

163/2005

Raport Badawczy

RB/37/2005

Research Report

Obrazowa baza danych

T. Jaworska, A. 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 2005

INSTYTUT BADAŃ SYSTEMOWYCH PAN
PRACOWNIA SYSTEMÓW INTELIGENTNYCH

Tatiana Jaworska
Andrzej Partyka

Obrazowa baza danych

Zadanie: A3.1.3/2005

Inteligentne metody analizy danych i odkrywania wiedzy dla
potrzeb wspomagania decyzji

Warszawa, listopad 2005

Spis treści

1. Wstęp.....	5
2. Ogólna charakterystyka obrazowych bazy danych	6
2.1. Wstępna obróbka obrazu.....	9
2.1.1. Modyfikacja histogramu.....	9
2.1.2. Segmentacja	10
2.2. Wydzielenie obiektów z tła.....	22
2.3. Model danych.....	25
2.4. Indeksowanie danych.....	28
2.4.1. Filtrowanie.....	29
2.4.2. R-drzewa.....	30
2.4.3. Indeksowanie w przestrzeni transformat.....	31
2.5. Przeszukiwanie danych – zapytania do bazy danych	34
2.5.1. Miary oparte na metrykach.....	36
2.5.2. Rozmyte miary podobieństwa.....	38
2.5.3. Miary oparte na łącznym rozkładzie prawdopodobieństwa.....	39
3. Instrukcja obsługi użytkownika.....	41
4. Dokumentacja techniczna programu.....	45
4.1. Potrzebne pliki i ich lokalizacja.....	45
4.2. Sposób wybierania koloru.....	46
4.3. Konfiguracja oprogramowania.....	46
5. Podsumowanie.....	48
6. Bibliografia.....	49
7. Spis ilustracji.....	52

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 łączny 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 zajmiemy się budową bazy zdjęć nieruchomości. W tym celu zostaną wykorzystane zdjęcia domów jednorodzinnych pobrane z Internetu w formacie JPEG. Następnie przedstawiona będzie struktura bazy danych i procedury analizy obrazów w celu wyselekcjonowania elementów elewacji na zgromadzonych zdjęciach domów. Każdy rekord bazy poza obrazem zawiera również dane tekstowe. Baza danych i mechanizmy analizy obrazu są zaimplementowane z pomocą pakietu MatLab (wersja 7.1) ze wspomaganiami następujących ToolBoxów: ImageProcessing, Fuzzy, Statistics, Wavelet i DataBase. Baza danych jest dostępna w trybie interaktywnym przez Internet. Interfejs internetowy jest wykonany w Perlu. Internauci nie działają bezpośrednio na MatLabie, a jedynie uruchamiają go *off-line* poprzez interfejs 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 przedstawione podstawowe operacje wykonywane na obrazowych bazach danych i omówione dotychczas stosowane algorytmy.

Historycznie jako pierwsze pojawiło się atrybutowe podejście do obrazowych bazach danych [11], [15], 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.

Zupełnie inne podejście reprezentują obecnie powstające obrazowe bazy danych, które opierają się na automatycznym rozpoznawaniu treści obrazu. W tym podejściu brane są pod uwagę dwie podstawowe grupy cech: pierwotne i logiczne. Cechy pierwotne to te, które są uzyskiwane bezpośrednio z obrazu, czyli takie jak: kolor, tekstura, kształt, powierzchnia, itp. Cechy logiczne nie są bezpośrednio związane z pikselami, z których składa się obraz i obejmują elementy semantyczne takie jak: położenie obiektów względem siebie, relacje sąsiedztwa itp. W celu określenia cech logicznych trzeba najpierw wydzielić z obrazu obiekty takie jak domy, samochody, drzewa itp. W tym celu prowadzi się obróbkę wstępną obrazu dołączanego do bazy. Obróbka wstępna może obejmować:

- modyfikację histogramu
- segmentację obrazu
- analizę skupień
- wydzielenie krawędzi
- odszumienie lub inne techniki poprawiania jakości obrazu
- wydzielenie obiektów w obrazie

W momencie dołączania nowego obrazu do bazy następuje półautomatyczna, a w bardziej zaawansowanych wersjach automatyczna analiza cech pierwotnych i obróbka wstępna obrazu, która pozwala na indeksowanie, a następnie wyszukiwanie obrazów za pomocą zapytań odwołujących się do cech występujących w nich obiektów [12]. Przy tworzeniu obrazowej bazy danych trzeba określić jakie cechy będą z obrazu ekstrahowane,

aby w możliwie najpełniejszy sposób móc potem korzystać ze zgromadzonych zasobów. Najczęściej wykorzystywane cechy pierwotne obrazu to:

- kolor
- tekstura
- obrys
- krawędzie
- kształt
- powierzchnia
- moment bezwładności
- osie symetrii
- środek ciężkości

Natomiast do cech logicznych zaliczamy między innymi:

- sąsiedztwo obiektów
- relacje większości-mniejszości
- interpretacja rodzaju obiektu

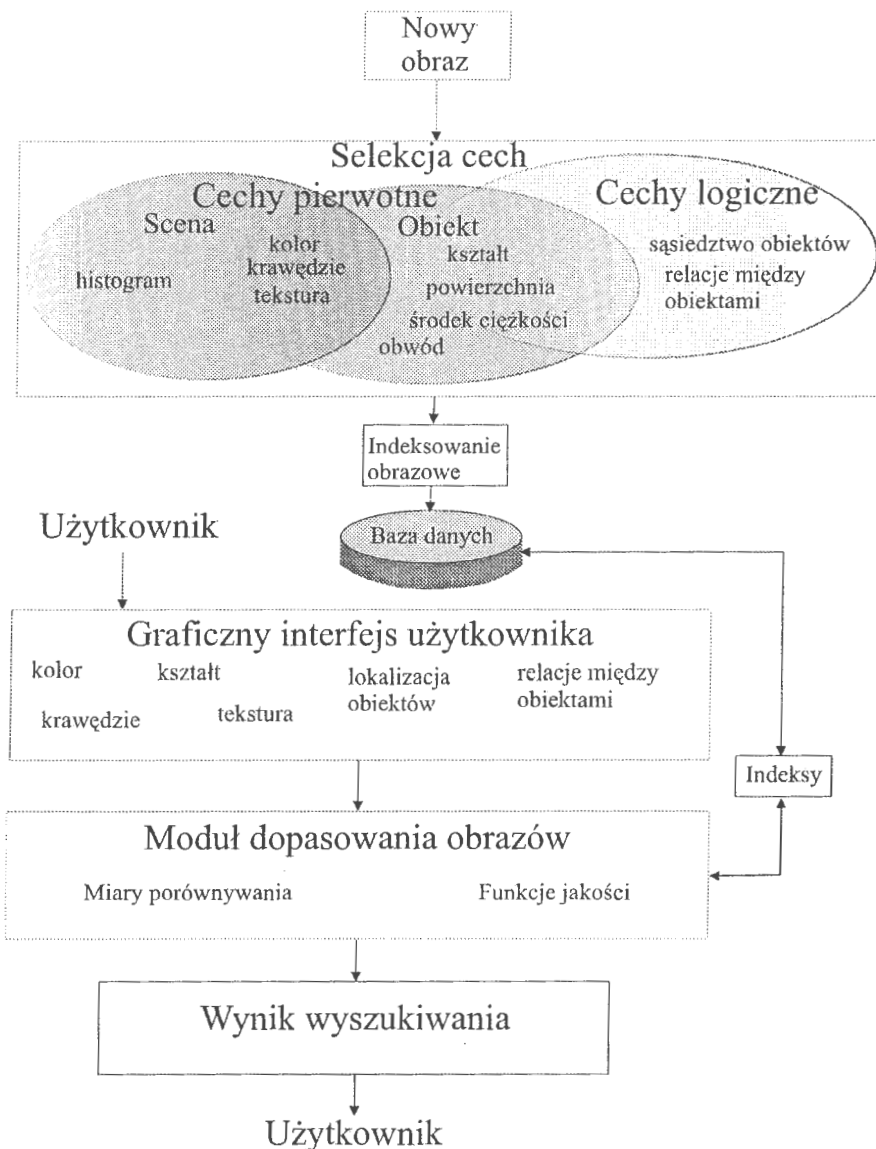
Przy opisie obrazów poza cechami pierwotnymi i logicznym wykorzystuje się też atrybuty tekstowe np.:

- nazwę obrazu
- datę uzyskania
- właściciela lub inne cechy pochodzenia
- powiązanie z innymi obrazami (film video)
- przyporządkowaną kategorię

co łącznie tworzy metadane przechowywane w bazie. Do zadawania pytań do bazy obrazowej musi zostać stworzony interfejs graficzny użytkownika, ponieważ musi on mieć możliwość łatwego formułowania zapytań o cechy pierwotne obrazu.

Indeksowanie obrazowej bazy danych powinno się opierać w pierwszym rzędzie na zawartości obrazów, a nie na opisie tekstowym. Jest to jedno z trudniejszych zagadnień po samym rozpoznawaniu obrazu.

Kolejnym etapem użytkowania takiej bazy jest jej przeszukiwanie w zależności od zadawanych zapytań. Muszą być stworzone miary umożliwiające porównywanie obrazów i formalizm do formułowania zapytań tak, by użytkownik otrzymywał obraz najbardziej zbliżony to tego, którego szuka. Rysunek 1 przedstawia ogólny schemat omawianej bazy danych.



Rysunek 1 Ogólny schemat obrazowej bazy danych

2.1 Wstępna obróbka obrazu

2.1.1 Modyfikacja histogramu

Histogram jest to funkcja pokazująca jaka liczba pikseli w obrazie ma określony poziom szarości. Na osi rzędnych odkładamy liczbę pikseli, a na osi odciętych odkładamy numery poziomów szarości (patrz przykład na rys. 2).

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, którego histogram opisujemy przez:

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

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

g, g_{\max} – najwyższy poziom szarości,

g_{\min} – najniższy poziom szarości.

Nowy obraz z histogramem opisanym jako:

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

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

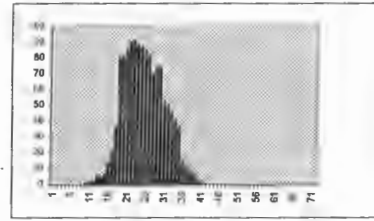
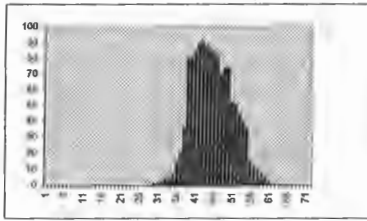
f, f_{\max} – najwyższy poziom szarości,

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. 2.



Rysunek 2 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:

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

Można też dodatkowo uśrednić sąsiadujące ze sobą wartości co daje efekt zwiększonej ostrości widoczny na rys. 3. Histogram bywa modyfikowany nie tylko w sposób liniowy, ale również w sposób wykładniczy, logarytmiczny lub hiperboliczny, co w określonych przypadkach pozwala uzyskać poszukiwane cechy obrazu[27], [5].



Rysunek 3 Przykład rysunku przed (po lewej) i po zmianie histogramu (po prawej). Wartości kolorów zostały obliczone według wzoru (4), a następnie uśredniono każde trzy sąsiednie poziomy do środkowego.

2.1.2 Segmentacja

Aby w procesie dalszej analizy wydzielić z obrazu jakiegokolwiek klasy należy najpierw przeprowadzić segmentację obrazu. Segmentacja jest to podział obrazu na rozdzielne

regiony. W przypadku obrazów rastrowych region jest to połączony zbiór pikseli. Każdy region charakteryzowany jest przez określone atrybuty. Najczęściej używanymi atrybutami przy segmentacji są:

- Jasność dla obrazów monochromatycznych
- Kolor dla obrazów barwnych
- Tekstura
- Krawędzie

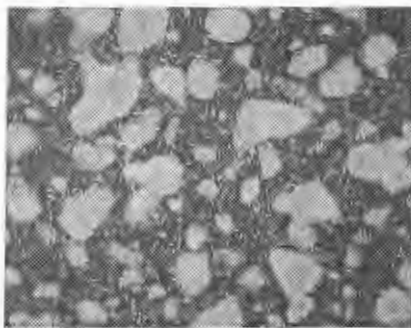
Możemy wyróżnić trzy różne filozofie segmentacji:

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 potrzebach segmentacji.

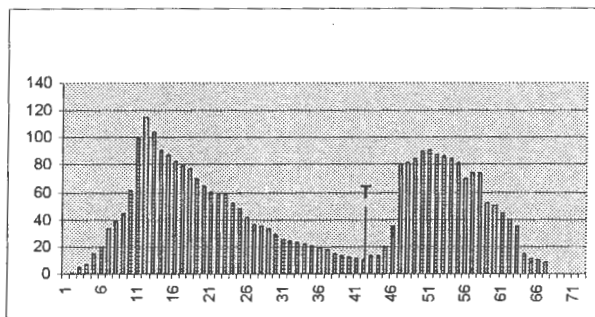
2.1.2.1 Segmentacja oparta na histogramie bimodalnym

Do podejścia opartego na regionach możemy zaliczyć progowanie. Polega ono na wyborze obiektów wyraźnie odcinających się od tła jak na rysunku 4. Obrazy z wyraźnym rozróżnieniem obiektów i tła mają histogram bimodalny (patrz rys. 5) co oznacza, że jeden z pików histogramu odpowiada za punkty należące do obiektów, a drugi - za punkty należące do tła.



Rysunek 4 **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. 5. Metoda ta niestety przestaje poprawnie działać w przypadku obrazów zaszumionych.



Rysunek 5 Bimodalny histogram z zaznaczonym progiem T . Punkty o mniejszym poziomie szarości uznajemy jako tło, a o większym jako punkty należące do poszukiwanych obiektów.

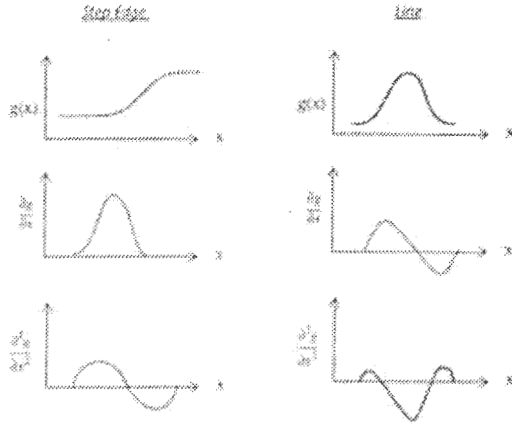
2.1.2.2 Segmentacja oparta na krawędziach

Jeśli segmentację opieramy na wyznaczonych krawędziach najpierw musimy się przyjrzeć algorytmom wyznaczania samych 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. 6.



Rysunek 6 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 $G(i,j)$:

$$\mathbf{H} = \mathbf{G} \star \mathbf{M} \quad (6)$$

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

W przypadku obliczania dyskretnego laplasjanu $H(i,j)$ używa się następujących macierzy \mathbf{M} :

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

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

2.1.2.3 Analiza skupień

Wykorzystywanie analizy skupień jest popularnym podejściem do segmentacji.[23] Wynikiem analizy skupień jest 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) $P = \{C_1, C_2, \dots, C_c\}$, 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:

$$f(P, V) = \sum_{j=1}^c \sum_{m=1}^M \sum_{n=1}^N \|x_{mn} - v_{C_j}\|^2 \quad (8)$$

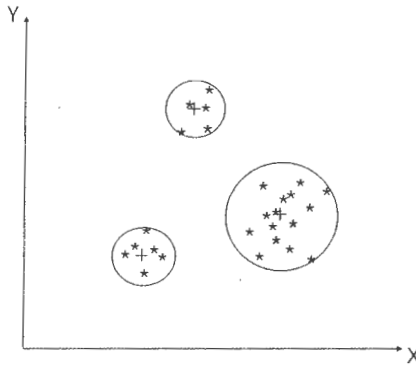
gdzie v_{C_j} - są środkami klastrów C_j dla $j = 1, \dots, c$.

Jako metrykę przyjmujemy metrykę euklidesową. Próbuje się zatem znaleźć takie centroidy v_{C_j} , które minimalizują wartość funkcji (8). 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).

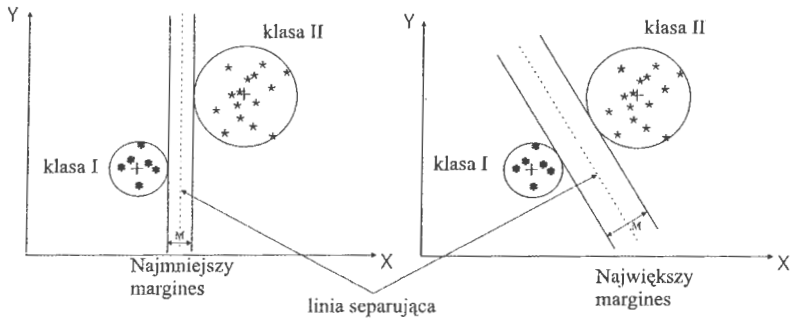
Algorytm analizy skupień jest zatem zbieżny do lokalnych minimów funkcji $f(P, V)$. Poglądowo skupienia zostały zobrazowane na rys. 8.



Rysunek 7 Przykłady skupień z zaznaczonymi centroidami „+” w przypadku dwuwymiarowym

2.1.2.4 Segmentacja oparta na metodzie wektorów wspierających (Support Vector Machines SVM)

W tym podejściu do segmentacji możemy posłużyć się metodą wektorów wspierających (SVMs wprowadzonych po raz pierwszy przez Vapnika w 1995r. [6]) w celu znalezienia optymalnej hiperpłaszczyzny separującej (OSH), czyli takiej hiperpłaszczyzny, która znajduje maksymalny przedział/ margines M pomiędzy szukanymi klasami [28].



Rysunek 8 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 docelowo chcemy uzyskać.

Jeśli dla łatwiejszego zrozumienia uprościmy sobie problem do dwóch klas $\mathbf{x} \in \mathbf{R}^2$ to możemy go przedstawić graficznie [33] jak na rysunku 11. 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 $d(\mathbf{x}, \mathbf{w}, b)$.

$$d(\mathbf{x}, \mathbf{w}, b) = \mathbf{w}^T \mathbf{x} + b = \sum_{i=1}^n w_i x_i + b \quad (9)$$

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

Okazuje się, że uzyskujemy tę funkcję dobierając odpowiednio przykłady uczące $\mathbf{w} = [w_1 \ w_2 \ \dots \ w_n]^T$ w kolejnych etapach uczenia. Linia przerywana na rys. 11 jest opisana jako $d(\mathbf{x}, \mathbf{w}, b) = 0$. W procesie uczenia korzystamy z przykładowych punktów \mathbf{x}_p i z funkcji wskaźnikowej (indicator function) danej wzorem:

$$i_F = o = \text{sign}(d(\mathbf{x}, \mathbf{w}, b)) \quad (10)$$

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

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

i

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

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

$$\min_{\mathbf{x}_j \in \mathbf{X}} |\mathbf{w}^T \mathbf{x} + b| = 1 \quad (11)$$

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:

$$M = \frac{2}{\|\mathbf{w}\|} \quad (12)$$

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:

$$y_i [\mathbf{w}^T \mathbf{x} + b] \geq 1 \quad \text{dla } i = 1, \dots, l \quad (13)$$

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

$$L(\mathbf{w}, b, \alpha) = \frac{1}{2} \mathbf{w}^T \mathbf{w} - \sum_{i=1}^I \alpha_i \{y_i [\mathbf{w}^T \mathbf{x}_i + b] - 1\} \quad (14)$$

gdzie: α_i są to mnożniki Lagrange'a, które w tym przypadku są wektorami nośnymi dla hiperpłaszczyzny separującej. Poszukujemy punktu siodłowego $(\mathbf{w}_0, b_0, \alpha_0)$, 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:

$$\frac{\partial L}{\partial \mathbf{w}_0} = 0 \quad \text{stąd} \quad \mathbf{w}_0 = \sum_{i=1}^I \alpha_i y_i \mathbf{x}_i \quad (15)$$

$$\frac{\partial L}{\partial b_0} = 0 \quad \text{stąd} \quad \sum_{i=1}^I \alpha_i y_i = 0 \quad (16)$$

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

$$\alpha_i \{y_i [\mathbf{w}^T \mathbf{x}_i + b] - 1\} = 0, \quad i = 1, \dots, I \quad (17)$$

Podstawiając równania (24) i (25) do równania (23) zmieniamy lagrangian $L(\mathbf{w}, b, \alpha)$ jednej zmiennej na lagrangian $L_d(\alpha)$ dualny i otrzymujemy:

$$L_d(\alpha) = \sum_{i=1}^I \alpha_i - \frac{1}{2} \sum_{i,j=1}^I \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j \quad (18)$$

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

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.

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

2.1.2.5 Segmentacja oparta 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 [2]. Algorytm segmentacji opartej na progowaniu przy zastosowaniu zbiorów rozmytych wygląda następująco:

Zbiór rozmyty A w przestrzeni U reprezentowany jest przez swoją funkcję przynależności $\mu_A \rightarrow [0,1]$. 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:

$$X = \{ \mu_X / x_{mn}; \quad m = 1, \dots, M, n = 1, \dots, N \} \quad (19)$$

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

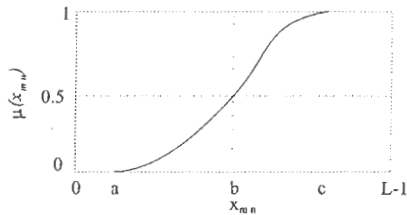
które z kolei są kryterium do określenia progów dla segmentowanego obrazu. Z całej tej gamy możliwości przyjrzymy się najprostszemu i najbardziej typowemu podejściu do segmentacji w oparciu o progowanie rozmyte, który przedstawia poniższy algorytm.

Krok 1

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

$$\mu_X(x_{mn}) = S(x_{mn}; a, b, c) = \begin{cases} 0 & \text{if } x_{mn} \leq a, \\ 2(x_{mn} - a)^2 / (c - a)^2 & \text{if } a \leq x_{mn} \leq b, \\ 1 - 2(x_{mn} - c)^2 / (c - a)^2 & \text{if } b \leq x_{mn} \leq c, \\ 1 & \text{if } x_{mn} \geq c. \end{cases} \quad (20)$$

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 9.



Rysunek 9 Standardowa S funkcja

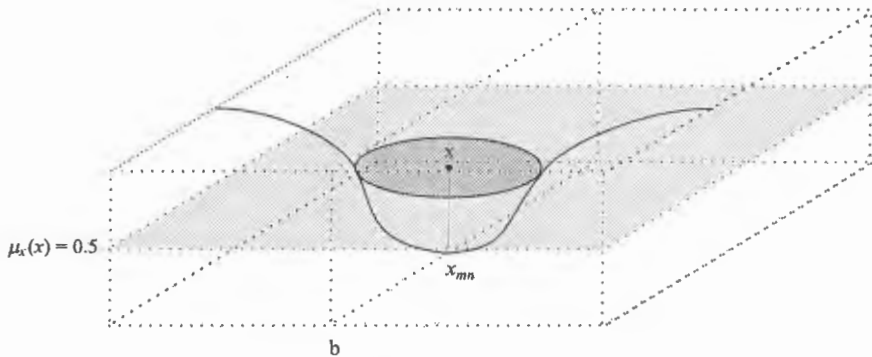
Krok II

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. 10).



Rysunek 10 Segmentacja obrazu oparta na funkcji przynależności $\mu_k(x_{mn})$ typu S

2.1.2.6 Rozmyta analiza skupień (fuzzy c-means clustering)

Natomiast rozmyta analiza skupień (fuzzy c-means clustering) [23] 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:

$$\bigwedge_{x_{mn} \in X} \sum_j \mu_{C_j}(x_{mn}) = 1 \quad (21)$$

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

$$\bigwedge_{x_{mn} \in X} \bigvee_{C_j \in P} \mu_{C_j}(x_{mn}) > 0 \quad (22)$$

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

$$f_X(P, U) = \sum_{j=1}^c \sum_{m=1}^M \sum_{n=1}^N [\mu_{C_j}(x_{mn})]^q (x_{mn} - v_{C_j})^2 \quad (23)$$

gdzie:

v_{C_j} - są centroidami C_j gdzie $j = 1, \dots, c$,

$U = \{ \mu_{C_j} \in [0, 1] \}$ - zbiór stopni przynależności punktów x_{mn} do klastra C_j ,

$q > 1$ - współczynnik rozmytości skupień.

Jako metrykę przyjmujemy metrykę euklidesową. Funkcja (23) przyjmuje wartości niezerowe. Natomiast gdyby nie było warunku (21) 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:

$$\frac{\partial f_X(P, U)}{\partial \mu_{C_j}} = q[\mu_{C_j}(x_{mn})]^{q-1} \sum_{m=1}^M \sum_{n=1}^N (x_{mn} - v_{C_j})^2 = 0 \quad \text{dla każdego } j \quad (24)$$

$$\frac{\partial f_X(P, U)}{\partial v_{C_j}} = \sum_{j=1}^c \sum_{m=1}^M \sum_{n=1}^N [\mu_{C_j}(x_{mn})]^q 2(x_{mn} - v_{C_j}) = 0 \quad (25)$$

Stąd obliczamy v_{C_j} dla danego j :

$$v_{C_j} = \frac{\sum_{m=1}^M \sum_{n=1}^N \mu_{C_j}(x_{mn})^q x_{mn}}{\sum_{m=1}^M \sum_{n=1}^N \mu_{C_j}(x_{mn})^q} \quad (26)$$

Natomiast przyrównując do siebie dwie pochodne (24) dla $j \neq i$ otrzymujemy:

$$q[\mu_{C_j}(x_{mn})]^{q-1} \sum_{m=1}^M \sum_{n=1}^N (x_{mn} - v_{C_j})^2 = q[\mu_{C_i}(x_{mn})]^{q-1} \sum_{m=1}^M \sum_{n=1}^N (x_{mn} - v_{C_i})^2 \quad (27)$$

$$\mu_{C_j}(x_{mn}) = \mu_{C_i}(x_{mn}) \left[\frac{\sum_{m=1}^M \sum_{n=1}^N (x_{mn} - v_{C_i})^2}{\sum_{m=1}^M \sum_{n=1}^N (x_{mn} - v_{C_j})^2} \right]^{\frac{1}{q-1}} \quad (28)$$

Z własności (21) otrzymujemy:

$$\mu_{C_i}(x_{mn}) = \frac{1}{\sum_{j=1}^c \left[\frac{\sum_{m=1}^M \sum_{n=1}^N (x_{mn} - v_{C_j})^2}{\sum_{m=1}^M \sum_{n=1}^N (x_{mn} - v_{C_i})^2} \right]^{\frac{1}{q-1}}} \quad (29)$$

Na podstawie równań (26) i (29) można napisać algorytm analizy skupień z iteracyjnym liczeniem funkcji przynależności. W 1990 Bezdek udowodnił jego zbieżność dla $q > 1$:

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} .

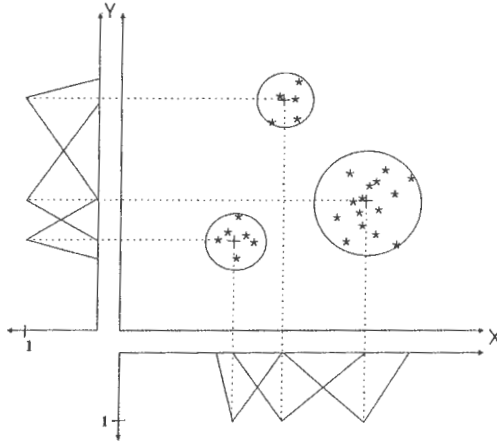
Krok 3: Liczymy funkcję przynależności według wzoru (29).

Krok 4: Uaktualniamy wstępne centroidy v_{c_i} w V według wzoru (26).

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 11 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 11 pokazane są funkcje przynależności dla przykładowych punktów leżących najbliżej centroidów 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.

Niezależnie od wprowadzanych przez Bezdeka i jego następców miar efektywności tego algorytmu do naszych celów przyjmuje się $c = 5, \dots, 7$. Wynik działania tego algorytmu widoczny jest na zdjęciu domu na rysunku 12.



Rysunek 12 Efekt segmentacji algorytmem rozmytych c-środków dla $c = 6$ z nałożonymi krawędziami dla lepszego podkreślenia segmentów. Każdy poziom szarości reprezentuje jeden segment.

2.2 Wydzielenie obiektów z tła

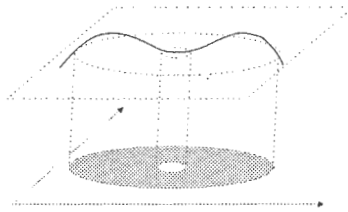
Jeśli obrazy mają bimodalny histogram to łatwo jest wydzielić obiekty z tła przez opisany powyżej algorytm progowania histogramu (Patrz pkt. 2.1.2.). Jeśli wydzielimy z obrazu skupienia to również możemy wydzielić obiekt w postaci jednego lub kilku segmentów z tła. Należy wtedy jedynie znać element charakteryzujący segment obiektu np. kolor.

Następną metodą wydzielenia obiektów z tła są aktywne kontury. Kontur może być definiowany jako krzywa parametryczna, wielokąt, bądź krzywa sklejana (B-spline), jednakże występuje wtedy problemy z opisaniem obiektów niejednorodnych topologicznie (patrz rysunek 13). Dlatego też powstał model geodezyjny konturów aktywnych [4], w których nie występuje ten problem. Formalna definicję konturów możemy przedstawić jako:

Definicja 2

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

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



Rysunek 13 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}

$$J(C) = E_{int}(C) + E_{ext}(C) \quad (30)$$

Kształt modelu obiektu jest kontrolowany przez energię wewnętrzną, podczas gdy energia zewnętrzna jest potrzebna do opisu zewnątrz obiektu. Energia zewnętrzna jest obliczana przez całkowanie funkcji energii wzdłuż konturu czyli:

$$E_{ext} = \int_0^1 P(C(p)) dp = -\nabla P(C(p)) \quad (31)$$

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:

$$P(x,y) = -|\nabla G_\sigma(x,y) * I(x,y)|^2 \quad (32)$$

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 zginięcie się konturu i jego odpowiednią sztywność. Wagi α i β są odpowiednio parametrami używanymi do sterowania elastycznością i sztywnością. Zatem energia wewnętrzna jest opisana jako:

$$E_{int} = \frac{1}{2} \int_0^1 \alpha |C'(p)|^2 + \beta |C''(p)|^2 dp \quad (33)$$

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

$$\alpha C''(p) - \beta C'''(p) - \nabla P(C(p)) = 0 \quad (34)$$

W tym miejscu należy zauważyć (co zrobili Xu i Prince [34]), że (34) jest to równanie sił, a mapę krawędzi można traktować jako pole siły potencjalnej. Taki przykładowy rozkład sił na aktywnym konturze prezentuje rysunek 14. Zatem równanie sił możemy zapisać jako:

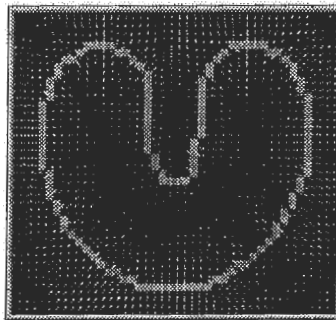
$$F_{\text{int}} + F_{\text{ext}}^{(p)} = 0 \quad (33)$$

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

Aby rozwiązać (34) trzeba potraktować „węża” jako krzywą dynamiczną, czyli zależną od czasu $C(p,t)$. Wtedy pochodna z C po czasie jest równa lewej stronie równania (34):

$$\frac{\partial C}{\partial t} = \alpha C''(p) - \beta C'''(p) - \nabla P(C(p)) \quad (34)$$

Kiedy rozwiązanie $C(p,t)$ jest stabilne, człon $\frac{\partial C(p,t)}{\partial t}$ znika i otrzymujemy rozwiązanie równania (34).



Rysunek 14 Przykład rozkładu gradientów sił na aktywnym konturze

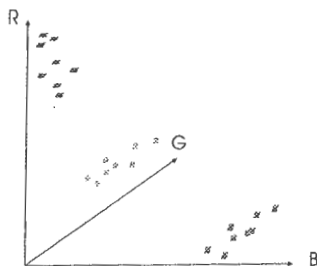
2.3 Modele danych

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 przez zamknięte kontury obiekty. 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:

- Kolor

- Tekstura
- Obwód
- Powierzchnia
- Środek ciężkości
- Momenty główne bezwładności
- Osie bezwładności
- Mimośród
- Kod granic
- Położenie w obrazie

Kolor jest cechą najlepiej chyba postrzeganą przez człowieka [31]. Dlatego też stworzono różne modele opisu koloru takiej jak RGB, VHS, HSI, itd. [24], [19] 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).



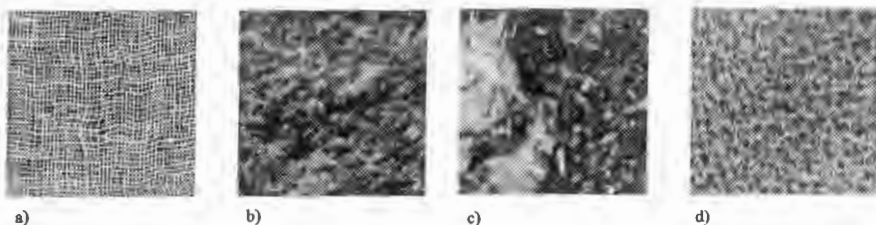
Rysunek 15 **Przestrzeń kolorów RGB (traktowana jako trójwymiarowy histogram)**

Tekstura naturalna według Pratta [27] to obrazy pochodzące z realnych scen zawierające samopowtarzające się układy pikseli np.: cegły, piasek, trawa, raffa, oraz tkanina, marmur, korek, itp. jak na rys. 16. Tekstura w ujęciu stochastycznym może być rozumiana jako realizacja dwuwymiarowego procesu stochastycznego. Proces stochastyczny $F(x,y,t)$ może być opisany dzięki znajomości jego łącznej gęstości prawdopodobieństwa

$$P_{t_1, \dots, t_J}(x_1, \dots, x_J, y_1, \dots, y_J) = P(F(t_1) = x_1, \dots, F(t_J) = x_J) \quad (35)$$

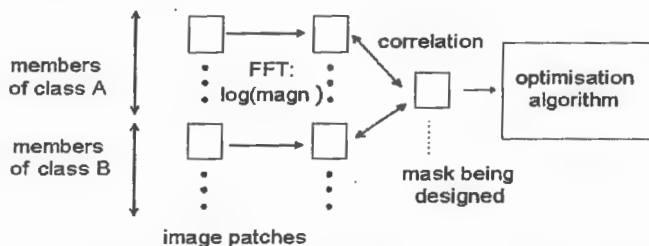
dla wszystkich próbek J gdzie (x_j, y_j, t_j) reprezentuje czasowo-przestrzenne próbki funkcji obrazu $F(x_j, y_j, t_j)$. W szczególności obraz możemy opisać jako dwuwymiarowy proces Markowa (pole Markowa). Przy takim podejściu naturalnymi parametrami

charakteryzującymi teksturę są parametry generującego ją procesu Markowa. Prekursorem opisu tekstury przy pomocy pól Markowa był Julesz [18].



Rysunek 16 Przykładowe tekstury naturalne a) tkanina, b) kamień, c) marmur i d) korek

Podjęmowane są też prace mające na celu rozróżnianie tekstur przy pomocy algorytmów genetycznych. [9], [30] 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 17.



Rysunek 17 Schemat rozpoznawania tekstur przy użyciu transformacji Fouriera i algorytmów genetycznych

Obwód liczymy jako liczbę pikseli, z których składa się kontur każdego obiektu, a powierzchnię jako sumę wszystkich pikseli wewnątrz konturu wraz z tym konturem.

Środek ciężkości obiektu (\bar{x}, \bar{y}) obliczamy jako $\bar{x} = \frac{1}{n} \sum_x x$ i $\bar{y} = \frac{1}{n} \sum_y y$,

momenty centralne bezwładności jako: $m_y = \sum_{k=1}^n (x_k - \bar{x})^l (y_k - \bar{y})^l$. Na podstawie momentów centralnych możemy obliczyć inne momenty bezwładności, które s:

niezmiennicze względem przesunięcia, skali i orientacji w przestrzeni co zapewnia nam dobrą reprezentację obiektu i łatwą jego rozpoznawalność. Przykładowe trzy liczymy następująco:

$$M_1 = m_{20} + m_{02}$$

$$M_2 = (m_{20} - m_{02})^2 + 4m_{11}^2$$

$$M_3 = (m_{30} - 3m_{12})^2 + (3m_{21} - m_{03})^2$$

Następną cechą charakteryzującą obiekt jest **mimośród** liczony według wzoru:

$$e = \frac{m_{20} + m_{02} + \sqrt{(m_{20} - m_{02})^2 + 4m_{11}^2}}{m_{20} + m_{02} - \sqrt{(m_{20} - m_{02})^2 + 4m_{11}^2}} \quad (36)$$

Kolejnym parametrem obiektu są **osie bezwładności**:

$$\theta = \frac{1}{2} \tan^{-1} \left(\frac{2m_{11}}{m_{20} - m_{02}} \right) \quad (37)$$

Do opisu **kształtu** można używać różnych technik, ale kodowanie kształtu „róża wiatrów” (co proponuje Castleman [5]) wydaje się jedną z prostszych acz skutecznych metod. Polega ona na przyporządkowaniu przybliżonemu obrysowi obiektu w postaci linii łamanej ośmiu kierunków, jak to pokazuje rysunek 18. W konsekwencji metoda to jest przydatna do indeksowania obiektów po kształcie (patrz pkt. 2.4).

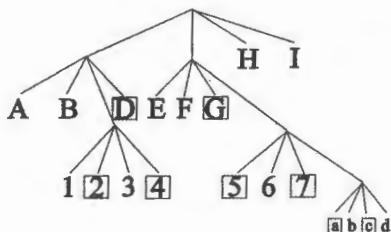


Rysunek 18 Reprezentacja kształtu w oparciu o "różę wiatrów"

Ostatnią cechą wyróżniającą obiekt jest jego położenie w obrazie. Jedną z proponowanych w literaturze metod jest reprezentacja w postaci drzewa, w której obraz dzielimy na połowy, a te połowy jeszcze na połowy i otrzymujemy coraz mniejsze regiony. Następnie zaznaczamy, który z regionów jest zajęty przez opisywany obiekt. Ilustrację tego procesu widać na rysunku 19.



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



Rysunek 19 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.

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.

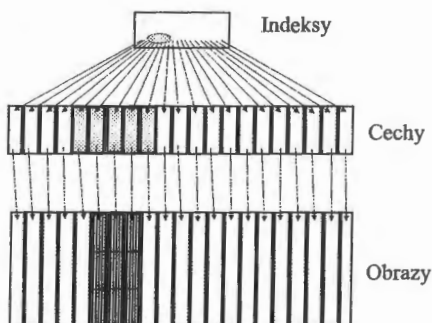
2.4 Indeksowanie danych

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. Weźmy pod uwagę przypadek, w którym zapytanie do bazy dotyczy podobieństwa kolorów. Takie zapytanie jest definiowane jako odległość w wielowymiarowej przestrzeni cech (patrz rozdz. 2.5.1.), tak więc indeksowanie musi dotyczyć cech obrazu lub cech poszczególnego obiektu. Najlepiej sprawdzone wydają się dwie metody: filtrowanie [1] i indeksowanie metodą R-drzew [22]. Obiecująco wyglądają też metody oparte na transformacjach: θ -transformacji [20] i transformacji falkowej [32].

2.4.1. Filtrowanie

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

- cały obraz
- wydzielone cechy obrazu lub obiektu
- indeksy



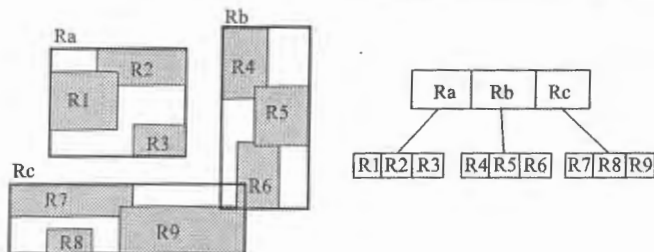
Rysunek 20 **Struktura indeksu obrazowej bazy danych. Szare pola na poziomie cech pokazują obszar poszukiwań, a ciemne pola to wyszukane obrazy.**

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 20.

2.4.2. R-drzewa

Następną metodą indeksowania i zarządzania danymi przestrzennymi są R-drzewa zaprojektowane przez Guttmana w 1984 r. [13]. Pierwotna propozycja dotyczyła obiektów prostokątnych, ale następnie została rozszerzona na obiekty w przestrzeniach wielo-

wymiarowych. 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 21.



Rysunek 21 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)

2.4.3. 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.

2.4.3.1. θ -transformacja

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:

$$x = x_n 2^{n-1} + \dots + x_2 2^1 + x_1 2^0 \quad (38)$$

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

$$x = x_{i_1} x_{i_2} \dots x_{i_m} = 2^{i_m-1} + \dots + 2^{i_2-1} + 2^{i_1-1} \quad (39)$$

Generuje on następujące przekształcenie: $p_m^{i_1 i_2 \dots i_m} = p_x = p_{2^{i_m-1} + \dots + 2^{i_2-1} + 2^{i_1-1}}$, 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:

$$FG = GF = I \Leftrightarrow G = F^{-1} \quad (40)$$

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 (39):

$$p(\theta) = (\theta_0; \theta_1^1, \dots, \theta_1^n; \theta_2^{12}, \dots, \theta_2^{n-1, n}; \theta_3^{123}, \dots, \theta_n^{12 \dots n}) \text{ we współrzędnych } \theta.$$

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

$$F[p(x)] = \theta_0 + \sum_i \theta_1^{i_1} x_{i_1} + \sum_i \theta_2^{i_1 i_2} x_{i_1} x_{i_2} + \sum_i \theta_3^{i_1 i_2 i_3} x_{i_1} x_{i_2} x_{i_3} + \dots + \theta_n^{12 \dots n} x_1 x_2 \dots x_n$$

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

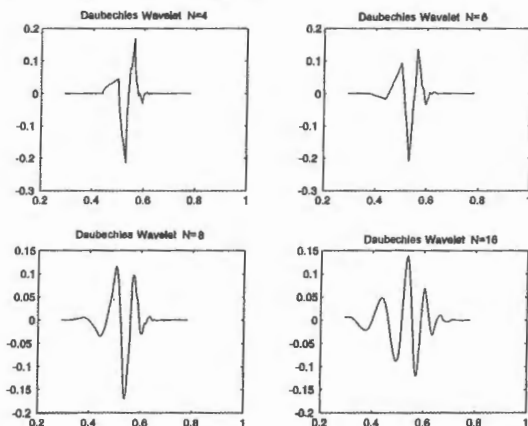
2.4.3.2. Transformacja falkowa

Jednowymiarowa funkcja falkowa określana wzorem [8]:

$$\psi_{su}(x) = \frac{1}{\sqrt{s}} \psi\left(\frac{x-u}{s}\right) \quad (41)$$

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

Przykład funkcji falkowej pokazano na rys. 22.



Rysunek 22 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 .

$$[W_\psi f](s, u) = \langle \psi_{su}, f \rangle = \frac{1}{\sqrt{s}} \int_{-\infty}^{+\infty} \psi\left(\frac{x-u}{s}\right) f(x) dx \quad (42)$$

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

$$\psi_{j,k}(x) = 2^{-j/2} \psi(2^{-j}x - k) \quad (43)$$

Transformacja falkowa w układzie falek diadycznych jest następująca:

$$[W_\psi f](k, j) = 2^{-j/2} \int_{-\infty}^{+\infty} \psi(2^{-j}x - k) f(x) dx \quad (44)$$

W przypadku obrazu stosuje się dwuwymiarową (2D) transformację falkową [21], 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: $\{\psi_{k,l}(x) \psi_{l,m}(y)\}_{(j,k,l,m) \in Z^4}$, 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 [8] można pokazać, że zaproponowany układ jest nadmiarowy i można go zredukować do:

$$\{\psi_{j,k,m}^1(x,y), \psi_{j,k,m}^2(x,y), \psi_{j,k,m}^3(x,y)\}_{(j,k,m) \in Z^3} \quad (45)$$

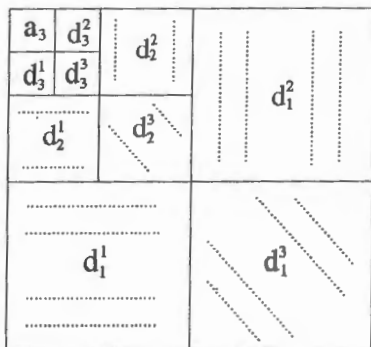
gdzie:

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

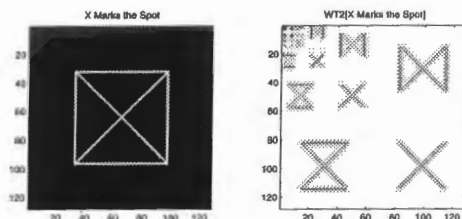
$$\psi^2(x,y) = \psi(x) \phi(y)$$

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

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. 23.



Rysunek 23 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 24 Przykładowy obraz (po lewej) i jego transformacja falkowa w układzie współczynników jak na rys. 21 (po prawej)

Przy indeksowaniu bazy danych można 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 [32] opisuje eksperyment w swojej bazie z falkami Haara i Daubechies dla kolorowych obrazów naturalnych.

2.5 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. [12] Zatem poza przeszukiwaniem bazy pod

kątem obrazów, musi istnieć możliwość wyszukiwania tekstowego w klasycznej jego formie. Można też wprowadzać metadane, które łączą w sobie informację o tekście i obrazie.

Taką strukturę zastosowano w przypadku bazy CHABOT zaprojektowanej na uniwersytecie w Berkeley [25]. 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 pytań 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 [10]. 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

2.5.1 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:

$$d = \sqrt{\sum_{i=0}^k (r_i - q_i)^2} \quad (46)$$

gdzie: r_i i q_i są to k -elementowe wektory, których i -ty element jest numerem koloru [29]. Do odpowiedzi system wybiera z bazy obrazy o minimalnej odległości d .

W trochę bardziej zaawansowanym porównywaniu [3] 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 [14]:

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

W większości przypadków użytkownicy są zainteresowani zapytaniem o indywidualne obiekty, które trzeba wyselekcjonować z obrazu (patrz rozdz. 2.2) 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, obwód, powierzchnia, itp. (patrz rozdz. 2.3)) ma określoną wartość liczbową, którą oblicza się wcześniej i przechowuje w bazie. Rozpatrujemy więc N -wymiarową przestrzeń cech $F_1 \times \dots \times F_i \times \dots \times F_N \in \mathbb{R}^n$, w której mamy znaleźć podobieństwo obiektów O_i i O_j . Określamy metrykę ϱ jako funkcję $\varrho: O_i \times O_j \rightarrow \mathbb{R}^+$ spełniającą następujące warunki:

1. $\varrho(O_i, O_i) = 0$
2. $\varrho(O_i, O_j) = \varrho(O_j, O_i)$
3. $\varrho(O_i, O_k) \leq \varrho(O_i, O_j) + \varrho(O_j, O_k)$

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:

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

gdzie: $r \geq 1$, a x_i i y_i są to k -elementowe wektory. Oczywiście dla $r = 2$ uzyskujemy odległość euklidesową [17].

Kolejną miarą podobieństwa opartą na wektorach cech jest cosinus wektorów:

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

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

2.5.2 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 [26].

Definicja 3

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

$$d(A, B) = \sqrt[r]{\sum_{i=0}^k [\mu_A(x_i) - \mu_B(x_i)]^r} \quad \text{i} \quad d_{\infty}(A, B) = \sup_i |\mu_A(x_i) - \mu_B(x_i)| \quad (51)$$

gdzie: $r \geq 1$, a $\mu_C(\cdot)$ 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:

$$\text{Sim}(A,B) = \frac{\max_{u \in U}(A \cap B)}{\max_{u \in U}(A \cup B)} \quad (51)$$

W przypadku obrazów obiecujący jest model Czerskiego [7], 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:

$$S(a,b) = \theta f(A \cap B) - \alpha f(A - B) - \beta f(B - A) \quad (52)$$

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

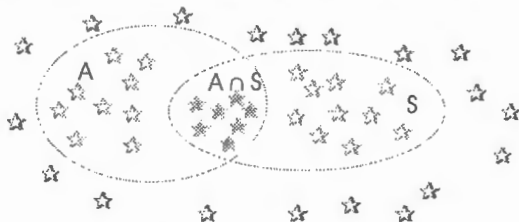
Można też posłużyć się modelem znormalizowanym do przedziału $[0,1]$:

$$S(a,b) = \frac{f(A \cap B)}{f(A \cap B) + \alpha f(A - B) + \beta f(B - A)} \quad (53)$$

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

2.5.3 Miary oparte na łącznym rozkładzie prawdopodobieństwa

Do poszukiwania podobieństwa można też stosować podejście probabilistyczne zaproponowane przez Jaccarda [16]. Przyjmujemy, że mamy obiekty na obrazie posiadające oddzielnie cechy A , S oraz obie cechy A i S na raz tak jak to pokazuje rys. 25.



Rysunek 25 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:

$$Sim(A, S) = \frac{P(A \cap S)}{P(A \cup S)} \quad (54)$$

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.

3. Instrukcja obsługi użytkownika

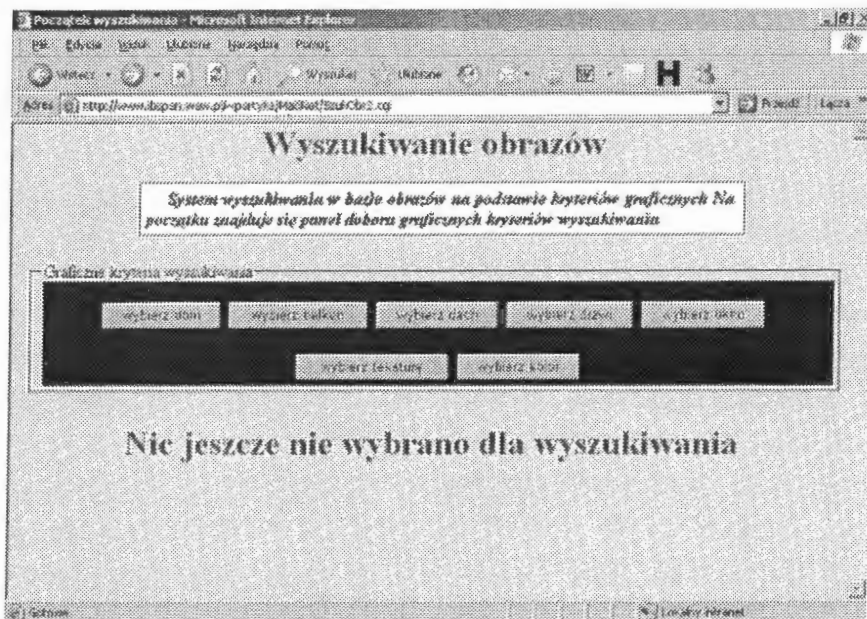
Obrazowa baza danych jest dostępna na instytucyjnej stronie internetowej. Dla rozpoczęcia pracy należy wpisać w polu adresu wyszukiwarki adres:

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

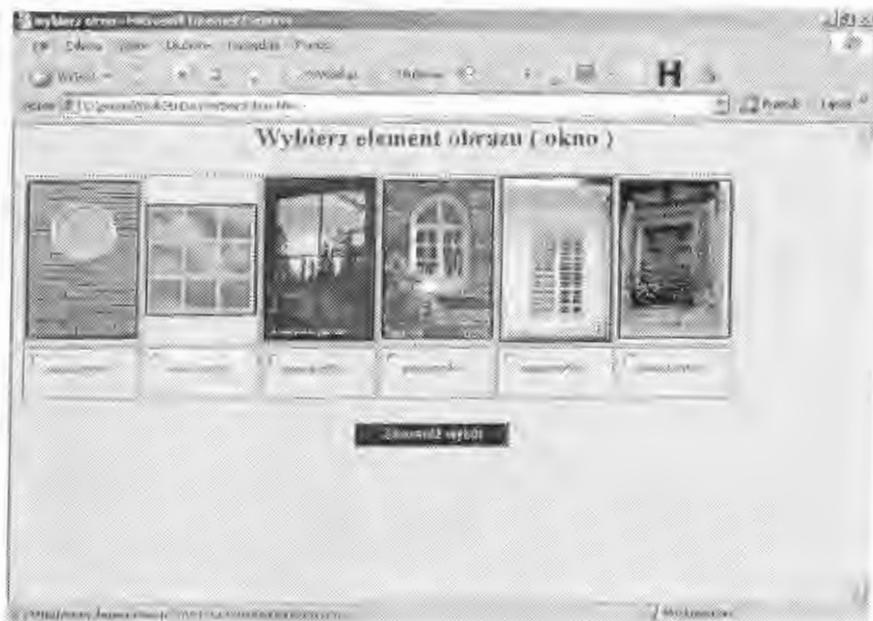
Jeżeli użytkownik chce ominąć korzystanie z apletu Javy przy ustalaniu koloru, powinien wpisać:

<http://www.ibspan.waw.pl/~partyka/MatI/Net/SzukObr2.cgi/BezJavy>

Powinien wtedy ukazać się następujący ekran:



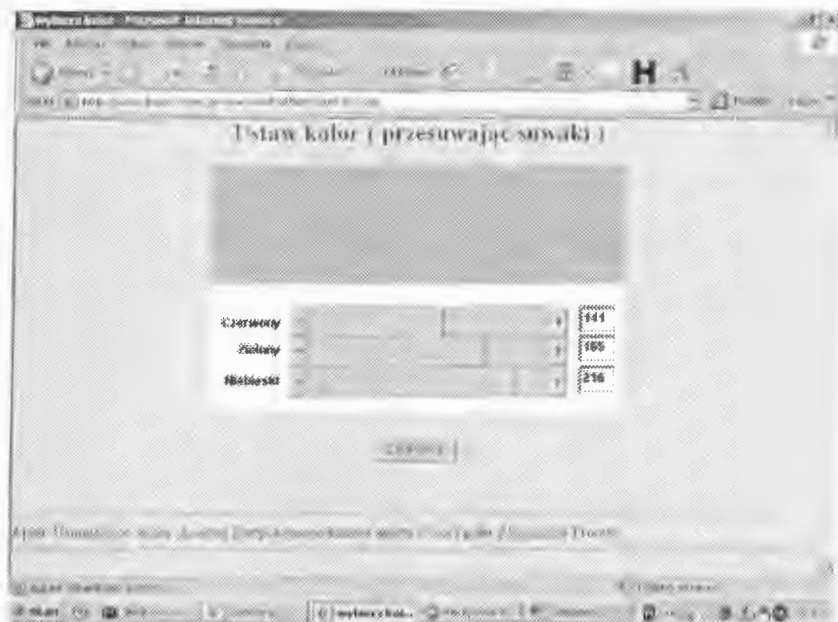
Dla wybrania określonego elementu graficznego należy kliknąć na odpowiedni przycisk. Jeżeli nie jest to przycisk „wybierz kolor, to pokaże się ekran z obrazami, z których należy wybrać jeden. Wygląd takiego ekranu przedstawiony jest poniżej



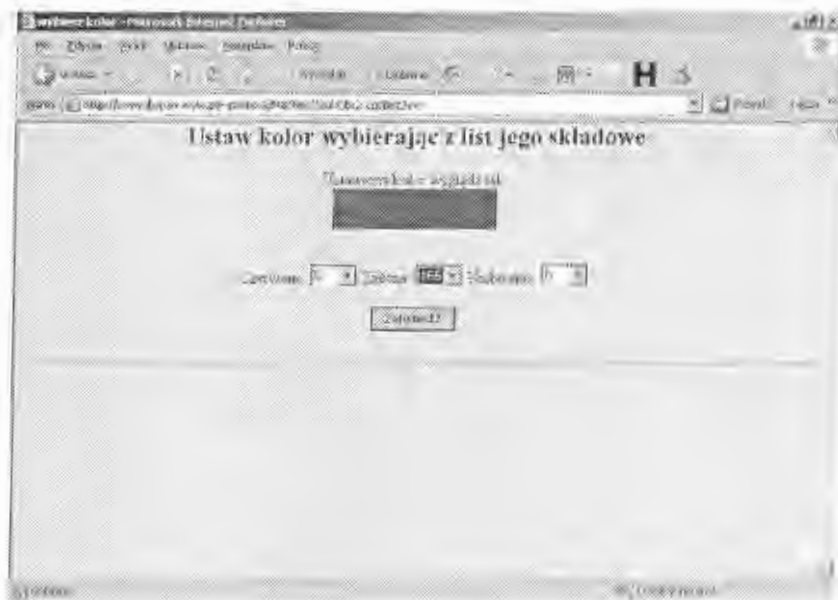
Dla wybrania obrazu należy zaznaczyć odpowiedni przycisk radiowy pod obrazem, a następnie zatwierdzić to przez kliknięcie przycisku „zatwierdź wybór”. Jeżeli użytkownik chce dokładniej obejrzeć dany obraz należy kliknąć na niego dwa razy i wtedy obraz pokaże się w oryginalnej wielkości na oddzielnym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

4. Dokumentacja techniczna programu

4.1 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/MatINet. 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 *tekstur*

DomyImg

dla obrazów *domów*

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

4.2 Sposób wybierania koloru

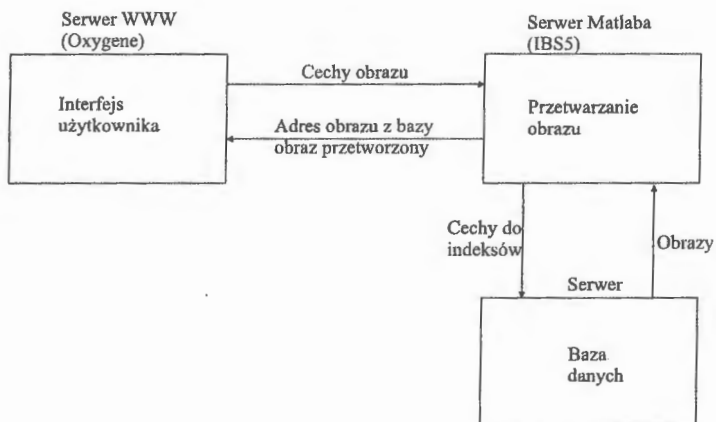
Domyślnym zadawaniem koloru steruje aplet Javy, zawarty w plikach z rozszerzeniem *.class*. Użytkownik ustawia trzema suwakami trzy składowe koloru (czerwoną, zieloną i niebieską) przy czym natychmiast widzi skutki swoich decyzji. Wymaga to jednak, żeby na komputerze użytkownika mogły być realizowane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

4.3 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 26.



Rysunek 26 Ideowy schemat konfiguracji pomiędzy serwerami obsługującymi obrazową bazę danych

5. Podsumowanie

Przedstawione opracowanie w szkicowy 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] Basir O., Zhu H., Karray F.: *Fuzzy Based Image Segmentation*, w *Fuzzy filters for Image Processing*, Springer, Berlin, 2003, str. 101-105
- [3] 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
- [4] Caselles V., Kimmel R., Sapiro G.: *Geodesic Active Contours*, International Journal of Computer Vision Vol. 22, Nr.1, 1997, str. 61-79
- [5] Castleman Kenneth R.: *Digital Image Processing*, Prentice Hall, 1996
- [6] Cortes C. i V. Vapnik. *Support vector networks*. Machine Learning, Nr. 20, 1995, str. 273-297
- [7] Czernski A. *Features of similarity*. Psychological Review, Vol 84, czerwiec 1977, str. 327-352
- [8] Daubachies I.: *Ten lectures on wavelets*, Society for Industrial and Applied Math., Filadelfia, 1992
- [9] 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
- [10] 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
- [11] Grosky W., Mehrotra R.: *Special Issue on Image Database Management*, Computer, Vol. 22 Nr. 12, grudzień 1989
- [12] Gudivada Venkar N.: *Content-Base Image Retrieval Systems*, IEEE Computer, Vol. 28, Nr. 9, sierpień 1995, str. 18-22.
- [13] Guttman A., *R-trees: a Dynamic Index Structure for Spatial Searching*, Proceedings ACM SIGMOD Conference, Boston, 1984, str. 47-57
- [14] Hafner J., i in.: *Efficient Colour Histogram Indexing for Quadratic Form Distance Functions*, IEEE PAMI Vol. 17, Nr. 7, lipiec 1995, str. 729-736
- [15] Jain R.: *NSF Workshop on Visual Information Management Systems*, SIGmod Record, Vol. 22, Nr. 3, grudzień 1993, str 57-75

- [16] Jaccard P.: *Nouvelles recherches sur la distribution florale*, Bulletin de la Societe Vaudoise des Sciences Naturelles, Vol. 44, str.223–270, 1908.
- [17] Jain R., Murthy J., Chen P.: *Similarity Measures for Image Databases*, SPIE, Vol. 2420, str. 58-62
- [18] Julesz B.: *Inability of Humans to discriminate Between Visual Texture. That Agree in Second-order Statistics – Revisited.*, Perception Nr. 2, 1973 str. 391-405
- [19] 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
- [20] Liu Y.: *θ -index of Imagebase*, SPIE, Vol. 2420, 1995, str. 68-81
- [21] Mallat S.: *A wavelet tour of signal processing*, Academic Press, Londyn, 1998
- [22] Manolopoulos Y. I in.: *R-trees Have Grown Everywhere*, ACM Computer Surveys, 2003, str. 1-62
- [23] Melin P., Castillo O.: *Hybrid Intelligent Systems for pattern Recognition Using Soft Computing*, Springer, Berlin, 2005, str. 169-184.
- [24] Munsell A. H.: *A Grammar of Color*, Van Nostrand-Reinhold, NY, 1969
- [25] Ogle V., Stonebraker M.: *CHABOT: Retrieval from a Relational Database of Images*, IEEE Computer, Vol. 28, Nr. 9, sierpień 1995, str. 40-48
- [26] 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, Wroclwaw, wrzesień 2001, str. 264-273
- [27] Pratt W. K.: *Digital Image Processing*, Wiley, 1991, str. 276-284
- [28] 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
- [29] Stricker M., Orengo M.: *Similarity of Colour Imagas*, SPIE Vol. 2420, str.381-392
- [30] 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
- [31] Wald G.: *The receptors for Human Color Vision*, Science, Nr. 145, 1964, str. 1007-1017

- [32] 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
- [33] Wang L.: *Support Vector Machines: Theory and Applications*, Springer, Berlin, 2005
- [34] Xu Ch., Prince J.: *Snakes, Shapes, and Gradient Vector Flow*, IEEE Transactions on Image Processing, Vol. 7, Nr. 3 Marzec 1998, str. 359-369

7. Spis ilustracji

Rysunek 1	Ogólny schemat obrazowej bazy danych	8
Rysunek 2	Zmiana jasności obrazów przez „przesunięcie” histogramów	10
Rysunek 3	Przykład rysunku przed (po lewej) i po zmianie histogramu (po prawej). Wartości kolorów zostały obliczone według wzoru (4), a następnie uśredniono każde trzy sąsiednie poziomy do środkowego	10
Rysunek 4	Przykład obrazu o bimodalnym histogramie. Ciemne tło tworzy jeden pik na takim histogramie, a jasne komórki drugi	11
Rysunek 5	Bimodalny histogram z zaznaczonym progiem T . Punkty o mniejszym poziomie szarości uznajemy jako tło, a o większym jako punkty należące do poszukiwanych obiektów	12
Rysunek 6	Wykres rodzaju krawędzi (u góry), pierwszej pochodnej (na środku) i drugiej pochodnej (u dołu)	13
Rysunek 7	Przykłady skupień z zaznaczonymi centroidami „+” w przypadku dwuwymiarowym	15
Rysunek 8	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 docelowo chcemy uzyskać	15
Rysunek 9	Standardowa S funkcja	19
Rysunek 10	Segmentacja obrazu oparta na funkcji przynależności $\mu_X(x_{im})$ typu S	20
Rysunek 11	Przykładowe skupienia z zaznaczonymi centroidami i zaznaczonymi funkcjami przynależności dla punktów najbliższych tym centroidom w danych skupieniach	22
Rysunek 12	Efekt segmentacji algorytmem rozmytych c -środków dla $c = 6$ z nałożonymi krawędziami dla lepszego podkreślenia segmentów. Każdy poziom szarości reprezentuje jeden segment	23
Rysunek 13	Przykład konturów dla obiektu niejednorodnego topologicznie	24
Rysunek 14	Przykład rozkładu gradientów sił na aktywnym konturze	25
Rysunek 15	Przestrzeń kolorów RGB (traktowana jako trójwymiarowy histogram)	26
Rysunek 16	Przykładowe tekstury naturalne a) tkanina, b) kamień, c) marmur i d) korek	

Rysunek 17	Schemat rozpoznawania tekstur przy użyciu transformacji Fouriera i algorytmów genetycznych	27
Rysunek 18	Reprezentacja kształtu w oparciu o "różę wiatrów"	28
Rysunek 19	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.....	29
Rysunek 20	Struktura indeksu obrazowej bazy danych. Szare pola na poziomie cech pokazują obszar poszukiwań, a ciemne pola to wyszukane obrazy.	30
Rysunek 21	Przykład struktury R-drzewa i jego reprezentacja przestrzenna (po lewo).	31
Rysunek 22	Przykład falek Daubechies	33
Rysunek 23	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).....	34
Rysunek 24	Przykładowy obraz (po lewej) i jego transformacja falkowa w układzie współczynników jak na rys. 21 (po prawej)	35
Rysunek 25	Obiekty w obrazie mogą mieć cechy A (zielone), cechy S (pomarańczowe) lub obie te cechy na raz (pomarańczowe z zieloną ramką)	39
Rysunek 26	Ideowy schemat konfiguracji pomiędzy serwerami obsługującymi obrazową bazę danych.....	47

