidami „+” w przypadku dwuwymiarowym.....	81
Rysunek 78	Standardowa S funkcja.....	83
Rysunek 79	Segmentacja obrazu oparta na funkcji przynależności $\mu_X(x_{mn})$ typu S.....	83
Rysunek 80	Przykładowe skupienia z zaznaczonymi centroidami i zaznaczonymi funkcjami przynależności dla punktów najbliższych tym centroidom w danych skupieniach.....	86
Rysunek 81	Linie separujące dla dwóch klas (w przypadku wielowymiarowym hiperpłaszczyzny separujące). Po lewo linia z małym marginesem M i po prawo linia z dużym marginesem M, który docelowa chcemy uzyskać.....	87
Rysunek 82	Przykład konturów dla obiektu niejednorodnego topologicznie.....	89
Rysunek 83	Przykład rozkładu gradientów sił na aktywnym konturze.....	91
Rysunek 84	Przykład fałek Daubechies.....	91
Rysunek 85	Konwencjonalny układ detali horyzontalnych d_j^1 , wertykalnych d_j^2 i diagonalnych d_j^3 przy wizualizacji dwuwymiarowej transformaty falkowej (a_3 oznacza aproksymację obrazu).....	93

Rysunek 86	Przykładowy obraz (po lewej) i jego transformacja falkowa w układzie współczynników jak na rys. 85 (po prawej).....	93
Rysunek 87	Ideowy schemat konfiguracji pomiędzy serwerami obsługującymi obrazową bazę danych.....	96

