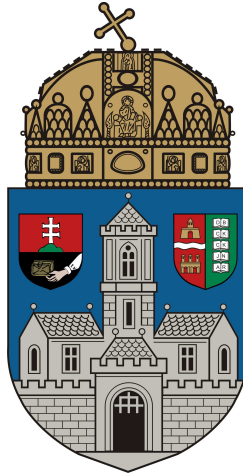


Óbudai Egyetem

Doktori (PhD) értekezés



Tartalom alapú keresési algoritmusok képi adatbázisokban

Sergyán Szabolcs

Témavezetők:
Szeidl László DSc
Rövid András PhD

Alkalmazott Informatikai Doktori Iskola

Budapest, 2011. július 15.

Tartalomjegyzék

1. Bevezetés	4
2. Tartalom alapú keresés áttekintése	6
2.1. Alacsony szintű képi jellemzők	9
2.1.1. Szín	10
2.1.2. Textúra	18
2.1.3. Alakzat	22
2.2. Tesztelési eljárások	23
3. Automatikus paraméterezés	25
3.1. Szegmentáló eljárások automatikus paraméterezése	26
3.1.1. Algoritmus leírása	26
3.1.2. Tesztelés	29
3.1.3. Eredmények	30
3.2. Élkereső algoritmusok automatikus paraméterezése	33
3.2.1. Az alkalmazott algoritmus	33
3.2.2. Tesztelés	35
3.3. Konklúziók	36
4. HOSVD alapú eljárások	37
4.1. HOSVD áttekintés	38
4.2. Digitális képek HOSVD alapú reprezentációja	40
4.2.1. Megjegyzések a kanonikus alakhoz	42

4.3.	A Fourier transzformáció és a HOSVD kapcsolata	44
4.3.1.	Példák a HOSVD eljárások használatára	45
5.	Távolsági- és hasonlósági mértékek	51
5.1.	Irodalmi áttekintés	51
5.2.	Újfajta távolság értelmezése	55
5.3.	Kísérletek	57
6.	Skicc alapú kereső rendszer	61
6.1.	Kifejlesztett rendszer	62
6.1.1.	A rendszer célja	62
6.1.2.	Rendszerünk általános felépítése	62
6.1.3.	Az előfeldolgozó alrendszer	64
6.1.4.	A jellemző vektor előállító alrendszer	65
6.1.5.	Visszakereső alrendszer	74
6.1.6.	Az adatbáziskezelő alrendszer	74
6.1.7.	A megjelenítő alrendszer	75
6.2.	Tesztelés	76
6.2.1.	Tesztkörnyezet	76
6.2.2.	Tesztelési szempontok	79
6.2.3.	Teszt eredmények az EHD leíró használatakor	79
6.2.4.	Teszt eredmények a HOG leíró használatakor	82
6.2.5.	Összehasonlítás más rendszerrel	84
6.3.	Többszintű visszakeresés SIFT leíró használatával	85
6.4.	Összegzés	87
7.	Összegzés (Tézisek)	91
7.1.	Az eredmények hasznosítása, továbbfejlesztési lehetőségek	94

Köszönetnyilvánítás

Ezúton mondok köszönetet témavezetőimnek, Prof. Dr. Szeidl Lászlónak és Dr. Rövid Andrásnak eredményeim elérésében, értekezésem elkészítésében nyújtott magas színvonalú, folyamatos és áldozatos segítségért.

Köszönöm az Alkalmazott Informatikai Doktori Iskola tagjainak, különösen vezetőjének, Prof. Dr. Galántai Aurélnak a hasznos szakmai tanácsokat és a kollegális segítséget.

Köszönöm a munkahelyi vitára készített alapos, sok hasznos észrevételt tartalmazó bírálatát Dr. Hermann Gyulának és Dr. Seebauer Mártának.

Az Óbudai Egyetemen dolgozó közvetlen kollégáimnak - különösen Dr. Tick Józsefnek, Dr. Vámosy Zoltánnak és Dr. Csink Lászlónak - köszönöm a támogatást, a kitartó ösztönzést értekezésem elkészítése során.

Köszönöm családomnak a megértést, biztatást és az áldozatvállalást, amivel az értekezésem elkészítését lehetővé tették.

1. fejezet

Bevezetés

A képi adatbázisokban tárolt képek mennyisége rohamosan növekszik elsősorban az internet, illetve a digitális képrögzítő eszközök széleskörű elterjedése folytán.

Képi adatbázisok felépülhetnek egyes szakterületek képeiből (pl. orvosi képek, műalkotások, csillagászati felvételek), vagy akár a képek tág halmazából, ami lehet az interneten fellelhető képek egy részhalmaza is.

A kutatókat régóta foglalkoztatja a valamilyen szempontból hasonló képek keresése egy adatbázisban. A keresésnél lényeges különbséget jelent, hogy az adatbázist milyen módon indexelték. Egyes adatbázisokban a képekhez szöveges indexeket, jellemző szöveges leírókat adnak, majd ezek alapján történik a keresés (pl. a Google képkeresője). A szövegesen indexelt adatbázisok esetén nagy mennyiségű munkát jelent az indexek elkészítése, illetve új tulajdonság bevezetése esetén az adatbázis újraindexelése. Manuális indexelés esetén a hibák előfordulása is jelentős számú lehet.

A másik indexelési módszer a tartalom szerinti indexelés, ahol a számítógép nyer ki információkat a kép pixeleiben tárolt információból valamilyen algoritmus alapján. Gyakran használt indexek a kép színjellemzői (pl. színmomentumok, színhisztogramok, szín koherencia vektor, stb.), mintázata (pl. statisztikai leírók, co-occurrence mátrix, stb.), a képen található objektumok száma, elhelyezkedése, alakja. Az indexek sora természetesen jelentősen bővíthető. Az indexek előállítására jelentős időt igényel, viszont ez jóval kevesebb a szöveges indexek begépelésénél, illetve új index meghatározása is gyorsabban megvalósítható. Tartalom szerint indexelt adatbázisban a keresés a számítógép által előállított indexek alapján történik valamilyen távolság-, vagy hasonlósági mérték használatával.

Értekezésem témája tartalom alapú keresés képi adatbázisokban, valamint ehhez kapcsolódó képfeldolgozási módszerek áttekintése.

Az 2. fejezetben ismertetem a tartalom alapú keresés alapfeladatát, illetve áttekintést nyújtok a gyakran használt megközelítésekről, a megvalósított rendszerekről.

A tartalom alapú keresésnél is használt képfeldolgozási algoritmusok általában paramétereiktől függenek. Ezeket a paramétereket előre meg kell határozni, amely sokféle képi adatbázisban való keresésnél rontja a keresés hatékonyságát, mivel a paraméterek általában függenek a képek fajtájától. Az összefüggő homogén régiók előállításánál, illetve él keresésnél használható algoritmusok automatikus paraméterezésére új eljárást dolgoztam ki [20, 97, 98, 99, 95], melyet a 3. fejezetben ismertetek részletesen.

A HOSVD alapú technikák a képfeldolgozásban hatékonyan használhatók különös tekintettel a felbontás növelésére, szűrésekre és képtömörítésre. Megvizsgálom, hogy egy többváltozós függvény approximáció esetében a HOSVD, illetve a Fourier transzformációra épülő eljárás esetén a megtartott komponensek száma milyen módon befolyásolja az approximáció hibáját. A komponensek HOSVD esetén specifikusan meghatározott ortonormált függvények, melyek függenek az approximálandó többváltozós függvénytől. Ezzel szemben a Fourier transzformációnál közismert, hogy a komponensek trigonometrikus függvények. E jellegbeli különbségnek köszönhetően érhető el ugyanaz az eredmény kevesebb komponens szám alkalmazásával HOSVD esetén. Az eredmények ismeretében megállapítható, hogy a Fourier transzformáció esetében alkalmazott komponensek nagy mértékű elhagyása koncentrikus íveket eredményez a képen. HOSVD használata esetén viszont a kép élessége nem változik nagy mértékben, csak az egyes tartományok közötti összefüggőség mértéke csökken. [88]

A disszertáció 4. fejezetében áttekintést nyújtok a HOSVD alapú eljárások használhatóságáról képi adatbázisok indexelése esetén.

A 5. fejezetben bemutatom az indexek összehasonlítására gyakran használt távolsági- és hasonlósági mértékeket [91], elemzem ezek alkalmazhatóságát a képi információk függvényében. Egyes távolságmértékek továbbfejlesztését is bemutatom, melyek alkalmazhatóságát tesztek igazolják.

Kifejlesztettem egy olyan rendszert [115], mely skiccek alapján keres hasonló alakú objektumokat képi adatbázisban. A rendszer más hasonló rendszerek módszereit is felhasználja, de ezeket továbbfejlesztettem és az elvégzett kísérletek alapján kijelenthető, hogy hatékonyabb keresést sikerült megvalósítani módosított algoritmusommal, mint más hasonló rendszereknél. A rendszert részletesen az értekezés 6. fejezetében mutatom be.

Az értekezés 7. fejezetében összefoglalom az elért új tudományos eredményeimet.

2. fejezet

Tartalom alapú keresés áttekintése

Az internet használatának gyors növekedésével, valamint a digitalizálás és tároló eszközök árának csökkenésével egyre népszerűbbé vált szövegek, képek, grafikák, hanganyagok digitális formában való készítése és tárolása. Ez növelte annak igényét is, hogy az eltárolt tartalmak között hatékony keresést lehessen végrehajtani. Ennek az általános problémának egy része a képi anyagok tárolásának és közöttük való keresésnek a megvalósítása.

Képi adatbázisokban történő keresésre alapvetően két különböző módszert használnak: a szöveg alapú, illetve a tartalom alapú megközelítést. A szövegesen indexelt rendszerek fejlesztése már az 1970-es években megkezdődött. Ezekben a rendszerekben a képekhez manuálisan rendelnek hozzá szöveges leírókat, amelyek ezt követően az adatbázisban történő keresés alapjául szolgálnak. Ennek a módszernek két hátránya van. Az első, hogy jelentős mértékű emberi munkát kíván a szöveges indexelés megvalósítása. A második pedig, hogy a szöveges indexek pontossága az emberi érzékelés szubjektivitásától függ [23, 100]. A képek közötti keresések egyik fontos feladata, hogy a képek tartalma alapján szemantikai jellemzőket nyerjünk ki, melyek alapján a felhasználó igényeihez illeszkedő találatokat kapunk egy keresés során. A szöveges alapú keresés hátrányainak kiküszöbölése érdekében indult el az 1980-as évek elején a tartalom alapú kereső rendszerek (CBIR - Content-based Image Retrieval) rohamos, máig is tartó fejlődése [14, 102, 60, 48].

A tartalom alapú kereső rendszerekben viszont a képeket saját vizuális tartalmuk alapján indexelik. A leggyakrabban figyelembe vett jellemzők a szín, a textúra, illetve az alak.

A tartalom alapú képkereséssel kapcsolatban első meghatározó cikket Chang és Liu publikálta 1984-ben [13], amelyben a szerzők bemutattak egy kép indexelési és absztrakciós eljárást képeslap adatbázisban való keresésre.

Az évek során számos tartalom alapú kereső rendszert fejlesztettek ki. A legjelentősebb és

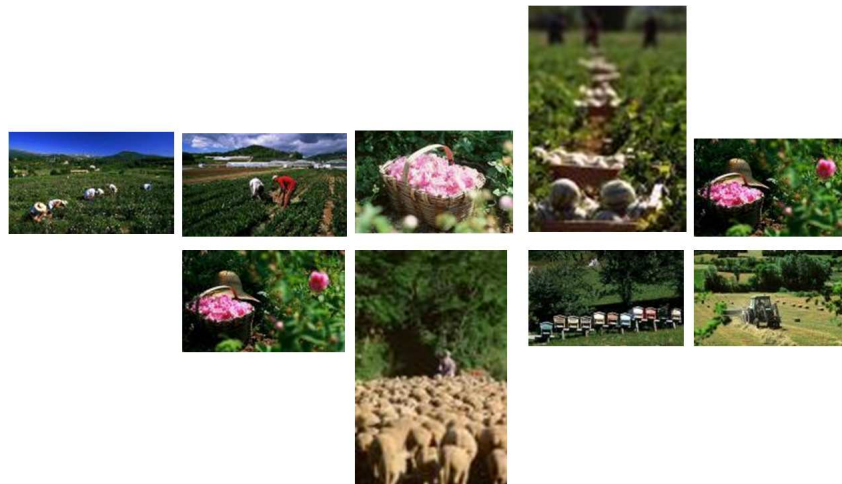
legismertebb rendszerek ezek közül:

- IBM QBIC [27]
- MIT Photobook [78]
- Berkeley Chabot [74]
- Blobworld [9]
- Virage [37]
- Columbia VisualSEEK and WebSEEK [103]
- PicHunter [19]
- UCSB NeTra [65]
- UIUC MARS [67]
- PicToSeek [34]
- Stanford WBIIS [127]
- SIMPLIcity [126]

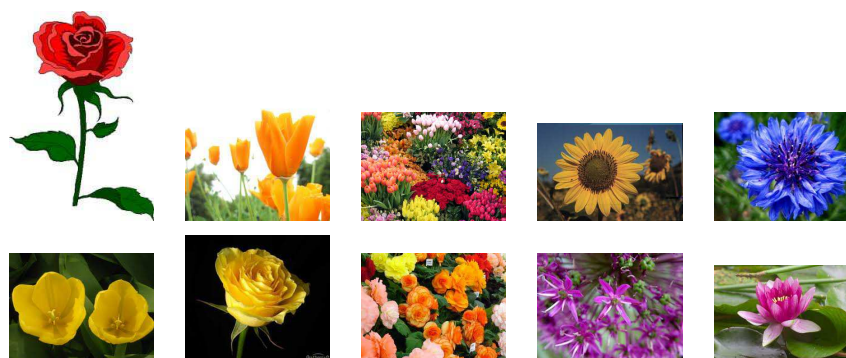
További rendszerek részletes bemutatása és összehasonlító elemzése megtalálható a [124, 87] cikkekben. A rendszerek részletes elemzése nem része a disszertációnak.

Az alapvető különbség a tartalom alapú és a szöveges visszakereső rendszerek között az, hogy az utóbbinak elhagyhatatlan része az emberi beavatkozás szükségessége. Az emberek viszont hajlamosak arra, hogy magas szintű jellemzőket, például fogalmakat használjanak kulcsszóként, szöveges leíróként. Ezzel szemben a számítógép által automatikusan előállított jellemzők a gépi látás és képfeldolgozás területén alkalmazott eljárások használatával állítanak elő főként alacsony szintű jellemzőket. Ilyen alacsony szintű jellemzőnek tekinthetjük a színt, textúrát, alakot, térbeli elhelyezkedést, stb. Általában viszont nincs közvetlen kapcsolat az alacsony és magas szintű jellemzők között [100]. Bár számos kifinomult algoritmust fejlesztettek már ki a szín, alak és textúra jellemzők leírására, ezek az eljárások nem képesek pontosan modellezni a képeken található szemantikai információkat, és így számos korlátja van annak, hogy olyan tartalom alapú kereső rendszer jöjjön létre, amely a felhasználók igényeit teljes mértékben ki tudja elégíteni széles spektrumú képi adatbázisokban történő keresés esetén [70]. A CBIR-rendszerekben végzett alapos kísérletek azt igazolták, hogy az alacsony szintű leírók

sok esetben nem alkalmasak az emberi agyban kialakuló magas szintű szemantikai fogalmak leírására [136]. Emiatt a tartalom alapú kereső rendszerek teljesítménye még mindig messze elmarad a felhasználók elvárásaitól.



2.1. ábra. Első szintű lekérdezés eredménye.



2.2. ábra. Második szintű lekérdezés eredménye.

A lekérdezéseknek három szintjét különböztetjük meg a tartalom alapú képkereső rendszerekben [23].

- 1. szint:** A kép primitív jellemzői (szín, textúra, alak, térbeli elhelyezkedés, stb.) alapján történő keresés. Tipikus esete a példa kép alapján történő keresés: „keress ehhez hasonlót”. A 2.1. ábrán egy szín alapján végrehajtott lekérdezés eredménye látható, ahol a bal felső képhez leginkább hasonló találatokat jelenítettem meg.
- 2. szint:** A képen található objektumok jellemzőiből logikai következtetések alapján kinyert adott típusú azonosítók alapján történő keresés. Például „találd meg egy virág képét” (lásd a 2.2. ábrát).



2.3. ábra. Harmadik szintű lekérdezés eredménye.

3. szint: Absztrakt attribútumok alapján történő keresés, beleértve magas szintű leírót a képnek, amelyek a kép készítésének körülményeiről árulnak el információt. Például „találj egy képet, amin örömteli tömeg látható” (lásd a 2.3. ábrát).

A 2. és 3. szintet együttesen szemantikai kép visszakeresésnek nevezzük, az első és második szint közötti hézag pedig az ún. szemantikai hézag [15, 102].

Hogyan lehet csökkenteni a szemantikai hézag mértékét, azaz hogyan nyerhetünk ki az alacsony szintű képi jellemzőkből magas szintű szemantikai jellemzőket? Jelenleg a szemantikai hézag csökkentésére főként ötféle különböző módszert alkalmaznak [60], melyeket az alábbiakban ismertetek. Mindegyik típushoz megadtam a meghatározó irodalmi hivatkozásokat, de mivel disszertációmban csak a releváns visszacsatolás egy lehetőségét mutatom be a 6. fejezetben, részletesen nem fejtem ki az egyes esetek hátterét.

- I. Ontológiák használata magas szintű fogalmak definiálására [69, 79, 108, 59, 82, 53, 17]
- II. A gépi tanulás eszközeinek használata [100, 119, 122, 64, 12, 134, 118, 29, 106, 15, 135, 31, 57, 58, 28]
- III. Releváns visszacsatolási technikák alkalmazása ciklikus lekérdezőként a folyamatos felhasználó által vezérelt tanulás érdekében [86, 84, 137, 47, 36, 85]
- IV. Szemantikus minták előállítás a magas szintű lekérdező támogatására [104, 138, 16]
- V. A kép látható jellemzőinek, valamint a képhez hozzárendelhető szöveges információknak az együttes használata [6, 30, 7]

2.1. Alacsony szintű képi jellemzők

A jellemzők (tartalom) kinyerése az alapja a CBIR rendszereknek [87]. A jellemzők két nagy csoportra oszthatók. Az elsőbe tartoznak az általános jellemzők, mint a szín, a textúra, illet-

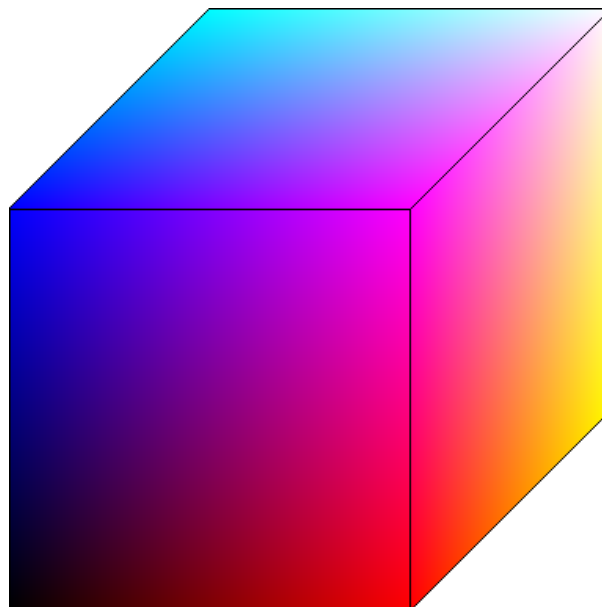
ve az alak, az utóbbiba pedig az alkalmazás függő leírók, például az emberi arcok [96, 93], ujjlenyomatok. A tartalom alapú kereső rendszerek általános áttekintésekor csak az általános jellemzőkre kívánok kitérni.

2.1.1. Szín

A szín jellemző egyike a kép visszakeresésben leggyakrabban használt vizuális jellemzőknek. Ez a jellemző viszonylagosan robusztus a háttér bonyolultságával szemben, valamint független a kép méretétől és irányától. Az alábbiakban áttekintem a gyakran használt színtereket, valamint a szín leírására használt jellemző vektorokat [92, 128, 61].

Színterek

Számos különböző színt és az azokhoz tartozó színteret használnak a gyakorlatban, és ezeket a színtereket egymásba lehet transzformálni. Amennyiben tökéletes reprezentációt adó színteret használunk, akkor ez a transzformáció kölcsönösen egyértelmű leképezés, így nem történik információ veszteség e során (kivéve a kerekítésből adódó hibákat). Mivel minden egyes színtérnek megvan a saját színskálája, így a színinformáció sérülhet a transzformáció során, amennyiben a szín ezen skálán kívülre esik [105].



2.4. ábra. Az RGB színekocka.

Az RGB színtér a katódsugárcsöves televízió készülékeknél került bevezetésre. Az RGB

színtér egy példája a nem abszolút színtereknek, mivel nem lehet minden színt ábrázolni benne. A színtér alapszínei a vörös (R - Red), a zöld (G - Green) és a kék (B - Blue). Az RGB model ún. additív színeket használ, melyek a három alapszín keveréséből jönnek létre. Az RGB színek kockát a 2.4. ábra szemlélteti. Egy meghatározott szín a színtérben három komponensű (r, g, b) vektorként ábrázolható, melynek elemei az egyes alapszínekhez rendelt (r, g, b) súlyok:

$$r \cdot R + g \cdot G + b \cdot B. \quad (2.1)$$

A színterek közötti transzformációt általában egy 3×3 -as mátrix szorzással lehet megvalósítani.

A megvilágítás intenzitásának mértékétől való függőség csökkentése érdekében a háromdimenziós RGB színtérből áttérhetünk egy kétdimenziós rgb színtérbe, ahol $r + g + b = 1$. Az egyes komponensek definíciója:

$$r = \frac{R}{R+G+B}, \quad g = \frac{G}{R+G+B}, \quad b = \frac{B}{R+G+B}. \quad (2.2)$$

Az RGB színtérből az rgb színtérbe történő transzformáció az egyik legegyszerűbb szín-normalizációs eljárás, melynek előnye a kevés számítási igény. Az RGB színtéren belüli hatékonyan használható színnormalizációs eljárás a szín klaszter forgatás [77].

Az XYZ színtér a színes filmek kifejlesztésekor, 1931-ben került bevezetésre a CIE (International Commission on Illumination) által. Az XYZ standard három elképzelt fényen alapul, ahol $X = 700,0\text{nm}$, $Y = 546,1\text{nm}$ és $Z = 35,8\text{nm}$ és ezek $X(\lambda)$, $Y(\lambda)$ és $Z(\lambda)$ szín illeszkedési függvényeiből. Ha az adott fényeket el szeretnénk képzelni, akkor mondhatjuk, hogy az X a vörös, az Y a zöld és a Z a kék színű fényt közelíti. Az CIE standard egy abszolút színteret eredményez, amelyben minden létező szín leírható.

Az XYZ színtér három követelményt elégít ki:

- A szín illeszkedési függvények nemnegatív értékűek
- Az $Y(\lambda)$ értéke megegyezik a fénysűrűség értékével
- Az egyes színilleszkedési függvények alatti területek megegyeznek

Minden egyes szín az alábbi lineáris kombinációként állítható elő:

$$c_X X + c_Y Y + c_Z Z, \quad (2.3)$$

ahol c_X, c_Y, c_Z az adott színt reprezentáló súlyok (intenzitások).

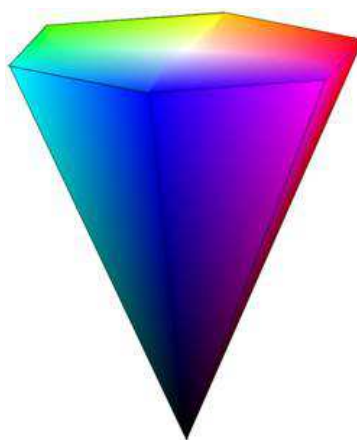
Az RGB és az XYZ színtér közötti transzformáció az alábbi módon valósítható meg:

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 3,24 & -1,54 & -0,50 \\ -0,98 & 1,88 & 0,04 \\ 0,06 & -0,20 & 1,06 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad (2.4)$$

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0,41 & 0,36 & 0,18 \\ 0,21 & 0,72 & 0,07 \\ 0,02 & 0,12 & 0,95 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (2.5)$$

Az YCbCr színtér egyik tagja a video rendszerekben használt színtereknek. Y értéke a színsűrűséget adja meg, míg Cb és Cr a kék és vörös kromaticitás komponensek. Az RGB színtérből az YCbCr színtérbe az alábbi módon térhetünk át:

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \frac{1}{255} \cdot \left(\begin{bmatrix} 65,481 & 128,553 & 25,966 \\ -37,797 & -74,203 & 112 \\ 112 & -93,786 & -18,214 \end{bmatrix} \cdot \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} \right) \quad (2.6)$$



2.5. ábra. A HSV színtér szemléltetése.

A HSV (Hue, Saturation, Value) színtér gyakran használt a festőművészek által, mivel nagyon közel áll az ő gondolkodási módjukhoz és technikájukhoz. Ha egy új színt kell képezni, akkor azt egy művész kikeveri a már meglévőkből, mint például a lilát vagy a narancssárgát. A HSV színtér egy hatszögletű gúlaként ábrázolható, ahol a felső hatszög az RGB kocka kétdimenziós vetülete (lásd a 2.5. ábrát). A HSV színtérbe az alábbi módon tudunk transzformálni

az RGB színtérből [132]:

$$H = \arctan \frac{\sqrt{3}(G-B)}{(R-G)+(R-B)} \quad (2.7)$$

$$S = 1 - \frac{\min\{R, G, B\}}{V} \quad (2.8)$$

$$V = \frac{(R+G+B)}{3} \quad (2.9)$$

$$(2.10)$$

A HSV színtérben az intenzitás információ elválik a szín információtól, így a szín (hue) és a telítettség (saturation) az emberi érzékelésnek megfelelő értékeket szolgáltat, ami nagyon hasznosnak bizonyult számos képfeldolgozási algoritmus esetén. A HSL színtér (vagy HSI) színtér nagyon hasonló a HSV színtérhez, csak a fényerő (lightness) helyettesíti a világosságot (brightness). A fő különbség, hogy a HSV színtérnél használt V érték egy tiszta szín esetén megegyezik a fehér szín világosság értékével, míg a HISL színtér esetén az 50%-os szürke szín világosságával.

Gyakran használt színtér még az úgynevezett opponens színtér, ahol az egyes színkomponensek az RGB színtér értékeiből állíthatók elő:

$$(R-G, 2B-R-G, R+G+B). \quad (2.11)$$

Ennek a színtérnek nagy előnye, hogy a világosság információt csak a harmadik komponens tartalmazza, az első két komponens pedig közel invariáns a megvilágításra nézve.

Az egyes színterek alkalmazhatóságának lehetőségei és előnyei a [92] műben találhatóak.

Szín leírók

Az alábbiakban bemutatom a leggyakrabban használt színleírókat: a szín momentumokat, a szín hisztogramot és a szín összefüggőségi vektort.

Az elsőrendű (közéérték), másodrendű (szórás) és harmadrendű (ferdeség) szín momentumok egyszerű, hatékony szín jellemzést tesznek lehetővé [109]. Ezek definíciója a megadott

sorrendben :

$$\mu_i = \frac{1}{M} \sum_{j=1}^M f_{ij}, \quad (2.12)$$

$$\sigma_i = \sqrt{\frac{1}{M} \sum_{j=1}^M (f_{ij} - \mu_i)^2}, \quad (2.13)$$

$$s_i = \sqrt[3]{\frac{1}{M} \sum_{j=1}^M (f_{ij} - \mu_i)^3}, \quad (2.14)$$

ahol f_{ij} a j indexű pixel intenzitás értéke az i -edik színcsatornában, M pedig a kép pixeleinek száma.

Az f és f' r darab színcsatornával ábrázolt kép momentumainak összehasonlítása az alábbi módon történhet [109]:

$$d_{\text{mom}}(f, f') = \sum_{i=1}^r (w_{i1} |\mu_i - \mu'_i| + w_{i2} |\sigma_i - \sigma'_i| + w_{i3} |s_i - s'_i|), \quad (2.15)$$

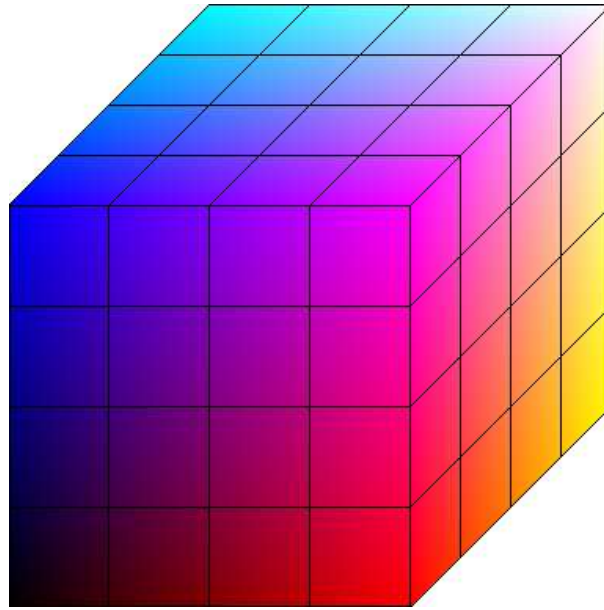
ahol w_{kl} ($k, l \in \{1, 2, 3\}$) felhasználó által definiált súlyok. Fontos megjegyezni, hogy d_{mom} nem egy metrika, lehetséges például, hogy két nem azonos színeloszlású kép esetén d_{mom} értéke 0. Emiatt d_{mom} -ot hasonlósági függvénynek nevezzük.

Egy szürkeárnyaltos kép hisztogramját úgy definiáljuk, hogy az egyes intenzitás értékek előfordulási gyakoriságát elosztjuk a kép pixeleinek számával:

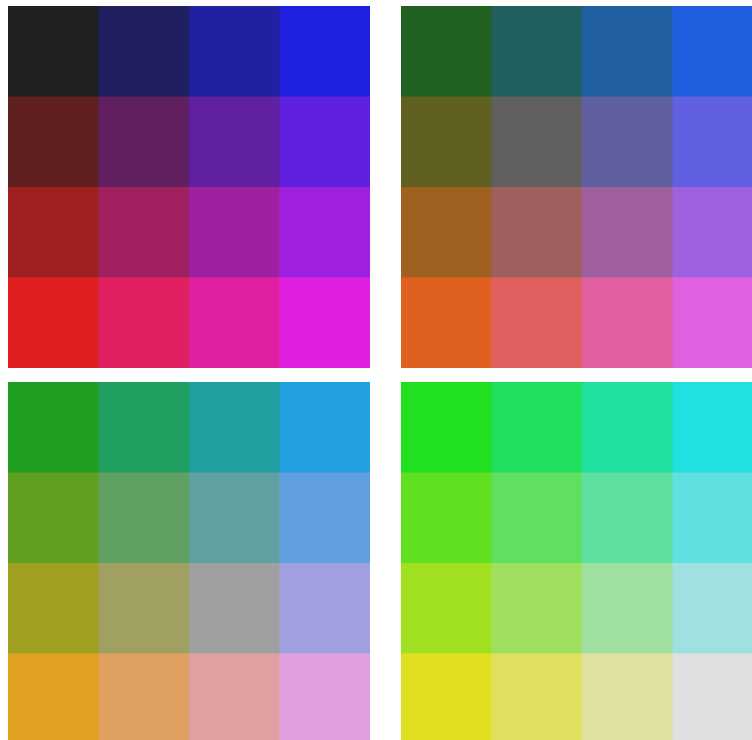
$$H(g) = \frac{N(g)}{M}, \quad (2.16)$$

ahol g jelöl egy adott szürkeárnyaltos intenzitás értéket, $N(g)$ pedig annak az előfordulási gyakorisága a képen [121, 94]. Színes képek esetén hasonló módon értelmezhető a színhisztogram, amelynek előállításakor az egyes színcsatornákat külön-külön vesszük figyelembe.

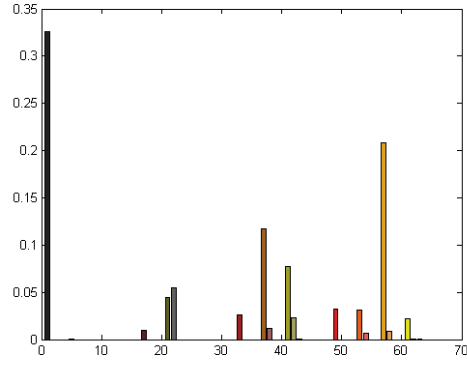
A hisztogramok összehasonlítása érdekében nem veszünk figyelembe minden intenzitás értéket, hanem a lehetséges intenzitás értékek halmazát részintervallumokra (ún. vödörökre) osztjuk, és az egyes vödörökre eső pixelintenzitások relatív gyakoriságaként értelmezzük a hisztogramot. Természetesen ez az eljárás színes képek esetén is alkalmazható, ahol az egyes színcsatornákon külön-külön megvalósítható a részintervallumokra osztás, ami a színekocka esetén a színekocka részkockákra osztását eredményezi. A 2.6. ábra szemlélteti az RGB kocka egy lehetséges 64 vödörös felosztását, a 2.7. ábrán pedig a kialakuló vödörök láthatók egy-egy reprezentáns szín alkalmazásával.



2.6. ábra. Az RGB színekocka felosztása 64 vödörré.



2.7. ábra. A kialakuló vödörök egy-egy reprezentáns színnel ábrázolva.



2.8. ábra. Színes kép és a hozzá tartozó színhisztogram.

A színes képekhez tartozó háromváltozós $H(i, j, k)$ színhisztogram könnyen átalakítható egyváltozós $h(l)$ hisztogrammá az alábbi módon:

$$h((B^2 - 1)i + (B - 1)j + k) = H(i, j, k), \quad (2.17)$$

ahol B a vödrök száma egy színcsatorna mentén, i, j és k értékei pedig 1 és B közötti egészek. Ilyen módon elégséges az egyváltozós hisztogramokat tárgyalni. A 2.8. ábrán látható egy színes kép, valamint a hozzá tartozó színhisztogram, mely az RGB színtérben minden egyes színcsatorna mentén 4-4 vödröt használva készült.

Hisztogramok összehasonlítását részletesen ismertetem a disszertáció 5. fejezetében, ezért itt nem térek ki erre.

Egyszerűen belátható, hogy a korábban definiált statisztikai momentumok leírhatók a hisztogramok segítségével is, az alábbi módon:

$$\mu = \sum_{g=1}^B gH(g), \quad (2.18)$$

$$\sigma = \sqrt{\sum_{g=1}^B (g - \mu)^2 H(g)}, \quad (2.19)$$

$$s = \sqrt[3]{\sum_{g=1}^B (g - \mu)^3 H(g)}, \quad (2.20)$$

ahol B az alkalmazott vödrök száma, g pedig a vödrök indexe.

A ferdeség (skew) leírót néhány esetben másként definiálják [121], mégpedig a középérték és a hisztogram módusz (*mode*) különbségének felhasználásával:

$$skew' = \frac{\mu - mode}{\sigma}. \quad (2.21)$$

A ferdeség ilyen értelmezése kevés számításigénnyel jár, főként ha feltételezzük, hogy más eljárásoknál az első és második momentumokat már kiszámítottuk.

Az *energia* mérték arról szolgáltat információt, hogy milyen az intenzitás értéke, vagy hisztogram vödrök eloszlása:

$$energia = \sum_{g=1}^B [H(g)]^2. \quad (2.22)$$

Az *energia* maximális 1 értéket ad, ha egyetlen hisztogram vödörbe esik az összes intenzitás érték, és annál kisebb lesz az értéke, minél inkább közelítik az intenzitás értékek az egyenletes eloszlást.

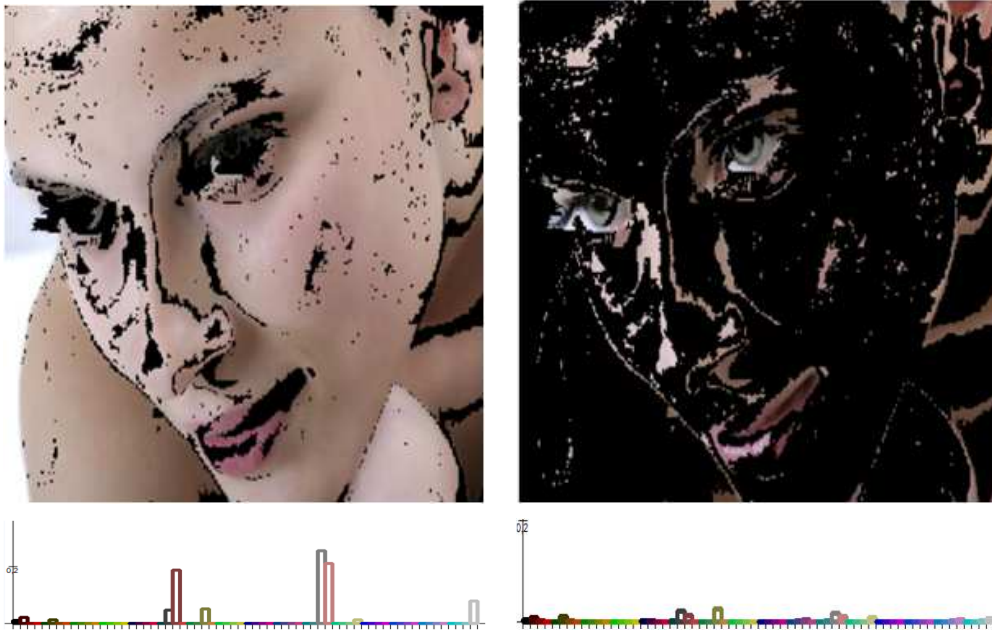
Gyakran használt leíró még az *entropia*, melyet az alábbi módon definiálunk:

$$entropia = - \sum_{g=1}^B H(g) \log_2 H(g). \quad (2.23)$$

Az entrópia minimális értékű, ha egy hisztogram vödörbe esik az összes intenzitás érték, így az *energia*-val ellentétes tulajdonsággal rendelkezik.

A szín összefüggőségi vektor (CCV - Color Coherence Vector) [76] egy olyan leíró, mely nagy mértékben hasonlít a színhisztogramra, de a képen megtalálható színek eloszlásán kívül az egyes pixelek szomszédsági viszonyait is figyelembe veszi. A CCV előállításakor először megfelelő számú vödörre osztjuk fel az alkalmazott színteret, majd minden egyes pixelre megállapítjuk, hogy színintenzitása melyik vödörbe esik. Idáig a módszer azonos a hisztogram előállítás eljárásával. Ezt követően viszont megvizsgáljuk, hogy egy adott pixel környezetében milyen pixelek találhatóak. Ezen lépés során a pixeleket két csoportra bontjuk. Az első csoportba azok a pixelek tartoznak, melyek környezetében (nyolc közvetlen szomszéd figyelembe vételével) minden pixel ugyanabba a vödörbe eső színintenzitású, mint a vizsgált pixel. Ezekről a pixelekről kijelenthető, hogy környezetük színre nézve homogén. A másik csoportba pedig azok a pixelek esnek, melyek nyolc közvetlen szomszédja között van olyan, amelynek színintenzitása másik vödörbe esik, mint a vizsgált pixelé. Ezek a pixelek heterogén régióba tartoznak. Ezt követően két hisztogramot készítünk a képről. Az első előállításakor a homogén régiókba eső pixelek színintenzitásait vesszük figyelembe, a másodikon pedig a heterogén régiókba eső pixelek intenzitásait. Természetesen az eljárás végén mindkét hisztogramot normalizáljuk a képen található pixelek számával osztva az értékeket. A 2.9. ábrán egy kép homogén és nem homogén régiói, valamint az azokhoz tartozó hisztogramok láthatóak.

A szín összefüggőségi vektor használatát az indokolja, hogy az emberi érzékelésnél nagyobb mértékben vesszük figyelembe a homogén színű tartományokat (nagy „pacákat”), mint a színre nézve változékony területeket.



2.9. ábra. Egy kép homogén régióba, illetve nem homogén régióba tartozó pixeljei, valamint a homogén és nem homogén pixeljelek alapján készített színhisztogramok.

A CCV-ben tárolt hisztogramok összehasonlítása hasonló módon történik, mint a színhisztogramok összehasonlítása, melyre részletesen a disszertáció 5. fejezetében térek ki.

2.1.2. Textúra

A textúrának kulcsszerepe van az emberi vizuális érzékelésben, a színhez hasonlóan fontos jellemző képi adatbázisokban való keresés esetén. A textúrát mindenki fel tudja ismerni, viszont a pontos definiálása és matematikai leírása nem olyan egyszerű feladat [42]. A textúra tulajdonságai például a periódicitás, vagy a skálázottság, jellemzői pedig leírhatóak az irányítottság, durvaság, kontrasztosság és hasonló kifejezésekkel [116]. A textúra jellemzése általában háromféle módon lehetséges, ezek az együttes előfordulási mátrix statisztikai jellemzői, másrészt az ún. Tamura leírók, valamint Gabor waveletek.

Együttes előfordulási mátrix

A szürkeségi értékek statisztikai jellemzői voltak az első felhasznált leírók a textúrák osztályozására. Haralick [39] javasolta a szürkeségi együttes előfordulási mátrixok (GLCM – Grey Level Co-occurrence Matrices) használatát másodrendű statisztikák kinyerésére képekből. A

GLCM használata azóta is nagyon hatékonyan bizonyult a textúrák osztályozása tekintetében [75].

Haralick definiálta az együttes előfordulási mátrixot, amely megmutatja, hogy adott intenzitású pixel párok egymástól meghatározott távolságra és irányban milyen gyakorisággal fordulnak elő. Az így előállított GLCM-ből változatos jellemzők nyerhetők ki, melyek egy kép textúrázottságának leírására alkalmasak. Ezen jellemzők közül négyet emelek ki, melyeket a 2.1. táblázatban foglalok össze.

Jellemző	Formula
Energia	$\sum_i \sum_j P^2(i, j)$
Entrópia	$\sum_i \sum_j P(i, j) \log P(i, j)$
Kontraszt	$\sum_i \sum_j (i - j)^2 P(i, j)$
Homogenitás	$\sum_i \sum_j \frac{P(i, j)}{1 + i - j }$

2.1. táblázat. Jellemzők kiszámítása a $P(i, j)$ normalizált együttes előfordulási mátrix felhasználásával.

Tamura leírók

A [116]-ban kidolgozásra kerültek azok a textúra leírók, melyek jól illeszkednek az emberi vizuális érzékeléshez. Hat különböző textúra leírót definiáltak, melyeket összehasonlítottak pszichológiai mérési eredményekkel. A bevezetett jellemzők a durvaság (coarseness), a kontrasztosság (contrast), az irányítottság (directionality), a vonalszerűség (line-likeness), a szabályosság (regularity) és az érdesség (roughness). Ezek közül az első három, melyet ma is széles körben alkalmaznak akár egymagukban, akár kombinálva azokat [42].

A *durvaságnak* közvetlen kapcsolata van a skálázottsághoz és az ismétlődési mértékhez, ezt tekinthetjük a legalapvetőbb textúra jellemzőnek. A durvaság definiálásához először be kell vezetnünk az $A_k(x, y)$ -lal jelölt átlagot egy $2^k \times 2^k$ méretű szomszédsági mátrixban az (x, y) koordináták körül:

$$A_k(x, y) = \sum_{i=x-2^{k-1}}^{x+2^{k-1}-1} \left(\sum_{j=y-2^{k-1}}^{y+2^{k-1}-1} \frac{f(i, j)}{2^{2k}} \right), \quad (2.24)$$

ahol $f(x, y)$ a kép szürkességi intenzitás értéke az (x, y) helyen. Ezt követően minden egyes pontra képezzük a különbséget mind vízszintes, mind függőleges irányban azon átlag pároknak, melyek ablakai szomszédosak, de nem fedik át egymást. Így például vízszintes szomszédok

esetén:

$$E_{k,h}(x, y) = |A_k(x + 2^{k-1}, y) - A_k(x - 2^{k-1}, y)|. \quad (2.25)$$

Ezután minden egyes pixelre meghatározzuk az

$$S_{\text{best}}(x, y) = 2^k \quad (2.26)$$

legjobb ablakméretet, amelyre k maximalizálja E értékét bármely irány esetén, azaz

$$E_k = E_{\text{max}} = \max \{E_1, E_2, \dots, E_L\}. \quad (2.27)$$

Amennyiben minden (x, y) pixelre előállítottuk az $S_{\text{best}}(x, y)$ legjobb ablakméretet, akkor a durvaságot az alábbi módon definiálhatjuk:

$$F_{\text{crs}} = \frac{1}{M \cdot N} \sum_{i=1}^M \sum_{j=1}^N S_{\text{best}}(i, j), \quad (2.28)$$

ahol M a kép sorainak, N pedig az oszlopainak száma.

A *kontraszt* definiálásához először bevezetjük a negyedi momentumot:

$$\mu_4 = \sqrt[4]{\frac{\sum_{i=1}^M \sum_{j=1}^N (f(i, j) - \mu)^4}{M \cdot N}}. \quad (2.29)$$

Ebből definiálható a lapoltság α_4 értéke:

$$\alpha_4 = \frac{\mu_4}{\sigma^4}. \quad (2.30)$$

A kontraszt mértéke pedig:

$$F_{\text{con}} = \frac{\sigma}{(\alpha_4)^n}, \quad (2.31)$$

ahol n értéke a Tamura által elvégzett kísérletekben 8, 4, 2, 1, 1/2, 1/4 és 1/8 volt. Tamura kísérletei alapján a leghatékonyabb eredményeket az $n = 1/4$ esetén kapjuk.

Az *irányítottság* egy régió felett a gradiens értékéből számítható ki [116].

Gabor wawelet

A $P \times Q$ méretű $I(x, y)$ kép esetén a diszkrét Gabor wawelet transzformáció az alábbi módon adható meg:

$$G_{mn}(x, y) = \sum_s \sum_t I(x - s, y - t) \psi_{mn}^*(s, t), \quad (2.32)$$

ahol s és t a szűrő maszk méretei, ψ^*_{mn} pedig a komplex konjugáltja a következő wavelet függvénynek:

$$\psi(x, y) = \frac{1}{2\pi\sigma_x\sigma_y} \exp\left[-\frac{1}{2}\left(\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2}\right)\right] \cdot \exp(j2\pi Wx), \quad (2.33)$$

ahol W a modulációs frekvencia. Az ön hasonlósági Gabor waveletek megkaphatók az alábbi generátor függvény alkalmazásával:

$$\psi_{mn}(x, y) = a^{-m}\psi(\tilde{x}, \tilde{y}), \quad (2.34)$$

ahol m és n meghatározza a wavelet skáláját és orientációját, $m = 0, 1, \dots, M-1$ és $n = 0, 1, \dots, N-1$ értékekkel, valamint

$$\tilde{x} = a^{-m}(x \cos \theta + y \sin \theta), \quad (2.35)$$

$$\tilde{y} = a^{-m}(-x \sin \theta + y \cos \theta), \quad (2.36)$$

ahol $a > 1$ és $\theta = n\pi/N$.

A fenti egyenletekben szereplő változókat az alábbi módon érdemes megválasztani [133]:

$$a = (U_h/U_l)^{\frac{1}{M-1}}, \quad (2.37)$$

$$W_{m,n} = a^m U_l, \quad (2.38)$$

$$\sigma_{x,m,n} = \frac{(a+1)\sqrt{2\ln 2}}{2\pi a^m (a-1)U_l}, \quad (2.39)$$

$$\sigma_{y,m,n} = \frac{1}{2\pi \tan\left(\frac{\pi}{2N}\right) \sqrt{\frac{U_h^2}{2\ln 2} - \left(\frac{1}{2\pi\sigma_{x,m,n}}\right)^2}}. \quad (2.40)$$

Zhang et al. javaslata [133], hogy $U_l=0,05$, $U_h=0,04$ értékeket érdemes választani, valamint a szűrő maszk mérete 60×60 -as legyen.

A Gabor szűrő alkalmazását követően meghatározható a magnitúdók mátrixa:

$$E(m, n) = \sum_x \sum_y |G_{mn}(x, y)|, \quad (2.41)$$

ahol $m = 0, 1, \dots, M-1$ és $n = 0, 1, \dots, N-1$.

Annak érdekében, hogy a homogén textúrával rendelkező régiókkal foglalkozhassunk definiálnunk kell a következő középértékeket és szórásokat:

$$\mu_{mn} = \frac{E(m, n)}{P \cdot Q}, \quad (2.42)$$

$$\sigma_{mn} = \frac{\sqrt{\sum_x \sum_y (|G_{mn}(x, y)| - \mu_{mn})^2}}{P \cdot Q}. \quad (2.43)$$

Az I és a J kép textúrája összemérhető az alábbi textúra hasonlósági mértékkel:

$$D(I, J) = \sum_m \sum_n d_{mn}(I, J), \quad (2.44)$$

ahol

$$d_{mn} = \sqrt{(\mu_{mn}^I - \mu_{mn}^J)^2 + (\sigma_{mn}^I - \sigma_{mn}^J)^2}. \quad (2.45)$$

2.1.3. Alakzat

Az alakzat leírására jól használható módszer az MPEG-7 szabványban definiált két alakzat leírási módszer, egyik a régió alapú, másik pedig a kontúr alapú [5].

A régió alapú alakzat leíró egy többretegű sajátvektor leíró, amely a Zernike momentumokon és az ún. angular-radial transzformáción (ART) alapul.

A Zernike momentumok [50, 51] a Zernike polinomokból származtathatók, amelyek teljes ortogonális halmazt alkotnak egy egység sugarú kör belsejében [68]. Jelöljük $V_{nm}(x, y)$ -nal ezen polinomok halmazát:

$$V_{nm}(x, y) = V_{nm}(\rho\theta) = R_{nm}(\rho) \cdot e^{jm\theta}, \quad (2.46)$$

ahol

n pozitív egész vagy nulla,

m nemnulla egész, mely teljesíti azt a feltételt, hogy $n - |m|$ páros, illetve $|m| \leq n$,

ρ az origó és az (x, y) koordináta távolsága,

θ ρ és az x -tengely által bezárt előjeles szög,

$R_{nm}(\rho)$ sugárirányú polinom az alábbi definíció szerint:

$$R_{nm}(\rho) = \sum_{s=0}^{(n-|m|)/2} (-1)^s \frac{(n-s)!}{s! \left(\frac{n+|m|}{2} - s\right)! \left(\frac{n-|m|}{2} - s\right)!} \rho^{n-2s} \quad (2.47)$$

Az ART együtthatók pedig az alábbi módon definiálhatók:

$$F_{nm} = \langle V_{nm}(\rho, \theta), f(\rho, \theta) \rangle = \int_0^{2\pi} \int_0^1 V_{nm}^*(\rho, \theta), f(\rho, \theta) \rho d\rho d\theta, \quad (2.48)$$

ahol $f(\rho, \theta)$ a kép polár koordinátákon értelmezett intenzitás függvénye, $V_{nm}(\rho, \theta)$ pedig az n és m rendű ART bázis függvény. A bázisfüggvények szeparálhatók a szög és távolság koordináták alapján:

$$V_{nm}(\rho, \theta) = \frac{1}{2\pi} \exp(jm\theta) R_n(\rho), \quad (2.49)$$

ahol

$$R_n(\rho) = \begin{cases} 1, & \text{ha } n = 0, \\ 2 \cos(\pi n \rho), & \text{ha } n \neq 0. \end{cases} \quad (2.50)$$

A kontúr alapú alak leíró két jellemzőt használ. Egyik az ún. köralakútság, amely az alakzat kerületének négyzete osztva az alakzat területével. Másik leíró az excentricitás az alábbi definíció alapján:

$$eccentricity = \sqrt{\frac{i_{20} + i_{02} + \sqrt{i_{20}^2 + i_{02}^2 - 2i_{20}i_{02} + 4i_{11}^2}}{i_{20} + i_{02} - \sqrt{i_{20}^2 + i_{02}^2 - 2i_{20}i_{02} + 4i_{11}^2}}}, \quad (2.51)$$

ahol

$$i_{02} = \sum_{k=1}^M (y_k - y_c)^2, \quad (2.52)$$

$$i_{11} = \sum_{k=1}^M (x_k - x_c)(y_k - y_c), \quad (2.53)$$

$$i_{20} = \sum_{k=1}^M (x_k - x_c)^2, \quad (2.54)$$

M a kontúrvonalon belüli pixelek száma, (x_c, y_c) pedig az alakzat tömegközéppontjának koordinátái.

2.2. Tesztelési eljárások

A kifejlesztett algoritmusok tesztelése gyakran szubjektív döntéseken alapul, hiszen a képi adatbázisokban történő kereséstől azt várjuk el, hogy olyan eredményt szolgáltatson, mely a felhasználó szubjektív igényeinek megfelel. A szubjektivitás kiküszöbölése érdekében többféle megoldás lehetséges.

A tesztelés során lehetséges mérőszámok definiálása, melyekre leggyakrabban a precíziót és a felidzését használják [26]. Ha van N darab képet tartalmazó teszt adatbázisunk, melyből Q darab számít relevánsnak találatnak egy keresés során, Z jelöli az elvárt releváns találatok számát, P pedig az eredmény lista hosszát. Ezek ismeretében meghatározható a rendszert jellemző

két mérőszám:

$$precizio = \frac{Q}{P}, \quad (2.55)$$

$$felidezes = \frac{Q}{Z}. \quad (2.56)$$

A tesztelési eredmények összehasonlíthatósága érdekében több tesztadatbázist is készítettek, melyek lehetővé teszik, hogy különböző algoritmusokat ugyanolyan körülmények között lehessen tesztelni. A leggyakrabban használt két adatbázis az Amsterdam Library of Object Images [33] és a Columbia Object Image Library [72]. Mindkét adatbázis homogén háttérbe helyezett tárgyakról készített fotókat tartalmaz azonos megvilágítás mellett, ahol a tárgyakról 5°-os függőleges tengely körüli elforgatással több fénykép is készült.

Az eredmények „jósága” olyan módon is mérhető, hogy ha vizsgáljuk, hogy egy adott képhez mely N darab kép van a legközelebb, akkor ezt követően megnézzük, hogy az eredményül kapott képekhez legközelebbi N kép között megtalálható-e az eredeti kép. Így módon azokat a képeket tekintjük releváns találatoknak, melyekhez legközelebbi N kép között szerepel az eredeti kép is.

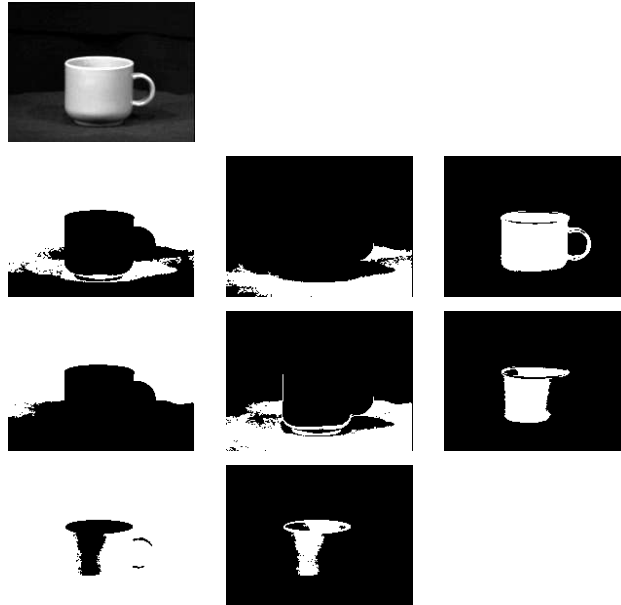
3. fejezet

Képfeldolgozási algoritmusok automatikus paraméterezése

A képfeldolgozás területén gyakran használt algoritmusok általában egy vagy több paramétert használnak. A paraméterek értékétől nagy mértékben függ, hogy az alkalmazott algoritmus milyen eredményt szolgáltat. A 3.1. ábrán látható például egy szürkeárnyaltos kép három különböző paraméter értékkel előállított homogén régiófelbontása. A 3.2. ábrán pedig élkereső algoritmus eredménye látható különböző paraméterek alkalmazása esetén. A használt paramétereket a gyakorlatban az adott képekre optimalizálják, majd ezt követően minden képre ugyanazt a paraméter értéket használják. Ez gyakran könnyen elvégezhető, mert egy adott képtípusnál a használt algoritmus számára legmegfelelőbb paraméterértékek ugyanazok.

Képi adatbázisban azonban, főleg olyan adatbázisokban, ahol a képek fajtája nagy spektrumot ölel fel, különböző típusú képekhez különböző paraméterek szolgáltatják a legmegfelelőbb eredményt. Így képi adatbázisokban való vizsgálatoknál általában nincsen lehetőség arra, hogy a paraméter értékeket minden képnél külön-külön optimalizáljuk. Ennek a problémának megoldása olyan automatizált paraméter meghatározó eljárás készítése és használata, amely az adott képhez leginkább illeszkedő paramétereket önmaga is képes megtalálni.

Automatikus paraméter meghatározás területén publikált cikkek közül hármat szeretnék megemlíteni. A [41] mű range képek szegmentálási algoritmusainak összehasonlítása kapcsán nyújt részletes elemzést. A [49] cikk valószínűségi alapon elemzi a legjobb paraméterezések kiválasztásának lehetőségét. Képi adatbázisokban használt wavelet alapú hasonlósági mértékek paraméterezésének összehasonlítását mutatja be a [71] cikk.



3.1. ábra. Szürkeárnyaltos kép régióinak meghatározása három különböző paraméterbeállítás esetén. Az első sorban a szürkeárnyaltos kép látható, alatta pedig minden egyes sorban valamilyen paraméterezés mellett előálló régiók bináris képe. A megtalált régiókat fehér színnel ábrázoltuk.

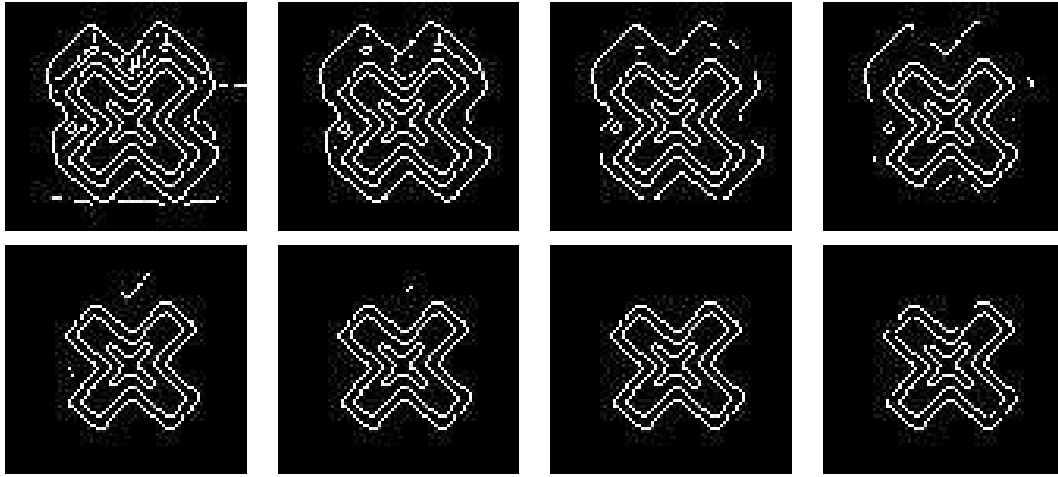
3.1. Szegmentáló eljárások automatikus paraméterezése

3.1.1. Algoritmus leírása

Algoritmusom egy kép több különböző paraméterezés használatával legyártott régiófelbontásából megpróbálja a legjobb régiófelbontást kiválasztani. Ez alapvetően kétféle módon történhet. Az első megközelítésnél rendelkezésünkre áll egy jónak tekinthető régiófelbontás és ahhoz hasonlítjuk az előállt régiófelbontásokat, majd a leginkább hasonló paraméterezését tekintjük a legjobb paraméterezésnek. Ez az eljárás használható például akkor, ha szín alapú régiókat akarunk előállítani úgy, hogy a kép szürkeárnyaltos régiói rendelkezésünkre állnak [97].

A másik megközelítés esetén nem áll rendelkezésünkre referenciaként szolgáló referenciáfelbontás. Ebben az esetben abból a feltételezésből indultunk ki, hogy a jó régiófelbontás valószínűleg gyakran előfordul, sőt ezt tekinthetjük a leggyakoribbnak. Így, ha megtaláljuk azt a régiófelbontást, amely a legtöbb másik régiófelbontáshoz hasonlít valamilyen hasonlósági mérték alapján, akkor ahhoz a régiófelbontáshoz tartozó paraméterezés lesz a legjobb paraméterezés. Az alábbiakban ezt az algoritmust mutatom be részleteiben.

Első lépésként előállítjuk egy kép több paraméterezését. Ezek eredménye legyen $\bar{p}_1, \bar{p}_2, \dots, \bar{p}_k$,



3.2. ábra. Élkereső algoritmus eredménye nyolc különböző paraméterérték használata esetén.

ahol k jelöli a paraméterezések számát. Minden α és β esetén, ahol $\alpha \neq \beta$ összehasonlítjuk (nem szimmetrikusan) p_α -t és p_β -t az alábbi módon:

Polárkoordináták szerint rendezzük az α ($\alpha = 1, 2, \dots, k$) paraméterezéshez tartozó tartományokat. A polárkoordinátás rendezésnél a tartományok tömegközéppontjainak a kép bal felső sarkától vett távolságát, illetve a vízszintes iránnyal bezárt szögét vesszük figyelembe. A rendezést követően megkapjuk a v_α tartományokat tartalmazó vektort.

A v_α és v_β vektorok összehasonlítását a 3.3 ábrán látható algoritmus alapján végezzük el.

Az összehasonlítás eredményeként kapunk egy $\bar{v}_{\alpha(\beta)}$, mely az eredeti v_α vektorból azokat a komponenseket tartalmazza, amelyekhez találtunk megfelelő komponenst a v_β vektorban. Ez a vektor nem csak \bar{p}_α paraméterezéstől, hanem a \bar{p}_β paraméterezéstől is függ, ezért jelenik meg az indexben α mellett β is. Hasonló módon az algoritmus szolgáltat egy $\bar{v}_{\beta(\alpha)}$ vektort is, mely v_β -nak azokat a komponenseit tartalmazza, amelyek v_α valamely komponenséhez megfeleltek. Az algoritmusból következik, hogy $\bar{v}_{\alpha(\beta)}$ és $\bar{v}_{\beta(\alpha)}$ hossza megegyezik és megfelelő elemeikre igaz, hogy $\bar{v}_{\alpha(\beta)i}$ tömegközéppontja benne van a $\bar{v}_{\beta(\alpha)i}$ tartományban. (Ez fordítva nem feltétlenül teljesül, ezért nem szimmetrikus az összehasonlítás.)

A $\bar{v}_{\alpha(\beta)}$ és $\bar{v}_{\beta(\alpha)}$ vektorok távolságának definiálásához bevezetjük a következő jelöléseket:

$$d'_{\alpha,\beta,\theta} = \frac{|\bar{v}_{\alpha(\beta)\theta} \Delta \bar{v}_{\beta(\alpha)\theta}|}{|\bar{v}_{\alpha(\beta)\theta} \cup \bar{v}_{\beta(\alpha)\theta}|}, \quad (3.1)$$

ahol Δ a halmazokon értelmezett szimmetrikus differencia műveletet jelöli, amelyet pixelhalmazok fölött értelmezünk esetünkben. \cup hasonlóképpen a pixelhalmazokon értelmezett unió műveletet jelöli. $|\cdot|$ az argumentumában szereplő pixelhalmaz elemszámát adja vissza. θ értéke

```

i:=1
j:=1
Ciklus amíg i <= |vα|
  Cilus amíg j <= |vβ|
    Ha vαi.center ∈ vβj akkor
      i++
      j++
      Kiugrás a belső ciklusból
    Különben
      j++
  Elágazás vége
Cilus vége
i++
Ciklus vége

```

3.3. ábra. A v_α és v_β vektorok összehasonlítása

1-től $\bar{v}_{\alpha(\beta)}$, illetve a vele egyező hosszúságú $\bar{v}_{\beta(\alpha)}$ vektor hosszáig fut.

$$\rho_{\alpha,\beta,\theta} = \begin{cases} 1 & \text{ha } d'_{\alpha,\beta,\theta} < \varepsilon \\ 0 & \text{egyébként} \end{cases} \quad (3.2)$$

ε értékét kísérletek alapján 0.75-nak választottuk.

Ezek felhasználásával definiáljuk a $\bar{v}_{\alpha(\beta)}$ és $\bar{v}_{\beta(\alpha)}$ vektorok távolságát az alábbi módon:

$$d(\bar{v}_{\alpha(\beta)}, \bar{v}_{\beta(\alpha)}) = \sum_{\theta=1}^{|\bar{v}_{\alpha(\beta)}|} \rho_{\alpha,\beta,\theta}. \quad (3.3)$$

Minden α -ra ($1 \leq \alpha \leq k$) és α -val nem megegyező β -ra ($1 \leq \beta \leq k$) legyártjuk a $d(\bar{v}_{\alpha(\beta)}, \bar{v}_{\beta(\alpha)})$ távolságot, így kapunk egy $k \times k$ méretű mátrixot:

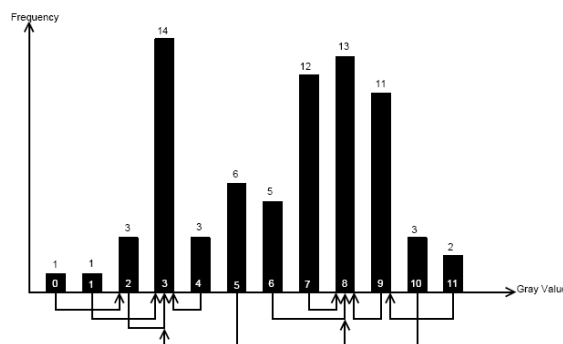
$$C = \begin{bmatrix} 0 & d(\bar{v}_{1(2)}, \bar{v}_{2(1)}) & \dots & d(\bar{v}_{1(k)}, \bar{v}_{k(1)}) \\ d(\bar{v}_{2(1)}, \bar{v}_{1(2)}) & 0 & \dots & d(\bar{v}_{2(k)}, \bar{v}_{k(2)}) \\ \vdots & \vdots & \ddots & \vdots \\ d(\bar{v}_{k(1)}, \bar{v}_{1(k)}) & d(\bar{v}_{k(2)}, \bar{v}_{2(k)}) & \dots & d(\bar{v}_{k(k)}, \bar{v}_{k(k)}) \end{bmatrix} \quad (3.4)$$

Ezt követően C minden sorának vesszük a sorösszegét, azaz a \bar{p}_α paraméterezéshez tartozó távolságok összegeit. Azt a \bar{p}_α paraméterezést tekintjük a legjobb paraméterezésnek, amelyhez tartozó sorösszege a C mátrixnak maximális. Lehetséges, hogy több ilyen paraméterezés is található, ha a maximális sorösszeg több esetben is előáll, ilyenkor mindegyik megfelelő paraméterezést a legjobbnak tekintjük.

3.1.2. Tesztelés

Alkalmazott szegmentáló algoritmus

A tesztelés során az ún. Peaks and Natural Intervals algoritmust [66, 52] használtam. Az algoritmus első lépésben n számú egyenlő szélességű vödört használva leggyártja a szürkeárnyaltos képünk hisztogramját. Ezt követően minden vödörré megnézi, hogy a tőle legfeljebb s távolságra lévő vödörök közül melyikben található meg a legnagyobb hisztogram érték. Az adott vödörből az s sugarú környezetében található legnagyobb hisztogram értékű vödörré fog mutatni egy mutató. Ezután klaszterezzük a hisztogramot oly módon, hogy a mutatóval összekötött vödörök ugyanazon klaszterbe kerüljenek. 3.4 ábra szemlélteti az algoritmust.



3.4. ábra. Peaks and Natural Intervals algoritmus $n = 12$ és $s = 2$ esetén

A kép szegmentálása során azokat a pixeleket tekintjük egy régióba tartozónak, melyek intenzitás értékei ugyanazon legyártott klaszterbe tartoznak és összefüggő tartományt alkotnak.

A használt algoritmus két paramétert is tartalmaz, de a paraméter meghatározó algoritmus ettől függetlenül alkalmazható.

Képi adatbázis

Tesztelésem során az Amsterdam Library of Object Images [33] 1000 különböző képet tartalmazó képi adatbázis véletlenszerűen kiválasztott 100 darab szürkeárnyalatos képét használtam. A használt képek közül mutatok be néhányat a 3.5 ábrán. Minden adatbázisbeli képen homogén sötét háttérben lévő egy-egy objektum található. Az objektumok között vannak szín alapján homogének és inhomogének is.



3.5. ábra. Az ALOI adatbázis néhány képe

Tesztelésnél használt beállítások

A tesztelésnél n értéke 4, 8, 16, 32, 64, 128 és 256 volt, míg s értéke 1-től n aktuális értékének feléig valamely kettőhatvány volt. Így összességében 35 különféle paraméterezést használtam.

Az előálló régiók közül csak a kép méretének 5%-át meghaladó méretűeket vettem figyelembe és azokkal a régiófelbontásokkal nem foglalkoztam, amikor a képen legfeljebb egy régió állt elő.

A tesztelést MATLAB környezetben végeztem.

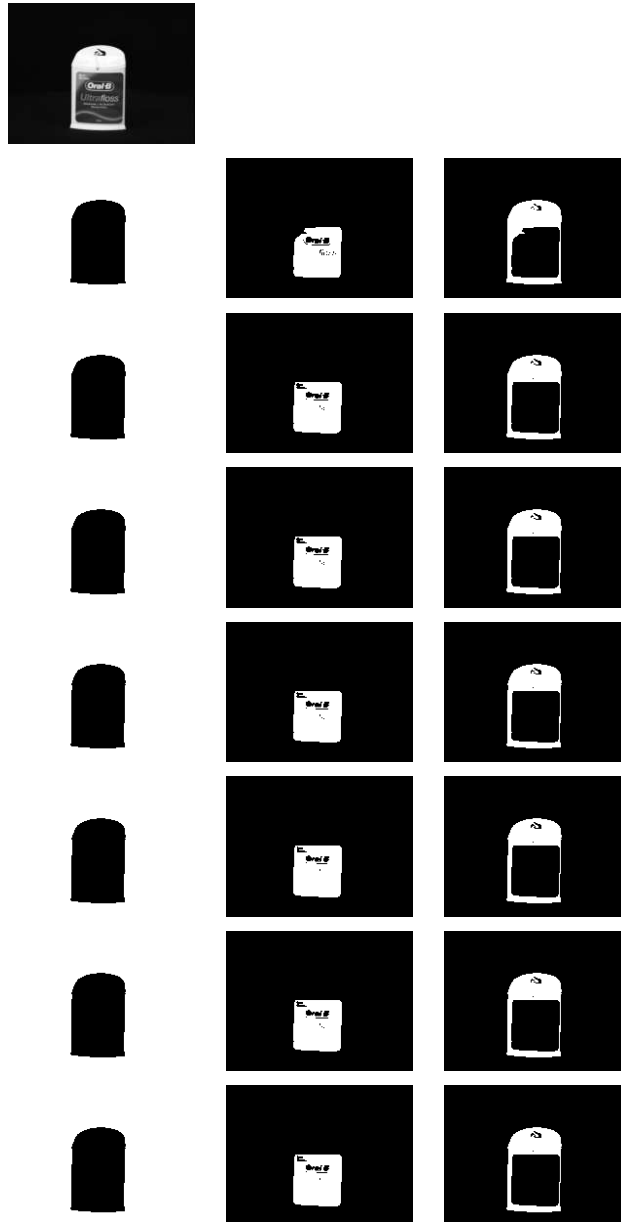
3.1.3. Eredmények

Az eredmények ismertetését a legjobbnak talált paraméterezések számának bemutatásával kezdem. A 3.1 táblázatban látható, hogy a legjobb paraméterezések egyes darabszámai hányszor fordultak elő a 100 vizsgált esetből. (A táblázatban csak azokat az eseteket tüntettem fel, amelyek legalább egyszer előálltak.)

Legjobb paraméterezések száma	Gyakoriság
1	20
2	13
3	19
4	10
5	9
6	1
7	4
9	2
10	1
11	3
12	1
13	1
14	3
15	2
17	1
18	2
19	1
20	1
21	1
22	1
24	2
25	1
32	1

3.1. táblázat. Legjobbnak talált paraméterezések számának gyakorisága

Ideális esetben azt vártuk volna, hogy minden esetben egyetlen legjobb paraméterezést talál az algoritmus. Ám sok esetben azért talált több paraméterezést is a legjobbnak, mivel ezek által előállított régiók nagymértékben hasonlítanak egymáshoz és az algoritmus nem tudja közülük a legmegfelelőbbet kiválasztani. Példaként egy ilyen esetet mutat a 3.6. ábra, ahol a 7 legjobbnak talált régiófelbontását szemléltetem egy képnek. Az ábráról jól látható, hogy a megtalált régiófelbontások lényegében teljesen azonosak.



3.6. ábra. Az első sorban látható kép különböző paraméterezésekhez tartozó megtalált régióit mutatják az egyes sorokban látható bináris képek. A megtalált régiókat jelöltük fehér színnel.

Érdeemes lenne tehát inkább azt megvizsgálni, hogy az előálló régiófelbontások közül az algoritmus által legjobbnak tekintettek esetében hányszor található olyan régiófelbontás, amely

alapvetően rossznak tekinthető. Ez a képek 25%-ánál fordult elő, mégpedig általában olyan esetben, amikor sok legjobbnak tekintett régiófelbontást szolgáltatott az algoritmus, amelyek között volt pár nem megfelelő. Összesen három esetben fordult az elő, hogy az 1–3 darab legjobbnak talált paraméterezéshez tartozó régiófelbontás nem volt megfelelő minőségű.

3.2. Élkereső algoritmusok automatikus paraméterezése

Célunk, hogy egy szürkeárnyalatos képnek olyan élmátrixát találjuk meg, amely a leginkább illeszkedik a képen található információkhoz. Természetesen ez nagyon szubjektív, de azt szeretnénk, hogy a kapott eredményt a felhasználó megfelelőnek találja.

Hasonló eljárások már léteznek, viszont kísérleteim azt igazolták, hogy a kifejlesztett algoritmus a felhasználói igényeknek megfelelőbb eredményeket szolgáltat. A MATLAB Image Processing Toolboxában implementált `edge` függvény használ egy paraméter meghatározó eljárást, mely a kép pixeleinek intenzitás eloszlásának statisztikai jellemzőit veszi alapul [80]. Egy másik paraméter meghatározó módszer [131] az ún. ROC görbe diagnózis és statisztikai χ^2 próba alkalmazásával határozza meg a leginkább használható paramétert.

3.2.1. Az alkalmazott algoritmus

Az élkereső algoritmusok általában azon az elven működnek, hogy meghatározzuk minden egyes pixelre a környezetében lévő pixelek intenzitásához való viszonya alapján a gradiens vektor nagyságát és irányát. Ezt követően egy pixelt akkor tekintünk élpixelnek, ha a gradiens vektor nagysága meghalad egy előre definiált küszöbértéket. A különbség főként ott van az élkereső algoritmusok között, hogy melyik eljárás milyen módon határozza meg a gradiens vektort [120, 35]. Természetesen ettől nagy mértékben függ a jó küszöbérték megválasztása is. A küszöb viszont emellett a kép tartalmától is függ, ezért szükséges, hogy automatizálni tudjuk annak meghatározását.

Kifejlesztett algoritmusom abból a hipotézisből indul ki, hogy ha rendelkezésünkre áll egy kép több különböző paraméterértékkel előállított élmátrixa, akkor az élmátrixok közül az a leginkább a felhasználó elvárásainak, amely a legtöbb más paraméterezéssel előállított élmátrixhoz hasonlít. Ennek meghatározásához elő kell állítanunk egy kép több paraméter értékhez tartozó élmátrixát, valamint értelmeznünk kell ezen élmátrixok között egy hasonlósági mértéket. Ennek megvalósítását az alábbi algoritmusban részletezem.

- I. Előfeldolgozásként 3×3 -as medián szűrőt használunk a zajok csökkentése érdekében.

II. Meghatározzuk a kép N darab különböző élmátrixát N különböző előre definiált küszöbérték használatával. Jelöljük ezeket E^i -vel, ahol $i \in \{1, 2, \dots, N\}$.

III. Kiszámítjuk az élpixelek számát minden egyes E^i élmátrixban, majd vesszük ezek átlagát.

$$\text{average}(|E^i|) = \frac{\sum_{i=1}^n |E^i|}{N} \quad (3.5)$$

IV. Az algoritmus további részében csak azokat az élmátrixokat vesszük figyelembe, melyek élpixeleinek száma meghaladja az előbb meghatározott átlagot, azaz

$$|E^i| \geq \text{average}(|E^i|). \quad (3.6)$$

A későbbiekben ezeket az élmátrixokat E'^j -vel jelöljük, ahol $j \in \{1, 2, \dots, M\}$ és M a megmaradó mátrixok száma.

V. A megmaradó M darab élmátrixon dilataciót hajtunk végre 3×3 -as méretű maszkkal. Az eredményül kapott dilatált élmátrixokat DE'^j -vel jelöljük.

VI. Definiáljuk két élmátrix metszetét oly módon, hogy a $DE'^k \cap DE'^l$ -lrel jelölt metszet mátrix azon pixeli élpixelek, melyek mindkét kiindulási élmátrixban is élpixelek voltak. Hasonló módon defináljuk két élmátrix unióját is: a $DE'^k \cup DE'^l$ -lrel jelölt unió mátrix azon pixeli élpixelek, melyek legalább az egyik kiindulási mátrixban élpixelek voltak.

Ezt követően már értelmezni tudjuk a metszet és unió segítségével két dilatált élmátrix távolságát az alábbi módon:

$$d(DE'^k, DE'^l) = 1 - \frac{DE'^k \cap DE'^l}{DE'^k \cup DE'^l}. \quad (3.7)$$

VII. Az előbb definiált távolság használatával elkészíthetjük az $M \times M$ -es D mátrixot, ahol

$$D_{ij} = d(DE'^i, DE'^j). \quad (3.8)$$

Könnyen belátható, hogy D szimmetrikus mátrix.

VIII. Utolsó lépésként meghatározzuk D minden egyes oszlopában lévő elemek összegét. Amelyik oszlopban (k indexű) ez az összeg minimális lesz, az ahhoz tartozó paraméterezést tekintjük a képhez tartozó legjobb paraméterezésnek.

3.2.2. Tesztelés

Tesztelésnél a Sobel és Prewitt élkereső algoritmusokat [35, 120] használtam. A Sobel algoritmus az

$$M_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad (3.9)$$

szűrőmaszkot alkalmazza az f szürkeárnyalatos képmátrix x -irányú $G_x(f)$ -fel jelölt gradiensek meghatározására. Az y irányú $G_y(f)$ gradiens előállításához pedig az

$$M_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad (3.10)$$

maszkot használjuk. A kapott gradiensekből az élmátrix úgy határozható meg, hogy azok a pixelek lesznek élpixelek, melyek gradiens vektorának nagysága egy előre meghatározott τ küszöbértéknél nagyobb, azaz

$$\sqrt{G_x^2(f) + G_y^2(f)} \geq \tau. \quad (3.11)$$

A Prewitt élkereső abban tér el a Sobel módszertől, hogy a szűrőmaszkok különböznek:

$$M_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \quad (3.12)$$

$$M_y = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \quad (3.13)$$

Az eljárás tesztelésekor véletlenszerűen kiválasztottam ötszáz képet az Amsterdam Library of Object Images (ALOI) adatbázisból [33], amely ezer különböző objektumról készült képeket tartalmaz. Az adatbázis néhány képe a 3.5. ábrán látható. Az adatbázis minden képe egy tárgyról készült, amely homogén sötét háttérbe van helyezve. Néhány tárgy szín tekintetében homogénnek tekinthető, míg vannak olyanok is, melyek több színből állnak.

A tesztelésnél 10 különböző paramétert használtam, azaz $N = 10$ értékkel dolgoztam. A használt küszöbértékek 0,2-től 0,2-es növekménnyel mentek 2-ig.

Annak eldöntése érdekében, hogy az automatikus meghatározott paraméterrel előállított képmátrix megfelel-e a felhasználói elvárásoknak, olyan tesztet hajtottam végre, ahol az algoritmus által generált élmátrixot és a MATLAB Image Processing Toolboxa által automatikus paraméter meghatározással előállított élmátrixot hasonlítottam össze.

Azt tapasztaltam, hogy Sobel maszk használatakor az esetek 26%-ában, Prewitt maszkot használva pedig az esetek 29,2%-ában tértek el a két vizsgált módszer által meghatározott paraméter értékek 0,1-nél kisebb mértékben egymástól. Így ezekben az esetekben lényegében közel azonos paramétereket állított elő a két módszer. A fennmaradó esetekben, tehát amikor az előállított paraméter értékek 0,1-nél nagyobb mértékben eltértek egymástól, vizsgáltam, hogy a felhasználó megítélése szerint melyik módszerrel előállított élmátrixok illeszkedik jobban az eredeti képen elvárt élekhez. Eredményeimet a 3.2. táblázatban foglaltam össze.

	Sobel szűrő	Prewitt szűrő
Közel azonos küszöbérték	26%	29,2%
Különböző küszöbérték esetén a saját eljárásunk jobb	46,8%	41,2%
Különböző küszöbérték esetén a másik eljárás jobb	13,6%	13,8%
Különböző küszöbérték esetén hasonló eredmény	13,6%	15,8%

3.2. táblázat. Két automatikus paraméter meghatározást alkalmazó élkereső eljárás összehasonlítása.

3.3. Konklúziók

Kifejlesztettem egy olyan algoritmust, mely képes különböző paraméterekkel előállított régiófelbontások közül a legjobbakat kiválasztani. Algoritmusom az esetek 71%-ában a vizsgált 35 paraméterezés közül ki tud választani legfeljebb öt olyat, amelyek jónak tekinthetők.

Hasonló alapötlet alapján kidolgoztam egy élkeresésnél alkalmazható automatikus paraméter meghatározó algoritmust. Kísérleteim azt igazolták, hogy más automatikus paraméterező algoritmusnál jobb eredményt szolgáltat az eljárásom.

4. fejezet

HOSVD alapú eljárások használata a képi adatbázisok indexelésében

A tartalom alapú kereső rendszereknél nagy jelentősége van annak, hogy az adatbázisban tárolt képek indexeinek elkészítését megelőzően milyen előfeldolgozó eljárást használunk annak érdekében, hogy az egyes képekből a képekre jellemző tulajdonságokat kinyerjük. Az előfeldolgozási eljárás számos esetben például valamilyen simítást, illetve zajcsökkentést jelent.

Simítást többféle módon is meg lehet valósítani. Leggyakoribb erre a kép szűrése például átlagoló, vagy Gauss maszkkal. Gyakran használt eljárás a kép Fourier transzformálása oly módon, hogy az előálló trigonometrikus tagokból csak az első pár tagot tartjuk meg. Hasonló módszer lehet a magasabb rendű szinguláris érték dekompozíció (HOSVD - High Order Singular Value Decomposition) használata. Ebben az esetben a képet, mint három dimenziós tenzort, ortonormált függvények kompozíciójaként állítjuk elő. Ha ebből az előállításból is csak pár tagot tartunk meg, akkor a Fourier transzformációhoz hasonlóan szűrést tudunk megvalósítani, amelynek eredménye részletgazdagabb, így a képi indexek legyártására alkalmasabb eredményt szolgáltat.

Az alábbiakban bemutatom a HOSVD eljárás matematikai hátterét és alkalmazási lehetőségeit a képek előfeldolozásában.

Jelenleg is folynak kutatások azzal kapcsolatban, hogy a HOSVD módszer által előállított ortonormált függvények a képek közvetlen indexelésében milyen módon használhatóak fel. Mivel az ezzel kapcsolatos eredmények még nem lettek publikálva, így a disszertációnak sem képezik részét.

4.1. HOSVD áttekintés

A matematika approximációs módszereit igen széleskörben alkalmazzák különböző problémák megoldása során. Legyen

$$f(\mathbf{x}), \mathbf{x} = (x_1, \dots, x_N)^T, x_n \in [a_n, b_n], 1 \leq n \leq N,$$

egy n -változós sima függvény. Az $f(\mathbf{x})$ függvény az alábbi módon approximálható egyváltozós ortonormált rendszert alkotó sima függvények segítségével:

$$f(\mathbf{x}) = \sum_{k_1=1}^{I_1} \dots \sum_{k_N=1}^{I_N} \alpha_{k_1, \dots, k_N} p_{1, k_1}(x_1) \cdot \dots \cdot p_{N, k_N}(x_N). \quad (4.1)$$

ahol a $p_{n, k_n}(x_n)$ függvények megválaszthatók egyrészt klasszikus módon ortonormált polinomok vagy trigonometrikus függvények formájában másrészt olyan jellegű függvények segítségével, melyek jellege a kiindulási n -változós függvényre nézve specifikus. Az approximáció pontossága az (4.1)-ben szereplő egyváltozós függvények számától erősen függ. Az ún. magasabb rendű szinguláris értékdekompozíció segítségével (HOSVD) egy újszerű módszer került kidolgozásra az egyváltozós függvények ill. a hozzájuk társuló súlyok numerikus meghatározására. [112, 113, 114]. A módszer ortonormált polinomok vagy trigonometrikus függvények helyett speciálisan meghatározott ortonormált rendszert alkotó függvényeket alkalmaz. Tételezzük fel, hogy $f(\mathbf{x})$ kifejezhető $\tilde{w}_{n, i}(x_n), x_n \in [a_n, b_n]$ függvények segítségével az alábbi módon:

$$f(\mathbf{x}) = \sum_{k_1=1}^{I_1} \dots \sum_{k_N=1}^{I_N} \alpha_{k_1, \dots, k_N} \tilde{w}_{1, k_1}(x_1) \cdot \dots \cdot \tilde{w}_{N, k_N}(x_N). \quad (4.2)$$

Legyen $\mathcal{A} \in \mathbb{R}^{I_1 \times \dots \times I_N}$ egy az $\alpha_{i_1, \dots, i_N}, 1 \leq i_n \leq I_n, 1 \leq n \leq N$ elemek által meghatározott N dimenziós tenzor. Vezessük be a következő jelöléseket:

- $\mathcal{A} \boxtimes_n \mathbf{U}$: n -módú tenzor-mátrix szorzat, [55]
- $\mathcal{A} \boxtimes_{n=1}^N \mathbf{U}_n$: többszörös szorzat $\mathcal{A} \boxtimes_1 \mathbf{U}_1 \boxtimes_2 \mathbf{U}_2 \dots \boxtimes_N \mathbf{U}_N$.

Az n -módú tenzor-mátrix szorzat az alábbi módon definiált:

Legyen \mathbf{U} egy $K_n \times M_n$ méretű mátrix és $\mathcal{A} \boxtimes_n \mathbf{U}$ egy $M_1 \times \dots \times M_{n-1} \times K_n \times M_{n+1} \times \dots \times M_N$ méretű tenzor, melyre fenáll a következő összefüggés:

$$(\mathcal{A} \boxtimes_n \mathbf{U})_{m_1, \dots, m_{n-1}, k_n, m_{n+1}, \dots, m_N} \stackrel{def}{=} \sum_{1 \leq m_n \leq M_n} a_{m_1, \dots, m_n, \dots, m_N} U_{k_n, m_n}$$

(4.2)-ből kiindulva az $f(\mathbf{x})$ függvény tenzor szorzat alakban az alábbi módon fejezhető ki:

$$f(\mathbf{x}) = \mathcal{A} \boxtimes_{n=1}^N \tilde{\mathbf{w}}_n(x_n), \quad (4.3)$$

ahol $\tilde{\mathbf{w}}_n(x_n) = (\tilde{w}_{n,1}(x_n), \dots, \tilde{w}_{n,I_n}(x_n))^T$, $1 \leq n \leq N$.

Belátható továbbá, hogy (4.3) felírható az alábbi formában [4, 112]:

$$f(\mathbf{x}) = \mathcal{D} \boxtimes_{n=1}^N \mathbf{w}_n(x_n), \quad (4.4)$$

ahol

– $\mathcal{D} \in \mathbb{R}^{r_1 \times \dots \times r_N}$ egy speciális ún. magntenzor az alábbi tulajdonságokkal:

I. $r_n = \text{rank}_n(\mathcal{A})$ az \mathcal{A} tenzor n -módú rangja.

$$\{(a_{i_1, \dots, i_{n-1}, 1, i_{n+1}, \dots, i_N}, \dots, a_{i_1, \dots, i_{n-1}, I_n, i_{n+1}, \dots, i_N})^T : \\ 1 \leq i_j \leq I_n, 1 \leq j \leq N\},$$

II. \mathcal{D} ortogonális: minden n, α és β , $\alpha \neq \beta$ esetében érvényes, hogy $\mathcal{D}_{i_n=\alpha}$ és $\mathcal{D}_{i_n=\beta}$ altenzorok ortogonálisak, azaz $\langle \mathcal{D}_{i_n=\alpha}, \mathcal{D}_{i_n=\beta} \rangle = 0$.

A $\langle \mathcal{D}_{i_n=\alpha}, \mathcal{D}_{i_n=\beta} \rangle$ skaláris szorzat a $\mathcal{D}_{i_n=\alpha}$ és $\mathcal{D}_{i_n=\beta}$, altenzorok megfelelő elemei szorzatának összegét jelöli.

III. Rendezettség: $\|\mathcal{D}_{i_n=1}\| \geq \|\mathcal{D}_{i_n=2}\| \geq \dots \geq \|\mathcal{D}_{i_n=r_n}\| > 0$ minden lehetséges n értékre ($\|\mathcal{D}_{i_n=\alpha}\| = \langle \mathcal{D}_{i_n=\alpha}, \mathcal{D}_{i_n=\alpha} \rangle$ a $\mathcal{D}_{i_n=\alpha}$ tenzor Kronecker-normáját jelöli).

– $\mathbf{w}_n(x_n) = (w_{n,1}(x_n), \dots, w_{n,r_n}(x_n))^T$, $1 \leq n \leq N$, elemei ortonormáltak L_2 értelemben az $[a_n, b_n]$ intervallumon azaz

$$\forall n : \int_{a_n}^{b_n} w_{n,i_n}(x_n) w_{n,j_n}(x_n) dx = \delta_{i_n,j_n}, \\ 1 \leq i_n, j_n \leq r_n,$$

ahol $\delta_{i,j}$ az ún. Kronecker féle függvény ($\delta_{i,j} = 1$, ha $i = j$ és $\delta_{i,j} = 0$, ha $i \neq j$)

A (4.4) alakot a (4.2) függvény HOSVD kanonikus alakjának nevezzük [4, 112].

Osszuk fel az $[a_n, b_n]$, $n = 1..N$ intervallumokat M_n darab diszjunkt Δ_{n,m_n} , $1 \leq m_n \leq M_n$ részintervallumra az alábbi módon:

$$\xi_{n,0} = a_n < \xi_{n,1} < \dots < \xi_{n,M_n} = b_n,$$

$$\Delta_{n,m_n} = [\xi_{n,m_n}, \xi_{n,m_n-1}).$$

Tételezzük fel hogy a (4.2) egyenletben szereplő $w_{n,k_n}(x_n)$, $x_n \in [a_n, b_n]$, $1 \leq n \leq N$ függvények szakaszonként folytonosan differenciálhatók. Tegyük fel továbbá, hogy az $f(x)$ függvény megfigyelhető annak $x_{n,m_n} \in \Delta_{n,m_n}$, $1 \leq m_n \leq M_n$, $1 \leq n \leq N$ pontjaiban.

A HOSVD-ből kiindulva egy új módszer került kidolgozásra az $f(\mathbf{x})$ függvény kanonikus formájának numerikus rekonstrukciójára annak $f(y_{i_1, \dots, i_N})$, $1 \leq i_n \leq M_n$, $1 \leq i_n \leq N$ értékei alapján[4].

Diszkrétizáljuk $f(\mathbf{x})$ -et annak rácspontjaiban az alábbi módon:

$$b_{m_1, \dots, m_N} = f(\mathbf{y}_{m_1, \dots, m_N}).$$

A kapott b_{m_1, \dots, m_N} értékek alapján hozzunk létre egy N dimenziós $M_1 \times \dots \times M_N$ méretű tenzort:

$$\mathcal{B} = (b_{m_1, \dots, m_N}). \quad (4.5)$$

Diszkrétizáljuk továbbá a $w_n(x_n)$ függvényeket x_{n,m_n} felett és az így kapott értékekből hozzunk létre a \mathbf{W}_n mátrixokat:

$$\mathbf{W}_n = \begin{pmatrix} w_{n,1}(x_{n,1}) & w_{n,2}(x_{n,1}) & \dots & w_{n,r_n}(x_{n,1}) \\ w_{n,1}(x_{n,2}) & w_{n,2}(x_{n,2}) & \dots & w_{n,r_n}(x_{n,2}) \\ \vdots & & \ddots & \vdots \\ w_{n,1}(x_{n,M_n}) & w_{n,2}(x_{n,M_n}) & \dots & w_{n,r_n}(x_{n,M_n}) \end{pmatrix} \quad (4.6)$$

A \mathcal{B} tenzor (4.4) és (4.6) segítségével egyszerűen megadható az alábbi módon:

$$\mathcal{B} = \mathcal{D} \boxtimes_{n=1}^N \mathbf{W}_n. \quad (4.7)$$

4.2. Digitális képek HOSVD alapú reprezentációja

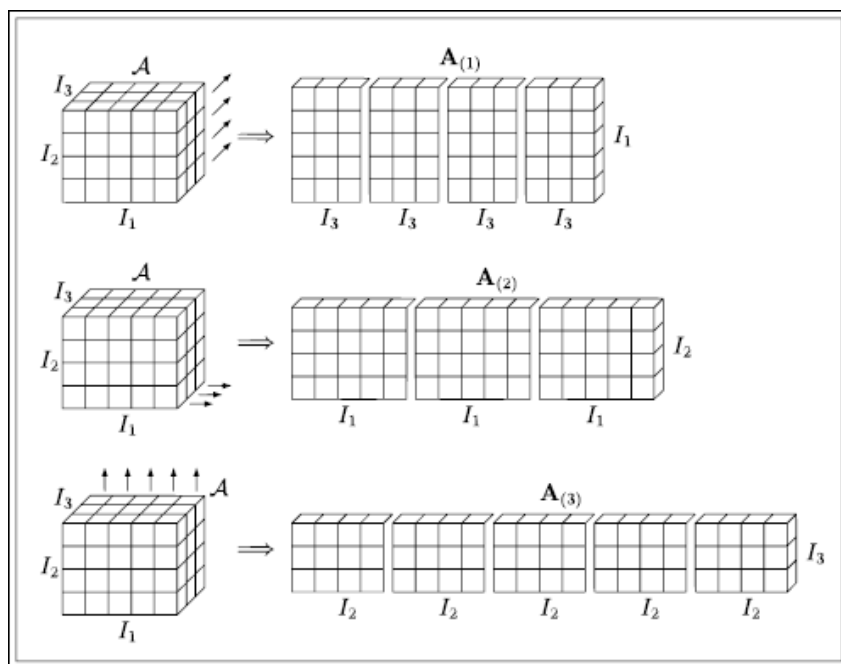
Legyen $f(\mathbf{x})$, $\mathbf{x} = (x_1, x_2, x_3)^T$ a digitális képek leíró függvény, ahol x_1 és x_2 a képpont koordinátáját, x_3 pedig a színterben fejezi ki, azaz *RGB* színtérben ez a piros, zöld és kék komponenseket jelenti. Az $f(\mathbf{x})$ függvény a fentebb már részletezett egyváltozós ortonormált rendszert alkotó függvények segítségével a következőképpen fejezhető ki:

$$f(\mathbf{x}) = \sum_{k_1=1}^{I_1} \sum_{k_2=1}^{I_2} \sum_{k_3=1}^{I_3} \alpha_{k_1, k_2, k_3} \tilde{w}_{1, k_1}(x_1) \cdot \tilde{w}_{2, k_2}(x_2) \cdot \tilde{w}_{3, k_3}(x_3). \quad (4.8)$$

A képpontok piros, zöld és kék komponensei egy $m \times n \times 3$ tenzorban tárolhatók, ahol n és m a kép szélességét és magasságát jelöli. Az így kapott tenzort jelölje \mathcal{B} . Első lépésként a $\tilde{w}_{n,k_n}, 1 \leq n \leq 3, 1 \leq k_n \leq I_n$ függvényeket kell rekonstruálni a \mathcal{B} tenzor szinguláris értékdekompozíciója segítségével, azaz \mathcal{B} felírható az alábbi módon:

$$\mathcal{B} = \mathcal{D} \boxtimes_{n=1}^3 \mathbf{U}^{(n)} \quad (4.9)$$

ahol \mathcal{D} az ún. magtenzort jelöli, az $\mathbf{U}^{(n)}, 1 \leq n \leq 3$ mátrix oszlopvektorai pedig az n -edik, $1 \leq n \leq 3$ dimenzióhoz tartozó egyváltozós ortonormált rendszert alkotó $\tilde{w}_{n,k_n}(x_n)$ függvények diskretizált változatainak felelnek meg. (Lásd a 4.1. és a 4.2. ábrát.)

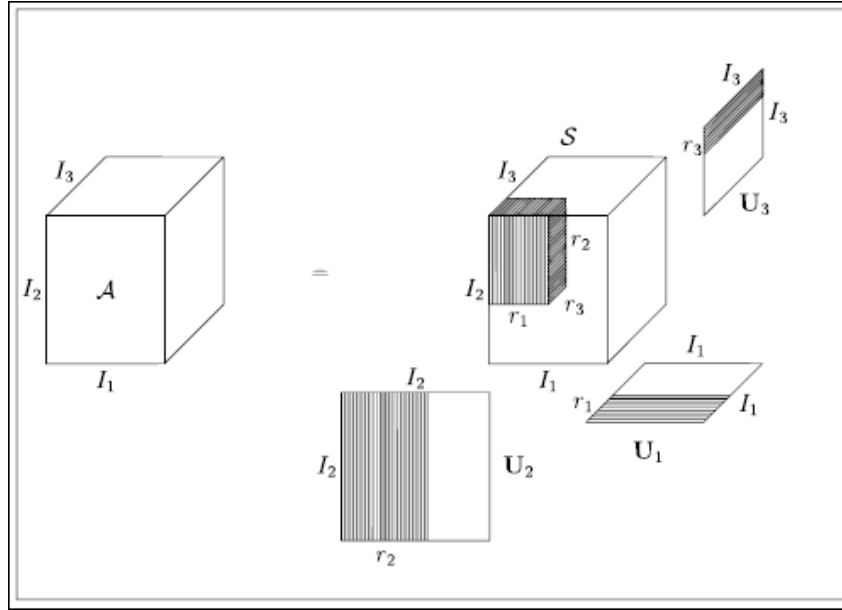


4.1. ábra. Egy három dimenziós tömb mátrixokra bontásának három lehetséges módja.

Legyen $s \in \{1, 2, \dots\}$ azon pixelek száma, melyeket a közvetlen szomszédságban lévő pixelek közé be szeretnénk ágyazni horizontális és vertikális irányokban egyaránt. Tekintsük először az $\mathbf{U}^{(1)}$ mátrix első oszlopát. Az előző fejezetek alapján belátható, hogy a $\tilde{w}_{1,1}(1)$ az $U_1^{(1)}$ vektor első, $\tilde{w}_{1,1}(2)$ a második, ..., $\tilde{w}_{1,1}(M_n)$ pedig az M_n -edik elemét jelöli.

Ahhoz, hogy a kép felbontását HOSVD-t használva megnöveljük az $\mathbf{U}^{(i)}, i = 1..2$ mátrixokat módosítani kell. Az oszlopok száma változatlan marad, a sorok száma s függvényében az alábbiak szerint fog megváltozni: (Jelöljük $\mathbf{V}^{(1)}$ -el a módosított mátrixot.)

Tekintsük példaként az $\mathbf{U}^{(1)}$ mátrix $U_1^{(1)}$ oszlopát. $V_1^{(1)}$ elemei a következőképpen alakulnak:



4.2. ábra. A HOSVD szemléltetése 3-dimenziós tömb esetén. Itt \mathcal{S} a magtenzor, U_l -ek pedig az l -módú szinguláris mátrixok.

$$V_1^{(1)}(1) := U_1^{(1)}(1), V_1^{(1)}(s+2) := U_1^{(1)}(2), V_1^{(1)}(2s+3) := U_1^{(1)}(3), \dots, V_1^{(1)}((M_n - 1)s + M_n) := U_1^{(1)}(M_n).$$

$V_1^{(1)}$ hiányzó elemeit interpolációval határozzuk meg. Interpolációs eljárásként a kubikus spline interpolációt alkalmaztuk. Hasonlóképpen járunk el az $U^{(1)}$ fennmaradó oszlopai esetében is. A nagyított képet a $V^{(n)}$ mátrixok és a magtenzor alapján a (4.9) összefüggés segítségével kapjuk meg.

4.2.1. Megjegyzések a kanonikus alakhoz

Az alábbiakban megvilágítjuk az alkalmazott HOSVD alapú algoritmus elméleti háttérét kétváltozós

$$f(x, y), 0 \leq x \leq T_1, 0 \leq y \leq T_2 \quad (4.10)$$

függvények esetében, megjegyezve, hogy a skalár esetre érvényes elméleti eredmények megfelelő módosításokkal átvihetők azokra az esetekre is, amikor az f függvény nem skalár, hanem vektorértékű (pl. a képfeldolgozás esetében 3 dimenziós vektor értékű), illetve ha mátrix értékű (ld. [1] és [2] 3. fejezetei). Az elméleti háttér segít megérteni az algoritmus jellegét és értelmezni az eredményeket, amelyek az egyes feladatokban (pl. kép (elő)feldolgozása, lényeg kiemelése, stb.) sajátos jelleget ölthetnek.

A kanonikus előállítás alapját az $N = 2$ esetben E. Schmidt integrálegyenletekre vonatkozó klasszikus eredménye [90] képezi, melyet általános formában a Hilbert tereken értelmezett ún. Hilbert-Schmidt típusú operátorokra vonatkozó elmélet tárgyal (ld. [1] és [2] 3. fejezetei). Megjegyezzük, hogy ez az előállítás felfogható úgy is, mint a mátrixok szinguláris érték felbontásának (SVD) folytonos analogonja.

Legyen $0 < T_1, T_2 < \infty$ és legyen adva egy $f(x, y)$ folytonos függvény a $[0, T_1] \times [0, T_2]$ -on. Jelölje \mathcal{H}_1 , illetve \mathcal{H}_2 a $[0, T_1]$, illetve $[0, T_2]$ felett négyzetesen integrálható függvények Hilbert terét. Tekintsük az $A_f : \mathcal{H}_2 \rightarrow \mathcal{H}_1$ Hilbert-Schmidt integráloperátort $f(x, y)$ magfüggvénnyel, azaz legyen

$$\varphi(x) = (A_f \psi)(x) = \int_0^{T_2} f(x, y) \psi(y) dy, \quad \psi \in \mathcal{H}_2. \quad (4.11)$$

Jelölje $A_f^* : \mathcal{H}_1 \rightarrow \mathcal{H}_2$ az A_f adjungált operátort, azaz legyen

$$\psi(x) = (A_f^* \varphi)(x) = \int_0^{T_2} f(x, y) \varphi(y) dy, \quad \varphi \in \mathcal{H}_1. \quad (4.12)$$

Ekkor az $A_f^* A_f$ és $A_f A_f^*$ Hilbert-Schmidt integráloperátorok K_1 , illetve K_2 folytonos szimmetrikus magfüggvényei:

$$K_1(r_1, r_2) = \int_0^{T_1} f(z, r_1) f(z, r_2) dz, \quad 0 \leq r_1, r_2 \leq T_2 \quad (4.13)$$

és

$$K_2(s_1, s_2) = \int_0^{T_2} f(s_1, z) f(s_2, z) dz, \quad 0 \leq s_1, s_2 \leq T_1. \quad (4.14)$$

Az $A_f^* A_f$ és $A_f A_f^*$ operátorok kompakt pozitív operátorok egyazon diszkrét spektrummal. Jelölje a nemnulla sajátértékek monoton csökkenő sorozatát $\lambda_1 \geq \lambda_2 \geq \dots > 0$ és legyen φ_k , $k = 1, 2, \dots$ az $A_f^* A_f$ operátor λ_k sajátértékeihez tartozó normált sajátfüggvények sorozata. Ekkor

$$A_f^* A_f \varphi_k = \lambda_k \varphi_k, \quad k = 1, 2, \dots \quad (4.15)$$

A

$$\psi_k = A_f \varphi_k, \quad k = 1, 2, \dots \quad (4.16)$$

függvények sorozata az $A_f A_f^*$ operátor λ_k sajátértékeihez tartozó normált sajátfüggvényeit definiálja, melyre

$$A_f A_f^* \psi_k = \lambda_k \psi_k, \quad k = 1, 2, \dots \quad (4.17)$$

A Hilbert-Schmidt operátorokra fennálló eredmények szerint az $f(x, y)$ függvény megadható

$$f(x, y) = \sum_{k=1}^{\infty} \lambda_k \varphi_k(x) \psi_k(y), \quad 0 \leq x \leq T_1, \quad 0 \leq y \leq T_2 \quad (4.18)$$

alakban, továbbá a legjobb bilineáris közelítésre igaz, hogy

$$\inf_{u_k \in H_{T_1}, v_k \in H_{T_2}, 1 \leq k \leq N} \left\| f(x, y) - \sum_{k=1}^N u_k(x) v_k(y) \right\|_{L_2} = \left\| f(x, y) - \sum_{k=1}^N \lambda_k \varphi_k(x) \psi_k(y) \right\|_{L_2}. \quad (4.19)$$

A közelítés pontossága a λ_k sajátértékek aszimptotikus viselkedésétől függ, amely az f függvényre fennálló simasági feltételek mellett becsülhető [11].

Ha az $f(x, y)$ magfüggvény előállítható $f(x, y) = \sum_{k=1}^N \lambda_k \varphi_k(x) \psi_k(y)$ véges alakban, akkor az f függvényt elfajult magfüggvénynek nevezzük és ebben az esetben az A_f operátor képtere végesdimenziós Hilbert tér. Ha a λ_k sajátértékek különbözőek, akkor ez az előállítás egyértelmű és megegyezik az f függvény kanonikus előállításával.

4.3. A Fourier transzformáció és a HOSVD kapcsolata

Az előzőekben bemutatott HOSVD alapú eljárás $\tilde{w}_{n,i}(x_n)$, $x_n \in [a_n, b_n]$ ortonormált egyváltozós függvényeket alkalmaz egy n -változós sima függvény approximálására. Megfigyelhettük, hogy a $\tilde{w}_{n,i}(x_n)$ függvényeket numerikusan elő tudjuk állítani és hogy azok milyen tulajdonságokkal rendelkeznek. Összehasonlítva a bemutatott eljárást a Fourier transzformációval, hasonlóságokat figyelhetünk meg a viselkedésükben. Közismert, hogy a Fourier transzformáció trigonometrikus függvényekkel van szoros kapcsolatban, míg a HOSVD eljárás során kapott $\tilde{w}_{n,i}(x_n)$ függvények az approximálandó n -változós függvény szempontjából specifikusak. Mindkét esetben a függvények ortonormált rendszert alkotnak. Mivel HOSVD esetében specifikus függvényekről van szó, sokkal kevesebb komponensre van szükség ugyanannak az approximációs pontosságnak az elérésére mint Fourier esetben. A további összehasonlítás kedvéért említsünk meg néhány közös alkalmazást:

Fourier esetben simítást hajthatunk végre, ha a nagyobb frekvenciájú komponenseket elhagyjuk (aluláteresztő szűrés).

HOSVD esetben hasonló hatást érünk el, ha a kisebb szinguláris értékekhez tartozó polilineáris függvényeket hagyjuk el. Ugyanez a koncepció adattömörítésre is alkalmazható mind HOSVD mind pedig Fourier transzformáció esetében.

Ellenkező esetben, azaz ha az alacsonyfrekvenciájú komponenseket hagyjuk el (felülát-eresztő szűrés) élkeresőt kapunk, ami HOSVD esetében a nagyobb szinguláris értékekhez tartozó függvények elhagyását jelenti.

A nagyfrekvenciás komponensek egy küszöb alatti elhagyása szignifikáns információvesztést jelent, ami hullámok formájában jelentkezik a képen. Ezek természetesen a trigonometrikus jelleg miatt jelennek meg. A bemutatott HOSVD alapú eljárás során viszont ilyen hullámok nincsenek, sokkal kisebb az a küszöbérték, amelynél már látványos hiba (információvesztés) jelentkezik a képen.

Legyen C_n , $0 \leq C_n \leq I_n$, $n = 1..N$ az elhagyott oszlopok száma az n -edik dimenzióhoz tartozó ortonormált mátrixra vonatkozóan. A fentebb említett képtömörítésre felírható az alábbi összefüggés:

$$f(x) = \sum_{k_1=1}^{I_1-C_1} \sum_{k_2=1}^{I_2-C_2} \sum_{k_3=1}^{I_3-C_3} \alpha_{k_1,k_2,k_3} \tilde{w}_{1,k_1}(x_1) \cdot \tilde{w}_{2,k_2}(x_2) \cdot \tilde{w}_{3,k_3}(x_3). \quad (4.20)$$

4.3.1. Példák a HOSVD eljárások használatára

Az alábbi példák jól szemléltetik, hogy a javasolt megközelítésnek jó tömörítési képessége van, amely igazolja alkalmazatóságát a képfeldolgozás területén is.

Az 4.3–4.7. ábrákon megfigyelhető, hogy ugyanazon kép közelítése a HOSVD-alapú, illetve a Fourier-alapú megközelítéssel milyen különbségeket ad. Ahogy a felhasznált komponensek száma csökken, a képminőségben felfedezhető különbségek egyre szignifikánsabbá válnak. A példákban mind a HOSVD-alapú, mind pedig a Fourier-alapú megközelítésnél ugyanannyi komponenszt használtunk fel annak érdekében, hogy láthatóvá váljon a felhasznált komponensek számának hatása a kép minőségére.

A 4.7. ábrán jól megfigyelhető, hogy a Fourier-alapú eljárás használata esetén a Fourier sorok tulajdonságának megfelelően periodikus hullámok jelennek meg a képen, míg a megfelelő HOSVD alapú eljárás esetén (lásd a 4.6. ábrát) nem érzékelhetők ilyen „hibák”.

A 4.8–4.11. ábrákon látható képek mutatják a HOSVD-alapú eljárás hatékonyságát képek nagyítása esetén. Az eredmény képeket összevetjük a bilineáris és bikubikus képinterpolációs eljárások eredményeivel.



4.3. ábra. Eredeti kép (24-bites RGB)



4.4. ábra. HOSVD-alapú approximáció 7500 polineáris függvény komponens felhasználásával



4.5. ábra. Fourier-alapú approximáció 7500 trigonometrikus függvény komponens felhasználásával



4.6. ábra. HOSVD-alapú approximáció 2700 polilineáris függvény komponens felhasználásával



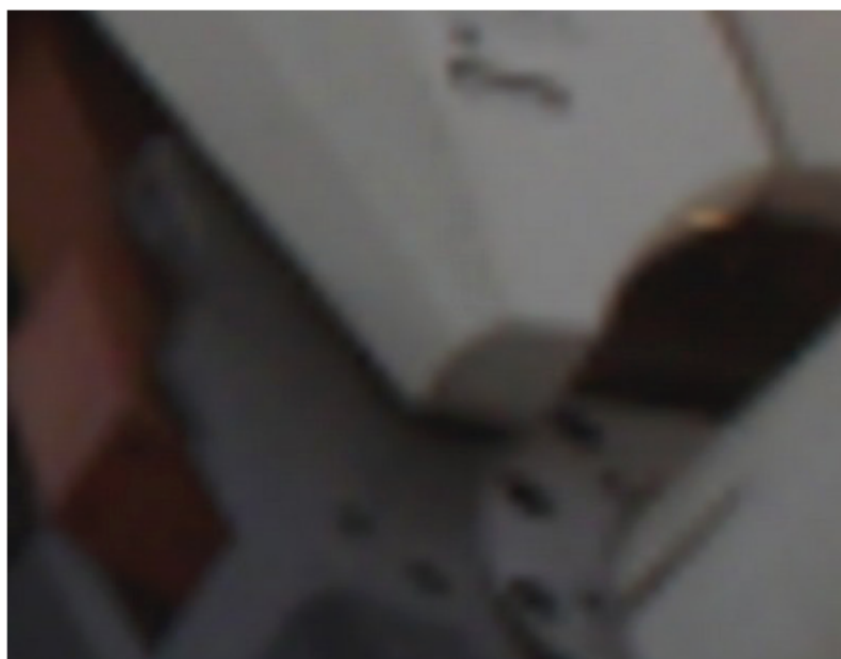
4.7. ábra. Fourier-alapú approximáció 2700 trigonometrikus függvény komponens felhasználásával



4.8. ábra. Az eredeti kép.



4.9. ábra. 10-szeres nagyítású kép bilineáris interpoláció használatával



4.10. ábra. 10-szeres nagyítású kép bikubikus interpoláció használatával



4.11. ábra. 10-szeres nagyítású kép HOSVD-alapú eljárás használatával. Jól látható, hogy az élek élesebbek, mint más eljárások esetén.

5. fejezet

Távolsági- és hasonlósági mértékek az indexek összehasonlítására

A tartalom alapú kereső rendszerek működésének hatékonysága két fontos jellemzőtől függ. Az első, hogy mennyire képes a rendszer olyan leírókat, indexeket előállítani, amelyek az adatbázisban tárolt képek tulajdonságait jól, a rendszer céljainak megfelelő módon jellemzik. A másik fontos jellemző pedig, hogy az előállított leírókat hogyan lehet egymással összemérni, és ezen összevetés alapján a valamilyen szempontból hasonló képeket megtalálni.

A disszertáció 2. fejezetében bemutattam, hogy az egyes színleírók, valamint a textúra- és alakzat leírók összehasonlítására melyek a legelterjedtebb módszerek. Jelen fejezetben be kívánom mutatni a leggyakrabban használt színleíró, a színhisztogram összehasonlítására kifejlesztett módszereket, illetve ezek javítására mutatok be egy eljárást az összehasonlításokor alkalmazott súlyozások megváltoztatásával. Az általam kifejlesztett új súlyok használatával végzett tesztek is ismertetem, melyek igazolják a hatékonyabb működést.

5.1. Irodalmi áttekintés

A $\{h_i\}$ hisztogram egy leképezés a N -dimenziós i egész elemű vektorok halmazáról a nem-negatív valós számok halmazára [83]. Szürkeárnyalatos képek esetén N az intenzitás értékek kvantálása után előálló vödorszámot jelöli, a h hisztogram i indexű értéke pedig megadja, hogy az i -edik intenzitás vödörbe eső képbeli intenzitásértékeknek milyen a relatív gyakorisága. Az irodalomban néhány esetben megkülönböztetik a szürkeárnyalatos hisztogramot és a normalizált hisztogramot, melyek közötti különbség, hogy első esetben az egyes vödörbe eső intenzitás értékek gyakoriságát, míg második esetben ezek relatív gyakoriságát (képmérettel normált

értékét) vesszük figyelembe. Mivel két hisztogram összehasonlítása csak normálást követően lehetséges, ezért értelemszerűen minden esetben a normalizált hisztogramot értem hisztogram alatt.

Színes képek hisztogramjai is könnyen értelmezhetőek oly módon, mint egy szürkeárnyaltos kép hisztogramja, ahogy ezt az 2. fejezetben bemutattam.

Számos távolság-, illetve hasonlósági mérték került bevezetésre a hasonlóság mérésére két hisztogram, pl. $H = \{h_i\}$ és $K = \{k_i\}$ között, melyeket röviden bemutatok az alábbiakban.

A leggyakrabban alkalmazott távolság az ún. Minkowski-féle távolság [56]:

$$d_{L_r}(H, K) = \left(\sum_{i=1}^N |h(i) - k(i)|^r \right)^{1/r}, \quad (5.1)$$

ahol r általában 1, 2, vagy $+\infty$. Könnyen belátható, hogy $r=2$ esetén az euklideszi távolsággal egyezik meg a Minkowski távolság. A L_1 norma használatakor a

$$d_{L_1}(H, K) = \sum_{i=1}^N |h(i) - k(i)| \quad (5.2)$$

távolságot, az L_∞ használatakor pedig a

$$d_{L_\infty}(H, K) = \max \{|h(i) - k(i)|\}_{i=1,2,\dots,N} \quad (5.3)$$

távolságot kapjuk.

Az 5.1. ábrán látható egy keresés eredménye, ahol hisztogramokat az L_1 norma használataival hasonlítottunk össze. Hasonló eredmények az L_2 norma használataival a 5.2., az L_∞ normával pedig a 5.3. ábrán láthatóak.

Swain és Ballard értelmezte a hisztogram-metszet [111] mértéket, mely az első CBIR rendszerekben széles körben alkalmazott módszer volt.

$$d_\cap(H, K) = 1 - \frac{\sum_{i=1}^N \min(h(i), k(i))}{\sum_{i=1}^N k(i)} \quad (5.4)$$

Gyakran alkalmazott módszerek még az alábbi hasonlósági mértékek.

– Kullback-Leibler divergencia [54]:

$$d_{KL}(H, K) = \sum_{i=1}^N h(i) \log \frac{h(i)}{k(i)} \quad (5.5)$$



5.1. ábra. A bal felső sarokban lévő képhez hasonló képek egy képi adatbázisból. A keresésnél színhisztogramokat hasonlítottunk össze az L_1 normájú Minkowski távolság használatával.



5.2. ábra. A bal felső sarokban lévő képhez hasonló képek egy képi adatbázisból. A keresésnél színhisztogramokat hasonlítottunk össze az L_2 normájú Minkowski távolság használatával.



5.3. ábra. A bal felső sarokban lévő képhez hasonló képek egy képi adatbázisból. A keresésnél színhisztogramokat hasonlítottunk össze az L_∞ normájú Minkowski távolság használatával.

– Jeffrey divergencia [81]:

$$d_J(H, K) = \sum_{i=1}^N \left(h(i) \log \frac{h(i)}{m(i)} + k(i) \log \frac{k(i)}{m(i)} \right), \quad (5.6)$$

ahol $m(i) = \frac{h(i)+k(i)}{2}$.

– Illeszkedési távolság [101]:

$$d_M(H, K) = \sum_{i=1}^N |\bar{h}(i) - \bar{k}(i)|, \quad (5.7)$$

ahol $\bar{h}(i) = \sum_{j \leq i} h(j)$ a $\{h(i)\}$ kumulált hisztogramja, és hasonlóan értelmezett $\bar{k}(i)$ is.

– Kolmogorov-Smirnov távolság:

$$d_{KS}(H, K) = \max_{i=1}^N (|\bar{h}(i) - \bar{k}(i)|) \quad (5.8)$$

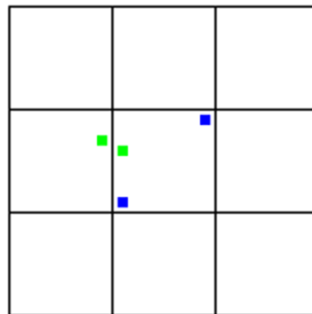
Elterjedt távolsági mérték még a kvadratikus formátumú távolság [73]:

$$d_A(H, K) = \sqrt{(\mathbf{h} - \mathbf{k})^T \mathbf{A} (\mathbf{h} - \mathbf{k})}, \quad (5.9)$$

ahol \mathbf{h} és \mathbf{k} a hisztogramok vektori alakjai. A vödrök közötti távolságok súlyozása az \mathbf{A} mátrixban jelenik meg. Vegyük észre, hogy súlymátrixként az egységmátrixot választva pont az euklideszi távolságot eredményezi a kvadratikus távolság. Az alábbiakban bemutatom, hogy a súlyokat milyen módon lehet meghatározni annak érdekében, hogy minél hatékonyabb keresést tudjunk megvalósítani.

5.2. Újfajta távolság értelmezése

A kvadratikus távolságnál értelmezett súlymátrix használatát többek között az is indokolja, hogy a hisztogramvödrök meghatározásakor végrehajtott kvantálás egyik következményeként előállhat olyan eset, hogy két egymáshoz nagyon közeli szín más hisztogram vödörbe esik, így Minkowski távolság használatakor nem hasonlítjuk össze őket, míg két egymástól távolabb található szín ugyanabba a hisztogram vödörbe kerülhet. Ezt szemlélteti a 5.4. ábra. A zöld pontokkal jelölt színintenzitások közelebb vannak egymáshoz, mint a kék pontokkal jelölt színintenzitás értékek, mégis csak az utóbbiak esnek azonos hisztogram vödörbe.



5.4. ábra. Közeli színek különböző hisztogram vödrökbe, míg távolabbi színek azonos hisztogram vödrökbe eshetnek.

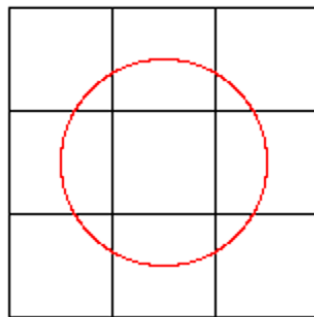
Az előző fejezetben említett kvadratikus formátumú távolság esetén az $\mathbf{A} = [a_{ij}]$ súlymátrix meghatározására a [10] cikkben a következő eljárást használják: Két hisztogram összetartozó vödre esetén a súly legyen 1, szomszédos vödröknél 0.5, egyéb esetben pedig 0.

Ennek az eljárásnak hátránya, hogy a szomszédos vödrök esetén nem veszi figyelembe, hogy azok egy háromdimenziós hisztogram esetén milyen szomszédok. Képzeljük el ugyanis, hogy $h(i, j, k)$ szomszédja $h(i - 1, j, k)$, $h(i - 1, j - 1, k)$ valamint $h(i - 1, j - 1, k - 1)$ is. Nem célszerű viszont mindhárom esetben ugyanazt a súlyt alkalmazni, hiszen a hisztogram adott vödre által reprezentált színek a legközelebb $h(i, j, k)$ és $h(i - 1, j, k)$ esetén lesznek egymáshoz, míg legtávolabb $h(i, j, k)$ és $h(i - 1, j - 1, k - 1)$ esetén.

Az eljárás azonban javítható a következők szerint. Képzeljük el, hogy a háromdimenziós térben a hisztogram vödrök minden irányban azonos szélességűek. Tekintsük ezt a szélességet

egy egységnek. Az éppen vizsgált hisztogram vödör $(h(i, j, k))$ középpontjából rajzoljunk fel képzeletben egy egységnyi sugarú gömböt. Vizsgáljuk meg, hogy ez a gömb milyen mértékig metsz bele a szomszédos hisztogram vödrökbe és ezek arányát használjuk a kvadrátikus távolság súlyozására. Az alábbi eredményeket fogjuk kapni négy tizedesre kerekítve, ha a középelem (vödör) súlyát egynek tekintjük (ld. a 5.5. ábrát):

- Középelem súlya: 1
- Lapközepont súlya: 0.2055
- Élközéppont súlya: 0.0147
- Sarokelem súlya: 0.0002



5.5. ábra. Hisztogram vödör középpontjától egy vödör szélességnyire lévő intenzitások halmaza két dimenzióban ábrázolva.

Ennek megfelelően az u és a v kép RGB hisztogramjának távolsága a következő módon alakul:

$$\begin{aligned}
 d(h_u, h_v) = & \sum_{r=1}^N \sum_{g=1}^N \sum_{b=1}^N [(h_u(r, g, b) - h_v(r, g, b))^2 + \\
 & + (0.2055 * (h_u(r-1, g, b) - h_v(r-1, g, b)))^2 + \dots + \\
 & + (0.0147 * (h_u(r-1, g-1, b) - h_v(r-1, g-1, b)))^2 + \dots + \\
 & + (0.0002 * (h_u(r-1, g-1, b-1) - h_v(r-1, g-1, b-1)))^2 + \dots]
 \end{aligned}$$

A távolságra érdekesebb viszont a kvadratikusság bevezetésénél használt 5.9 képletet alkalmazni, amihez szükséges az \mathbf{A} mátrix meghatározása. Ennek felépítéséhez azt kell figyelembe venni, hogy a háromdimenziós $h(i, j, k)$ hisztogram hogyan alakítható át egydimenziós $h'(l)$ hisztogramá. Ehhez az

$$l = (N - 1)^2 \cdot i + (N - 1) \cdot j + k \quad (5.10)$$

összefüggés szükséges, ahol N jelöli minden egyes színsatorna vödreinek számát. A jelölésrendszerrel figyelembe vettük, hogy $i, j, k \in \{1, \dots, N\}$, és úgy $l \in \{1, \dots, N^3\}$. Ez alapján viszont az \mathbf{A} súlymátrix mérete $N^3 \times N^3$ lesz, amivel igen lassúak lennének az egyes távolságok kiszámításai a

$$d(H'_u, H'_v) = \sqrt{(\mathbf{h}'_u - \mathbf{h}'_v)^T \mathbf{A} (\mathbf{h}'_u - \mathbf{h}'_v)} \quad (5.11)$$

képlet alkalmazásával.

Ezen könnyíthet, ha \mathbf{A} mátrixon végrehajtunk egy szinguláris érték dekompozíciót.

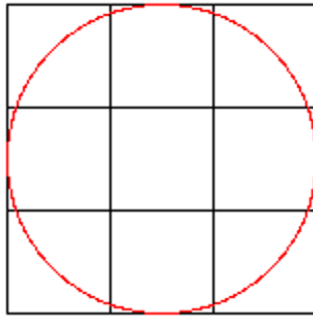
Hasonlóan megvizsgálhatjuk, hogy milyen súlyokat eredményez, ha az elképzelt gömb sugara 1.5 egység (ld. a 5.6. ábrát):

- Középlelem súlya: 1
- Lapközéppont súlya: 0.9430
- Élközéppont súlya: 0.5089
- Sarokelem súlya: 0.1718

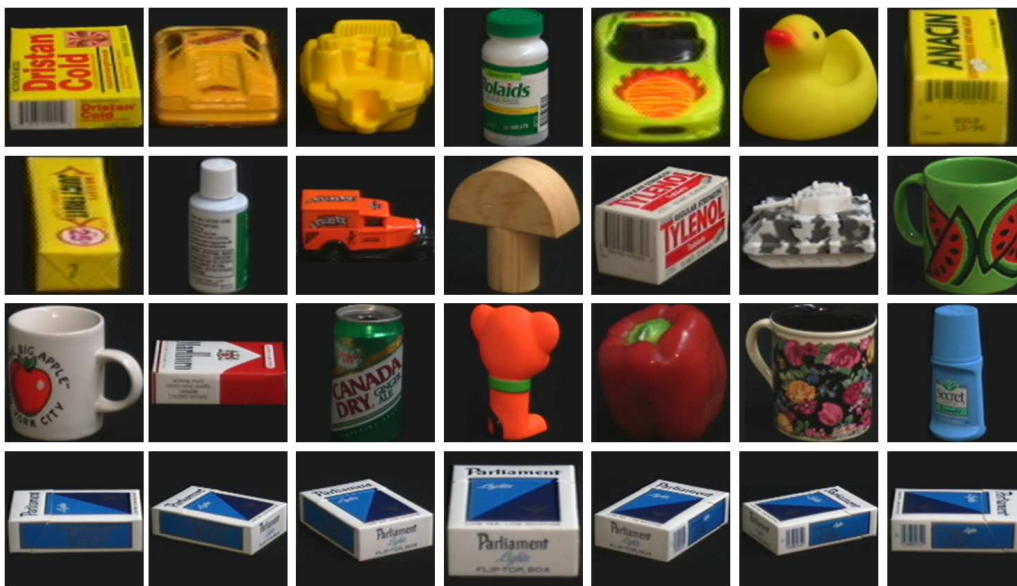
5.3. Kísérletek

A kísérleteket a Columbia Object Image Library (COIL-100) képi adatbázis [72] képeivel végeztem el. Ebben az adatbázisban 100 különböző objektumról készült képek találhatók, melyek homogén sötét háttérben készültek. Minden objektumról 72 különböző felvétel készül, melyek az objektumot körbejárva 5° -kal eltérő irányokból készültek azonos megvilágítási viszonyok mellett. Az adatbázis néhány képe példaként a 5.7. ábrán látható.

A kísérleteket úgy végeztük el, hogy minden objektumról kiválasztottunk 12 különböző felvételt véletlenszerűen. Így kaptunk egy 1200 képet tartalmazó adatbázist. Ebből kiválasztottunk minden objektum esetén egy-egy képet, majd vizsgáltuk, hogy a különböző távolságok



5.6. ábra. Hisztogram vödör középpontjától másfél vödör szélességre lévő intenzitások halmaza két dimenzióban ábrázolva.



5.7. ábra. A Columbia Object Image Library (COIL-100) néhány mintaképe.

használata esetén a kereséshez használt képen található objektummal azonos objektumot tartalmazó képek hányadik helyen vannak a távolsági rangsorban. Vizsgáltuk a minimális, az átlagos és a maximális távolságát ezeknek a képeknek.

A 5.1. táblázatban összefoglaljuk, hogy 1200 képre elvégezve a kísérletünket, hány esetben adott jobb, illetve rosszabb eredményt a súlyozást alkalmazó algoritmus, mint a súly nélküli kvadratikus távolság.

	Súlyozott jobb	Azonos	Súlyozás nélküli jobb
Minimális távolság	49	1108	43
Távolságok mediánja	338	617	245
Távolságok átlaga	558	301	341
Maximális távolság	518	350	332

5.1. táblázat. A súlyozott és a súlyozás nélküli kvadratikus távolsággal való keresés eredményeinek összehasonlítása.

A 5.2. táblázat megmutatja, hogy a távolságok átlaga mennyi.

	Súlyozott	Súlyozás nélküli
Minimális távolság	2.1533	2.1792
Távolságok mediánja	58.2933	61.5633
Távolságok átlaga	90.4008	94.2705
Maximális távolság	317.8300	327.9233

5.2. táblázat. A súlyozott és a súlyozás nélküli kvadratikus távolságok statisztikai jellemzői.

A 5.3. táblázatban összefoglaljuk, hogy 1200 képre elvégezve a kísérletünket, hány esetben adott jobb, illetve rosszabb eredményt a súlyozást alkalmazó algoritmus, mint a más súlyt használó távolság.

A 5.4. táblázat megmutatja, hogy a távolságok átlaga mennyi.

A bemutatott eredmények igazolják, hogy szín alapján hasonló képek keresése esetén érdemes a Minkowski távolság helyett súlyozott kvadratikus távolságot alkalmazni, a súlyok meghatározásánál pedig érdemes figyelembe venni, hogy a vizsgált hisztogram vödrök elhelyezkedése milyen egymáshoz képest.

	Súlyozott jobb	Azonos	Más súly jobb
Minimális távolság	77	1073	50
Távolságok mediánja	357	564	279
Távolságok átlaga	505	278	417
Maximális távolság	460	325	415

5.3. táblázat. Két különböző súlyozású kvadratus távolsággal való keresés eredményeinek összehasonlítása.

	Súlyozott	Más súly
Minimális távolság	2.1533	2.1942
Távolságok mediánja	58.2933	55.5358
Távolságok átlaga	90.4008	87.7430
Maximális távolság	317.8300	310.3650

5.4. táblázat. Két különböző súlyozású kvadratus távolságok statisztikai jellemzői.

6. fejezet

Skicc alapú kereső rendszer

A képen található információk közül sok esetben nem a színek, azok eloszlása, vagy a textúra az igazán fontos a felhasználó számára, hanem a képen található objektumok alakja. Amennyiben alakzat alapján szeretnénk keresést végrehajtani egy képi adatbázisban az alakot valamilyen módon reprezentálni kell. Erre egy gyakran használt lehetőség az, hogy a felhasználó felvázol egy skiccet a kereső felületen, majd ezen skicchez hasonló alakú határvonalakkal rendelkező objektumot tartalmazó képeket keresünk az adatbázisban.

Skicc alapú kereső rendszerek használata nagyon fontos és hasznos lehet az élet több területén. Sok esetben a gondolatainkat legjobban rajzok, ábrák segítségével tudjuk ugyanis kifejezni. A következőkben bemutatok pár területet, ahol skicc alapú kereső rendszerek használata indokolt lehet.

A tartalom alapú kereső rendszereknek nagy jelentősége van a rendőrségi nyomozások területén. Fantom képek, tetoválások és graffitik azonosítása hasznos lehet ezen eljárások lefolytatása esetén. Hasonló azonosító rendszerek, melyek skiccek azonosításán alapulnak már kifejlesztésre kerültek [44, 46, 45].

A skicc alapú kereső rendszerek másik felhasználási lehetősége az analóg áramköri rajzok nagy adatbázisában való keresés [38]. A felhasználó elkészíti egy analóg áramkör vázlatát, majd a rendszer megtalálja a rajzhoz leginkább hasonló áramköri tervrajzokat az adatbázisból.

A skicc alapú keresés először a QBIC [32] és a VisualSEEK [103] rendszerekben jelentek meg. Ezekben a rendszerekben a felhasználó színes vázlatokat és foltokat rajzolhat a rajz felületre. A képek több területre vannak felosztva, és a szín valamint textúra jellemzők ezen területekre vannak meghatározva. A képek felosztásának módszere más algoritmusokra is jellemző, például az élhisztogram leíró (EHD) módszerre [25]. Hátránya ezeknek a rendszereknek, hogy nem invariánsak a forgatásra, átméretezésre és eltolásra. Később a bonyolult és robosz-

tus leírók fejlesztése vált meghatározóvá. Más kutatási megközelítés a fuzzy logika és neurális hálózat alapú eljárások használata. Ezekben az esetekben a fejlesztés fő célja a képjellemzők megfelelő súlyozásának meghatározása [63].

6.1. Kifejlesztett rendszer

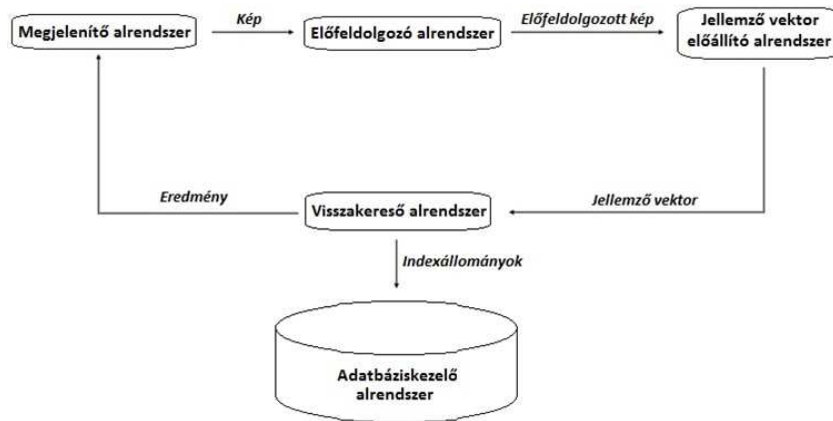
Ebben a fejezetben ismertetem kifejlesztett rendszerünk célját és általános felépítését. Bemutatom az egyes komponenseket és azok egymás közötti kommunikációját, valamint az egyes alrendszerek funkcióit és a felhasznált algoritmusokat.

6.1.1. A rendszer célja

Bár a skicc alapú kereső rendszerek (SBIR) kutatása nagy mértékben növekszik, jelenleg még nem létezik széles körben használható SBIR rendszer. Célunk az volt, hogy kifejlesszünk egy olyan tartalom alapú asszociatív kereső motort, amely létező képi adatbázisokból kinyeri a felhasználó által felvázolt rajzhoz hasonló alakzatokat tartalmazó képeket. Ehhez a felhasználó számára rendelkezésre áll egy rajz felület, ahol alakzatokat és momentuomokat vázolhat fel, a rendszer pedig az alakzat az elhelyezkedés és méret információk figyelembe vételével keresi meg a rajzhoz hasonló képeket. A kinyert eredmény képeket pedig szín alapján tovább rendezzük, így különítve el egymástól a különböző típusú eredmény képeket. Legfőbb feladatunk volt, hogy áthidaljuk a szabadkézi rajzok és a digitális képek közötti információs szakadékot, amelyet saját előfeldolgozó algoritmusunkkal valósítottunk meg. A kifejlesztett rendszerünkben visszacsatolási lehetőség is van, melynek segítségével a felhasználó igényeihez még jobban illeszkedő képeket tudunk eredményként szolgáltatni.

6.1.2. Rendszerünk általános felépítése

A rendszet építőkövei közül első az előfeldolgozó alrendszer, amely a képek diverzitása által okozott problémákat feloldását valósítja meg. A jellemző vektor generáló alrendszer minden egyes képhez hozzárendel egy numerikus leíró képből kinyert adott tulajdonságok figyelembe vételével. Az adatbáziskezelő alrendszer egy interfészt valósít meg a programunk és az adatbázis között. A visszakereső alrendszer a jellemző vektorok és a minta képek felhasználásával a legjobb találatokat tartalmazó képlistát ad át a megjelenítő alrendszernek. A rendszer globális felépítése az 6.1. ábrán látható.



6.1. ábra. A rendszer globális felépítése

A tartalom alapú visszakeresés módszere két fő részre osztható. Az első az adatbázis építésének fázisa, amelyben az előfeldolgozott képek eltárolása történik a kinyert jellemző vektorok formájában. Ezt a fázist tekinthetjük a program off-line részének. Ez a rész foglalja magában a leginkább számítás igényes feladatokat, amelyeket a program aktuális használata előtt kell elvégezni. A másik fázis a visszakeresési eljárás, amely az on-line egysége a programnak.



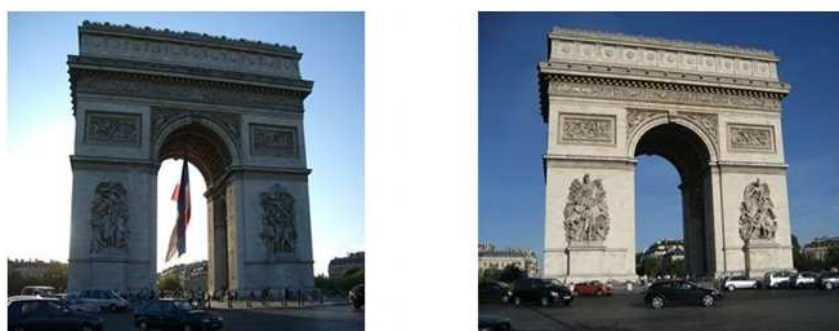
6.2. ábra. A rendszer folyamatábrája a felhasználó szemszögéből

Tekintsük át a rendszer folyamatábráját a felhasználó szemszögéből (lásd a 6.2. ábrát). Először a felhasználó rajzol egy skiccet, vagy betölt egy már korábban elkészített vázlatrajzot. Ezt követően a visszakeresési eljárás elindul. A vizsgált képen képen először végrehajtunk egy előfeldolgozást, majd a jellemző vektorok elkészítése történik meg. Ezt követően a jellemző vektorokat hasonlítjuk össze az adatbázisban tárolt képek már korábban legenerált jellemző vektoraival. A legjobb találatok megjelenítésre kerülnek a felhasználói felületen színek alapján csoportosított formában. Ezt követően még az eredmények között a felhasználó megjelölheti, hogy melyik képet a leginkább megfelelőnek és a rendszer ehhez hasonlókat keres az adatbázisban.

zisból.

6.1.3. Az előfeldolgozó alrendszer

Rendszerünket alapvetően viszonylag egyszerű képeket tartalmazó adatbázisban való keresésre terveztük, de még ebben az esetben is nagy mértékű különbségek adódhatnak a képek méretében, felbontásában, stb. Emellett még a képek zajosak lehetnek, valamint az egyes képek megvilágításának mértéke és iránya is különböző lehet (lásd a 6.3. ábrát), és így a jellemző vektorok hatékony összehasonlítása nem lehetséges. Ahhoz, hogy kiküszöböljük ezeket a problémákat egy többlépéses előfeldolgozó mechanizmust dolgoztunk ki.



6.3. ábra. Megvilágítási és nézőpontbeli különbségek azonos tárgyról készült két képen

Az előfeldolgozó alrendszer bemenete egy kép, kimenete pedig ennek a képnek egy megfelelően átalakított változata. Az egyes lépések a 6.4. ábrán láthatók.



6.4. ábra. Az előfeldolgozó alrendszer egymást követő eljárásai

Első lépésként a képeket azonos méretűre alakítjuk át. Második lépésként a fényviszonyok kiegyenlítése érdekében histogram kiegyenlítést végzünk, mely eljárás során a kép szürkárnyalatos színhistogramját úgy transzformáljuk, hogy a lehető legjobb mértékben közelítse az egyenletes eloszlás sűrűségfüggvényét [129]. Ezt követően csökkentjük a képen található színek

számát az uniform és minimum variancia szerinti kvantálás alkalmazásával [125], így redukáljuk a kép textúrázottságát, aminek hatására a nem releváns élek jelentős hányada eltűnik a képről. Negyedi lépésként éldetektálást hajtunk végre a képen, így az hasonlónak válik a felhasználó által készített, alapvetően éleket tartalmazó vonalrajzhoz. Éldetektálásként a Canny-féle éldetektáló módszert [8] alkalmaztuk. Az éldetektálást követően morfológiai nyitással eltüntettük a rövid élszakaszokat, mert általában az objektumokat határoló, rendszerünk számára releváns élszakaszok a hosszú élszakaszok. Utolsó lépésként egy távolság transzformációt [26] hajtunk végre, így minden pixelhez hozzárendeljük azt a számot, amely a hozzá legközelebb eső, nem-nulla értékű pixel távolságát mutatja meg az aktuális pixelhez képest.

6.1.4. A jellemző vektor előállító alrendszer

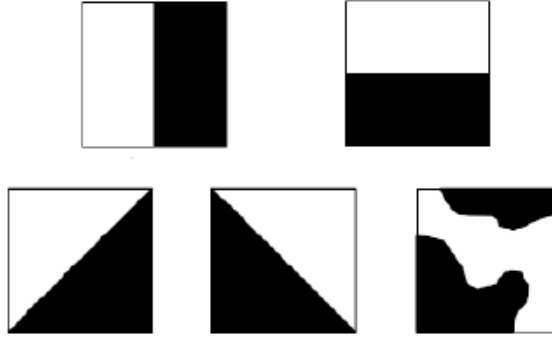
Ez az alrendszer állítja elő az egyes képekhez tartozó jellemző vektorok, amelyek a képeken található releváns információkat tartalmazzák. Alapvetően három különböző eljárást használtunk fel, az élhisztogram leíró (EHD – edge histogram descriptor) [25], az irányított gradiensek hisztogramját (HOG – histogram of oriented gradients) [21] és a skála invariáns jellemző transzformációt (SIFT – scale invariant feature transform) [62].

Élhisztogram leíró

Az MPEG leírók között a textúra leíróknál találjuk meg az élhisztogram leíró (*EHD – Edge Histogram Descriptor*), mely a képen található lokális élek irányultsága alapján épít hisztogramot.

A módszer első lépésében a képet 4×4 alképre osztjuk fel. A hisztogram előállítása során az alképeken található éleket öt-öt osztályba soroljuk az irányultságai szerint. A használt irányultságok: függőleges, vízszintes, 45° -os átlós, 135° -os átlós és konkrét iránnyal nem rendelkező élek csoportjába (lásd a 6.5. ábrát). A képet tizenhat alképre osztottuk, az egyes alképeken található pixelek öt-öt állapotot vehetnek fel, így nyolcvan hisztogram vödörre van szükségünk.

Az élek osztályozása céljából tovább finomíthatjuk az kép felosztását. Az egyes alképeket ún. nem átfedő, azonos méretű és kettővel osztható szélességű és magasságú, négyzet alakú képblokkokra osztjuk fel. A blokkok méretét a kép felbontásának függvényében választjuk ki. Minden képblokkot élkategóriákba sorolunk, a fentebb már említett csoportosítási szempont szerint. Az osztályozás elvégzése érdekében az egyes képblokkokat 2×2 -es méretű szuperpixelnek tekintjük. Az egyes szuperpixel értékek a képblokk adott sarkában lévő pixelek átlagaként állnak elő. A megfelelő élerősségek meghatározása érdekében lineáris szűrést hajtunk



6.5. ábra. Az egyes élosztályok reprezentációja.

vége.

Az élek kategorizálásának bemutatása érdekében bevezetjük az alábbi jelöléseket. Az $a_0(i, j)$, $a_1(i, j)$, $a_2(i, j)$ és $a_3(i, j)$ jelentse az i -edik sorban és j -edik blokkban található 2×2 -es szuperpixel intenzitás értékét. Az alkalmazott szűrők együtthatóinak értékeit az egyes irányok és pozíciók esetén jelölje $f_v(k)$, $f_h(k)$, $f_{d-45}(k)$, $f_{d-135}(k)$ és $f_{nd}(k)$, ahol $k = 0, 1, 2, 3$, a szuperpixelen belüli pozíciót adja meg. Az él nagyságát jelölje $m_v(i, j)$, $m_h(i, j)$, $m_{d-45}(i, j)$, $m_{d-135}(i, j)$ és $m_{nd}(i, j)$, ahol i és j azonos jelentésű mint az $a_k(i, j)$ -knél. Az egyes m értékek kiszámítása az alábbi módon történik:

$$m_v(i, j) = \left| \sum_{k=0}^3 a_k(i, j) \cdot f_v(k) \right| \quad (6.1)$$

$$m_h(i, j) = \left| \sum_{k=0}^3 a_k(i, j) \cdot f_h(k) \right| \quad (6.2)$$

$$m_{d-45}(i, j) = \left| \sum_{k=0}^3 a_k(i, j) \cdot f_{d-45}(k) \right| \quad (6.3)$$

$$m_{d-135}(i, j) = \left| \sum_{k=0}^3 a_k(i, j) \cdot f_{d-135}(k) \right| \quad (6.4)$$

$$m_{nd}(i, j) = \left| \sum_{k=0}^3 a_k(i, j) \cdot f_{nd}(k) \right| \quad (6.5)$$

Ha $m(i, j) = \max(m_v(i, j), m_h(i, j), m_{d-45}(i, j), m_{d-135}(i, j), m_{nd}(i, j))$ értéke nagyobb egy küszöbnél, akkor az adott blokk tartalmaz élt, ellenkező esetben pedig úgy tekintünk rá, hogy nem. A 6.6. ábrán láthatóak az alkalmazott szűrőmaszkok.

A nyolcvan vödörös élhisztogram alapján még nem érdemes döntést hozni, mivel az csak főként globális tulajdonságokat vesz figyelembe. Fontos, hogy a meglévő hisztogramot ki kell

$f_v(1) = 1$	$f_v(1) = -1$	$f_{d-45}(0) = \sqrt{2}$	$f_{d-45}(1) = 0$
$f_v(2) = 1$	$f_v(3) = -1$	$f_{d-45}(2) = 0$	$f_{d-45}(3) = -\sqrt{2}$

vertikális élekre szűrő maszk 45°-os diagonális élekre szűrő maszk

$f_h(0) = 1$	$f_h(1) = 1$	$f_{d-135}(0) = 0$	$f_{d-135}(1) = \sqrt{2}$
$f_h(2) = -1$	$f_h(3) = -1$	$f_{d-135}(2) = -\sqrt{2}$	$f_{d-135}(3) = 0$

horizontális élekre szűrő maszk 135°-os diagonális élekre szűrő maszk

$f_{nd}(0) = 2$	$f_{nd}(1) = -2$
$f_{nd}(2) = -2$	$f_{nd}(3) = 2$

konkrét iránnyal nem rendelkező élekre szűrő maszk

6.6. ábra. A különböző szűrőmaszkok a különböző élırányokra érzékenyek, így segítenek hozzá az osztályozáshoz.

bővíteni bizonyos lokális felbontások alapján kinyert információkkal. Egy heurisztikus megoldás az, ha a 6.7. ábrán szemléltetett felosztásokra előállított hisztogram vödrök és a globális kép összevont értékeit is bele vesszük az eredeti hisztogramba, így kapunk egy 150 értékből álló leíró ($16 \cdot 5 + 13 \cdot 5 + 5$) [130].



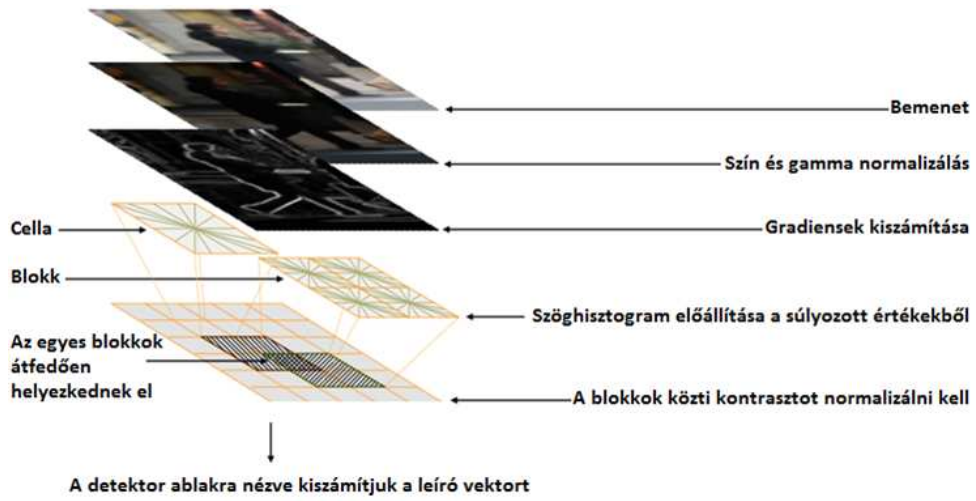
6.7. ábra. A lokális tulajdonságok kiemelésére szolgáló további felosztások.

Természetesen az élhisztogram leírók összehasonlítása előtt az egyes hisztogram vödröket normalizálni kell. Normalizálás céljából az élek előfordulásának számát elosztjuk az alképen található képblokkok számával. Meg kell jegyezni, hogy a homogén területeket is számításba vesszük a normalizálás esetén, viszont a hisztogramban nem jelennek meg, így az élhisztogram értékeit azok a területek is befolyásolják, amelyek nem is tartalmaznak éleket.

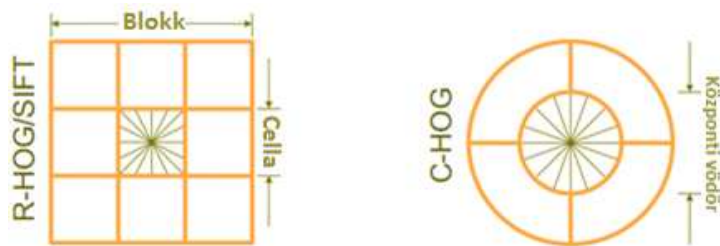
Irányított gradiensek hisztogramja

Az irányított gradiensek hisztogramja (*HOG – Histogram of Oriented Gradients*) egy olyan leíró, mely az EHD módszer továbbfejlesztésének is tekinthető [21]. Az alapötlet abból fakad, hogy régiókra bontjuk a képet, majd a régiókhoz olyan irányhisztogramokat rendelünk, melyek vödreit az egyes gradiensek nagyságával vannak súlyozva. A részletes mechanizmust a 6.8. ábra szemlélteti.

Az algoritmus első lépéseként végre kell hajtani egy gamma szűrést, illetve színnormalizálást. Második lépésként éldetektálást hajtunk végre. A harmadik lépésben felosztjuk a képet – az EHD módszertől eltérő módon – átfedő blokkokra, azokat pedig cellákra. A cellákhoz egy hisztogramot rendelünk, melyet úgy állítunk elő, hogy a pixelekhez kiszámítjuk a gradiens értékeket, és az adott irányú vödröket nem egyesével növeljük, hanem a gradiens nagyságával súlyozottan. Az alkalmazott vödrök a $0^\circ - 180^\circ$ tartományba esnek 20 fokoskénti eltéréssel. Az említett cellák lehetnek négyszögletesek, illetve kör alakúak is (lásd a 6.9. ábrát). A negyedik lépésben normalizálást kell végrehajtani. A gradiens nagyságok széles skálán mozoghatnak, melyben szerepet játszik a megvilágítás, illetve az előtér és háttér kontrasztja. Ahhoz, hogy azonos léptékkal kezelhessük a különböző nagyságrendű értékeket elengedhetetlen a közös nevező



6.8. ábra. A HOG jellemző előállításának szintjei.



6.9. ábra. A HOG-nál használt blokk típusok.

létrehozása. A normalizálási faktort az L2-norma használatával az alábbi módon határozhatjuk meg [110]:

$$nf = \frac{V}{\sqrt{\|V\|_2^2 + \varepsilon^2}}, \quad (6.6)$$

ahol nf a normalizálási faktor, V egy adott blokk hisztogramjai, ε pedig egy nullához közeli konstans a nullával való osztás kiküszöbölésére.

Ezt követően elő kell állítani a jellemző vektort az úgynevezett detektor ablakra nézve, amelyben feltételezhetően előfordul a keresett objektum. Ez egy a kép méreténél általában kisebb, a kép egy részét lefedő képrészlet. A végső vektor mérete függ a paraméterezéstől. Megadhatjuk a cella méretét, ami tulajdonképpen az egy cellába eső pixelek száma. A blokkok tulajdonságait is lehet befolyásolni, úgymint méret, átfedés mértéke, normalizálás módja. Végül a hisztogram paramétereit is beállíthatjuk, pl. vödrök száma, előjeles értékek használata, súlyozás módja. A különféle paraméterek értékei függenek a konkrét megoldandó feladat jellegétől.

A HOG módszert általában egy adott alakú objektum detektálására szokás használni valamilyen lineáris osztályozóval – általában SVM [40] – kiegészítve. Az általunk használt képeknél általában egyetlen objektum szerepel a képen, így esetünkben a detektor ablakmérete megegyezik a kép méretével.

Skála invariáns jellemző transzformáció

A SIFT jellemző leírót (*SIFT – Scale Invariant Feature Transform*) David Lowe [63] fejlesztette ki. Ez a leíró algoritmus a számítógépes látás jól definiált módszere, mely a képtartalomra nézve eltolás-, elfordulás-, skála- és megvilágítás invariáns.

A transzformáció első lépésében Gauss-féle konvolúciós szűrővel hajtunk végre simítást. A simítást többször is elvégezzük, a kép méretét mindig felére csökkentve. Így a képpiramist hozunk létre, mely a feldolgozás gyorsítását szolgálja. Ezt követően a szomszédos konvolúciós szintek különbségét képezzük, így egy különbség skála szintjei jönnek létre. Ennek leírására használjuk a $D(x, y, \sigma)$ függvényt:

$$D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) = L(x, y, k\sigma) - L(x, y, \sigma), \quad (6.7)$$

ahol $*$ a konvolúciós operátor, $G(x, y, \sigma)$ a Gauss-féle konvolúciós maszk, $I(x, y)$ a bemeneti kép, L a skálatér egy szintje, k pedig egy pozitív egész, mely a skálaszintet jelöli. σ az alkalmazott Gauss függvény szórását jelöli. Leegyszerűsítve D olyan különbség szinteket képez, melyben egyik szint éppen k -szor feljebb található a skála-térben, mint a kivonandó másik. Ezzel

a lépéssel biztosítható, hogy hatékonyan találjunk „biztos” pontokat és megtartsuk a módszer „erejét” adó invarianciát.

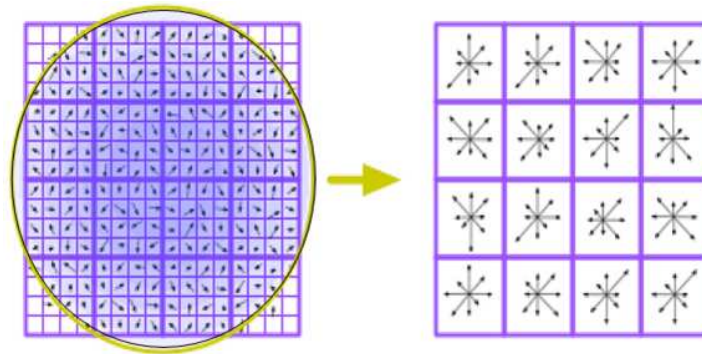
A következő lépésben szélsőértékeket kell keresni a D által létrehozott Gauss-különbség térben. A kulcspontok a lokális maximum, illetve minimum értékek lesznek, melyek az aktuális pixel saját szintjén elhelyezkedő nyolcas környezetével, illetve a szomszédos szinteken lévő 18 másik szomszédjával való összehasonlításból számítandóak ki. Ez így összesen huzsonhat összehasonlítás. Ha ez a pont a többivel való összehasonlítások után minimális vagy maximális, akkor szélsőérték. Az így kapott pontokon további műveleteket kell végezni. Mivel többször is elmostuk a képeket és a méretüket is csökkentettük, ezért interpoláció segítségével a környező adatok alapján vissza kell keresnünk a kijelölt pontok eredeti helyét az eredeti képen. Ha ez megvan, akkor a hatékonyság növelése érdekében csökkenteni kell a kulcspontok számát, ugyanis ezek közül még nem mind hordoz fontosnak mondható információt. Ez azért szükséges, hogy csak a stabil pontok maradjanak meg a további számításokhoz. Az alacsony kontrasztú pontokat és a gyenge élpontokat el kell távolítani egy küszöböléssel. Ezt megtehetjük könnyedén, ha kiszámítjuk a Laplace operátorral számított értékét az adott pontnak. Ha a kontraszt érték egy megadott küszöb alatt van, akkor kivesszük a pontot a kulcspont listából. A stabilitáshoz azonban nem elegendő a kis kontrasztú pontok kiszűrése. A Gauss-különbség függvény jól használható információkkal szolgál az élekről, azonban ha egy él elég gyenge, akkor érzékeny lesz a zajra. Ahhoz, hogy kiszűrjük a további gyenge pontokat, meg kell vizsgálnunk, hogy van-e fő görbület az éllel párhuzamosan, illetve gyenge görbület a merőleges irányban D -ben az adott helyen. El kell vetni a pontot, ez a különbség a legnagyobb és legkisebb sajátvektor aránya alatt van, amit 2×2 -es Hesse-féle mátrixból számíthatunk ki az adott helyre illetve skálaszintre nézve.

A feldolgozás következő lépéseként a pontokhoz irányokat – akár többet is – kell rendelni, amelyeket a lokális gradiens jellemzők alapján határozunk meg. A gradiens m nagyságát és θ irányát az alábbi módon határozhatjuk meg:

$$m(x, y) = \sqrt{(L(x+1, y) - L(x-1, y))^2 + (L(x, y+1) - L(x, y-1))^2} \quad (6.8)$$

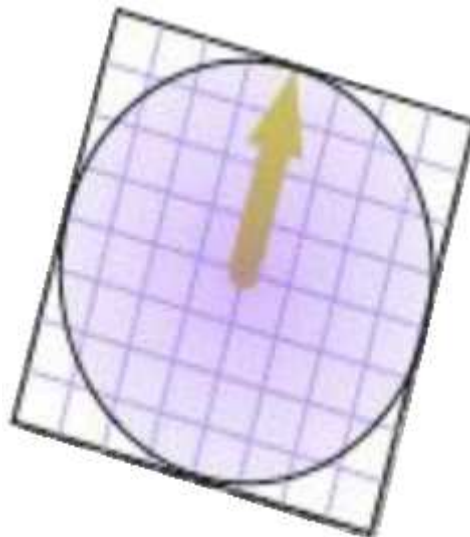
$$\theta(x, y) = \arctg \frac{L(x, y+1) - L(x, y-1)}{L(x+1, y) - L(x-1, y)} \quad (6.9)$$

Ezt elvégezzük az adott kulcspont adott sugarú – általában 4 vagy 8 – környezetére is, majd ezeket az értékeket egy olyan σ paraméterű Gauss-maszk szerint súlyozzuk, amely rendszerint másfélszerese a kulcspont léptékének (lásd a 6.10. ábrát). Ezt követően készítünk egy szöghisztogramot 36 vödörrel. Az így létrejött régiókat összegezzük 4×4 -es felbontású részekre úgy, hogy közben csökkentjük a vödrök számát nyolcra. Nagyon fontos, hogy az egyes kulcspon-



6.10. ábra. Szöghisztogram készítése.

tokhoz tartozó hisztogramok a ponthoz előzőleg kiszámolt orientációhoz vannak igazítva (lásd a 6.11. ábrát).



6.11. ábra. Irányított szöghisztogram.

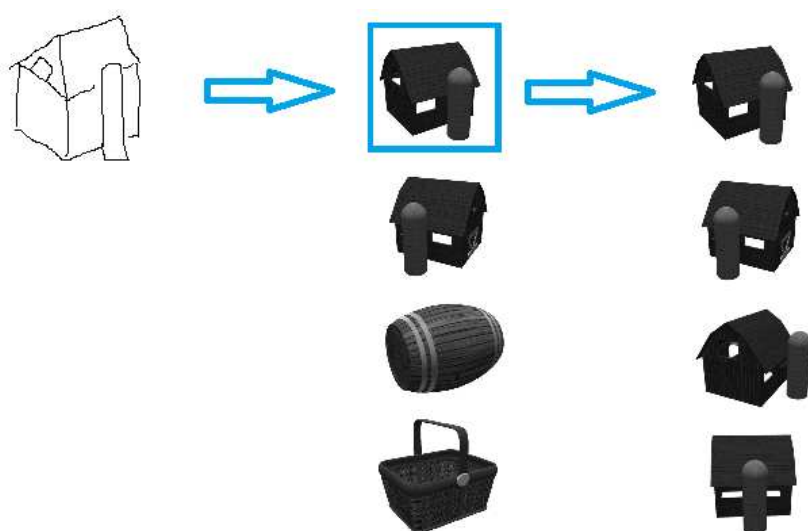
A domináns irányok a hisztogram kiemelkedő értékei lesznek. Alapvetően a legnagyobb ilyen csúcsot tekintjük fő iránynak, de ha előfordul még olyan érték, mely a maximális érték 80%-án belül van, akkor létre kell hozni vele egy új kulcspontot ugyanazon a helyen.

Általánosságban a pontok 15%-a rendelkezik többirányú hozzárendeléssel, ezzel csak növelik az ilyen kulcspontok stabilitását. A leíró végül úgy jön létre, hogy minden kulcspontoz és környezetéhez tartozik 4×4 darab hisztogram egyenként 8-8 értékkel. Ez összesen egy 128 elemű vektort eredményez. Ha a megvilágításból adódó változásokat szeretnénk kiküszöbölni, akkor normalizálnunk kell a vektort, így elérhetünk bizonyos fokú invarianciát e változásokkal

szemben.

A jellemző vektor előállító alrendszer felépítése

Rendszerünkben a skiccek alapján történő keresésnél először az EHD vagy a HOG leírót használjuk. Ezt követően pedig az eredmények és a felhasználó beavatkozását követően a SIFT leíró használatával javítunk a keresési eredményeken. A második lépésben már a színes, részletekben gazdag eredményképek alapján keresünk egy olyan módszerrel, ami ugyan színes kép és skicc között nem tud összehasonlítást végezni, de két színes kép között már hatékonyan tud keresést végrehajtani. A módszer előnyeit kihasználva egy interaktív visszakeresési folyamatot valósíthatunk meg, melyet a 6.12. ábra szemléltet.



6.12. ábra. Interaktív visszakeresési folyamat eredménye

Hasonló analógiát figyelhetünk meg az osztályozás területén is, amikor nagy adathalmazokat több klaszterbe osztanak szét. Az olyan módszereket, ahol több úgynevezett gyenge osztályozó algoritmusból összerakunk egy hatékonyabb algoritmust boosting technikának nevezzük. Ilyen jellegű klaszterező algoritmus pl. az AdaBoost [89] is.

A jellemzővektorokat előállító alrendszer bemenete az előfeldolgozáson átesett kép. Kimenete a leírótól függ. Abban az esetben, ha HOG vagy EHD leírót használunk, normalizált hisztogramot kapunk, mely az EHD használata esetén 150 vödröt tartalmaz, HOG-nál pedig a cellák száma szorozva a szöghisztogram vödorszámával. A SIFT leíró esetében a képen található minden egyes jellemző ponthoz 128 hosszúságú, szintén normalizált szöghisztogramot készítünk. Ha az adatbázishoz állítjuk elő a leírókat, akkor az alrendszer az adatbáziskezelővel

fog kapcsolatba lépni a továbbiakban. Amennyiben a visszakeresendő kép leíróját számoljuk, akkor pedig a visszakereső rendszerrel fog közreműködni.

6.1.5. Visszakereső alrendszer

A visszakereső alrendszerben az előállított jellemző vektorokat hasonlítjuk össze. Összehasonlításként a

$$D(x, y) = \left(\sum_{i=1}^n |x_i - y_i|^p \right)^{1/p} \quad (6.10)$$

Minkowski távolságot [22] használtuk, $p = 1$, $p = 2$ és $p \rightarrow +\infty$ paraméterekkel.

A rendszer bemenete a visszakeresendő kép és az adatbázisban lévő képek jellemző vektora, melyeket az EHD és HOG eljárások használatával állítottunk elő. A visszakeresés során meghatározzuk, hogy melyik jellemző vektorhoz van a legközelebb a skicc leírója a használt metrika szerint. Az eredményt sorba rendezve továbbadjuk a megjelenítő alrendszernek.

6.1.6. Az adatbáziskezelő alrendszer

Mind a képeket, mind a róluk készült leírókat eltároljuk, illetve biztosítjuk a későbbi feldolgozásához szükséges mechanizmusokat. Erre szolgál az adatbázis-kezelő alrendszer, mely három fő részből áll, a tároló-, a lekérdező- és az adatmanipulációs modulból [107].

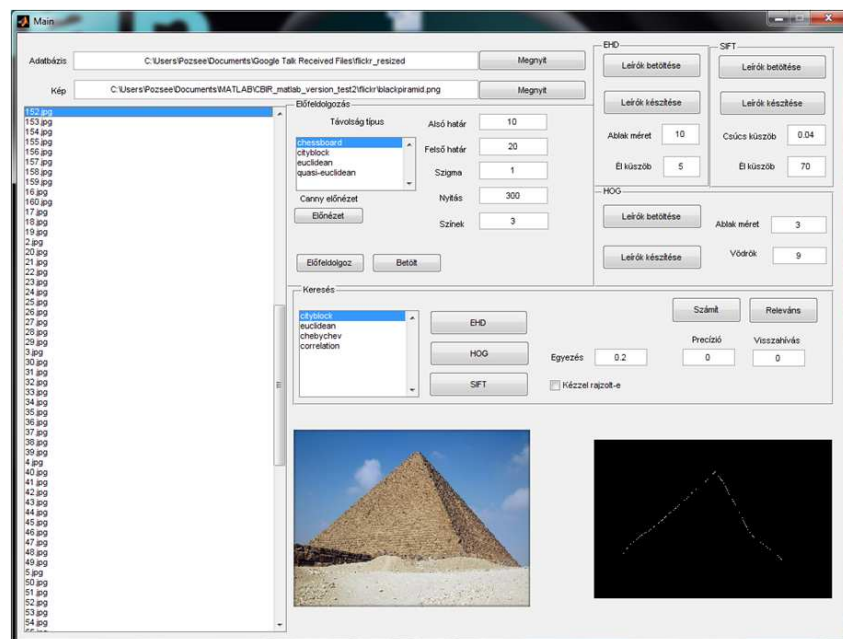
A tároló modul segítségével töltünk fel képeket az adatbázisba, a hozzá tartozó információkat és a jellemző vektorokat lementjük. A képpel kapcsolatban rögzítjük a fájl nevét, méretét, formátumát. Összegyűjtjük az elkészítés körülményeihez kapcsolódó információkat, a készítő nevét, a készítés időpontját, a kép címét, a felvevő egység márkáját és típusát. Ezek mellett szükségünk lehet még a színmélységre, a felbontásra, esetleg a kép származási helyére (URL cím), ezért ezek tárolásáról is gondoskodunk. A nagyméretű képeket tárolási célokból méretarányosan lecsökkentjük. Az adatokat egy globális helyen tároljuk.

A lekérdező modul segítségével kapjuk meg a visszakeresés eredményét. A visszakereső alrendszer kapcsolatba lép az adatbázissal, ami szolgáltatja a leírókat. Optimalizálás céljából ezt már induláskor betöltjük egy változóba. Ha megvan a visszakeresés eredménye, akkor az adatbázis felé fordulunk, ahol a leíróhoz tartozó elsődleges kulcs segítségével lekérjük az eredményképeket. Ezen felül különböző szempontok alapján készíthetünk statisztikákat SQL lekérdezés formájában az adatbázisunkkal kapcsolatban.

Az adatmanipulációs modul segítségével a képekhez tartozó háttér-információkat módosíthatjuk, kiegészíthetjük és létrehozhatjuk. Ennek segítségével kaput nyitunk a hagyományos kulcsszavas keresés felé is, illetve az automatizált címkézéshez is.

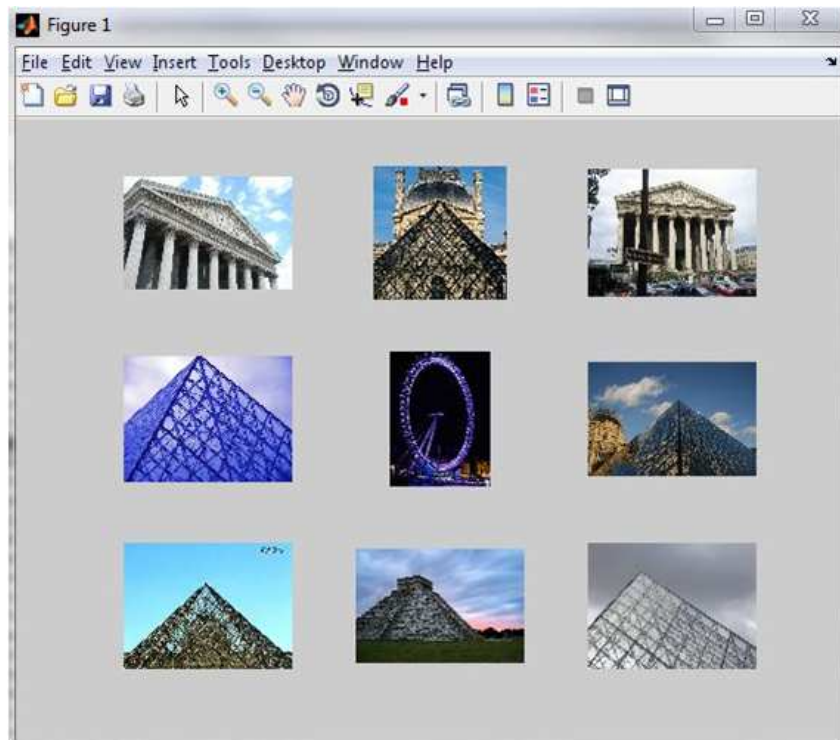
6.1.7. A megjelenítő alrendszer

Mivel rajzok képezik a visszakeresésünk alapját, így biztosítunk egy rajzfelületet, ahol ezeket elő lehet állítani. A kereséshez szükség van egy adatbázisra is, amelyet szintén be kell állítani a keresés előtt. A keresési eredmények rendszerezett megjelenítése nagy eredményhalmaz esetén nagyban megkönnyíti az eligazodást, így ezt is biztosítjuk, melyet a következőben bemutatunk. A rendszerünkben található módszerek nem működhetnének paraméterek nélkül, ezért lehetőséget biztosítunk ezek beállítására is. A felhasználói felület a 6.13. ábrán látható. A rendszer lehetőséget nyújt arra is, hogy az első kilenc találatot külön ablakban jelenítsük meg, ahogy ez a 6.14. ábrán is látható.



6.13. ábra. A rendszer felhasználói felülete.

Rendszerünkben a lehetséges találatokat osztályozzuk és az osztályoknak megfelelően jelenítjük meg [24]. Ezáltal a megoldáshalmaz átláthatóbb és rendezettebb lesz. Alapesetben a találatokat relevancia alapján jelenítjük meg, azonban előfordulnak fals pozitív találatok, melyek rontják a visszakeresés képét. Ha a találatokat valamilyen szempont szerint újra osztályozzuk, akkor az egy találati osztályra eső fals pozitív találatok száma csökken. Így a felhasználói



6.14. ábra. Az első kilenc legjobbnak ítélt találat.

megítélés is jobbá válik a rendszer felé. Úgy véltük, hogy számunkra a szín alapján való klaszterezés a legmegfelelőbb, mivel a szín áll legközelebb az emberi érzékeléshez egy kép kapcsán, így választásunk a k-közép algoritmuson alapuló osztályozási módszerre [18] esett, amely erre a célra tökéletesen megfelel.

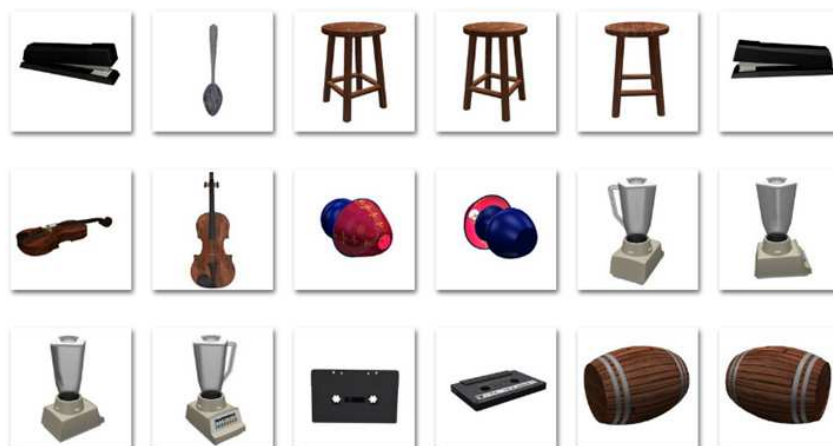
6.2. Tesztelés

6.2.1. Tesztkörnyezet

A rendszert több képi adatbázison is teszteltük annak érdekében, hogy pozitív és negatív tulajdonságairól minél objektívabb képet kapjunk.

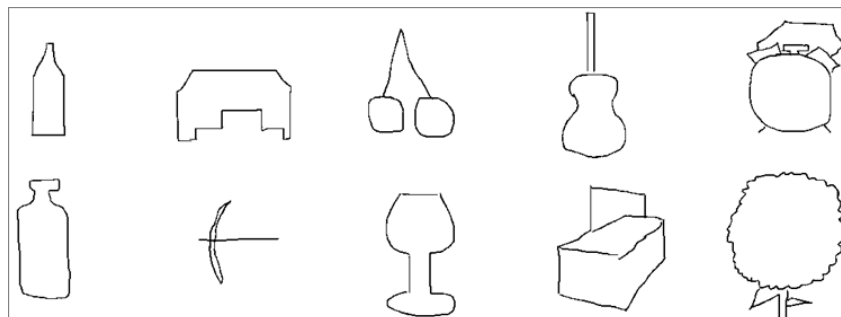
Első adatbázisként az Object Databank adatbázist [117] használtuk, amely kettőszázkilenc életszerű tárgyat tartalmaz. Minden egyes tárgyról tizennégy különböző irányból készítettek 450×450 -es felbontású képeket. Az adatbázist gyakran használják számítógépes és pszichológiai tanulmányokhoz. Az adatbázis néhány képe a 6.15. ábrán látható.

Az Object Databank adatbázis képeink végzett tesztek során a 6.16. ábrán látható típusú



6.15. ábra. Néhány mintakép az Object Databank adatbázisból.

szabadkézi rajzokkal hasonlítottuk össze az adatbázisban található képeket.



6.16. ábra. Az Object Databank adatbázissal végzett tesztekhez használt szabadkézi rajzok.

Másik használt tesztadatbázis a Flickr 160 adatbázis volt. Ezt az adatbázist egy szótár alapú kereső rendszer tesztelésénél használták fel a rendszer mérésére [43]. A FlickrR elnevezésű képmegosztó webhelyről válogattak ki százhatvan általános témájú képet. A képeket öt osztályba sorolták a rajtuk ábrázolt objektumok alakja szerint. Az adatbázis erősen redundáns, amely kiváló lehetőséget ad az ilyen jellegű tartalom alapú kereső rendszerek tesztelésére. A szerzők az adatbázis mellé csatolták azokat a példákat is, amely alapján a visszakereséseket végezték. Mivel dokumentálták teszteredményeiket és a visszakeresendő skiccek is rendelkezésünkre állnak, így a két rendszert össze lehet hasonlítani egymással. A 6.17. ábrán jól megfigyelhetőek az alakjuk szerint jól elkülönülő objektumtípusok.

Az adatbázis készítője a képeket öt csoportra bontotta, csoportonként harminckettő darab elemmel. A tesztelés során – hasonlóan az előző esethez – elkészítettük saját adatbázisunkat is, mely a 6.18. ábrán látható.



6.17. ábra. A Flickr 160 adatbázis egymástól jól elkülöníthető objektumtípusai.



6.18. ábra. A Flickr 160 adatbázis képeihez használt szabadkézi rajzok.

6.2.2. Tesztelési szempontok

Annak érdekében, hogy a rendszert alkotó módszerek hatékonyságáról objektív képet alkothassunk, valamint az egyes felhasznált módszereket összehasonlíthassuk, szükséges mérőszámok definiálása, melyekkel meg tudjuk határozni, hogy melyik módszer milyen körülmények között működik hatékonyan.

Ha van egy N darab képet tartalmazó teszt adatbázisunk, melyből Q darab számít releváns találatnak egy keresés során. Jelölje Z az elvárt releváns találatok számát, P pedig az eredmény lista hosszát. Ezek ismeretében meghatározható a rendszert jellemző alábbi két mérőszám [26]:

$$Precizio = \frac{Q}{P} \quad (6.11)$$

$$Felidezes = \frac{Q}{Z} \quad (6.12)$$

Az elvárt és releváns találatok száma függ a teszt adatbázistól is, az összes találatot pedig mi határozzuk meg. Ez utóbbi szám lényegében azt adja meg, hogy egy keresés során hány találatot vizsgál meg a felhasználó átlagosan. Ezekre a későbbiekben még vissza fogunk térni. Célunk számszerűsíteni, hogy a többszintű keresés milyen hatással van az eredményességre.

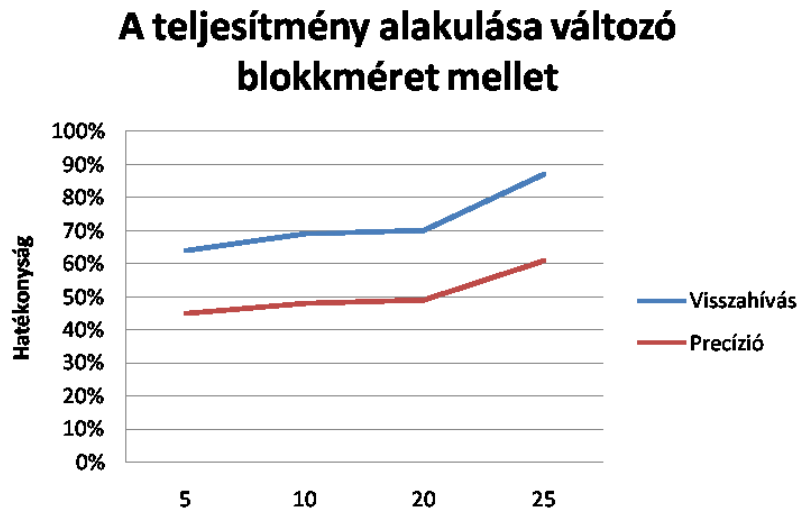
6.2.3. Teszt eredmények az EHD leíró használatokor

A leírók készítésekor befolyásolhatjuk a képblokkok számát. Egy alképen belül minél nagyobb ez a szám, annál részletesebb leírást kapunk. Ekkor viszont veszítünk a módszer robusztusságából, mivel a rajzolt kép és a visszakeresendő képek közötti különbségek nőnek. Beállíthatunk egy küszöböt, mely meghatározza, hogy az adott élt besoroljuk-e egy irányultsági osztályba, vagy elvetjük.

A szakirodalom több helyen is ajánl paraméter értékeket [3, 130]. Az összes képblokk számára egy alképen 1100 darabot javasolnak. Tehát ez a mi implementációnk szerint 33-as ablakméretet jelent, mivel $33 \cdot 33 \approx 1100$.

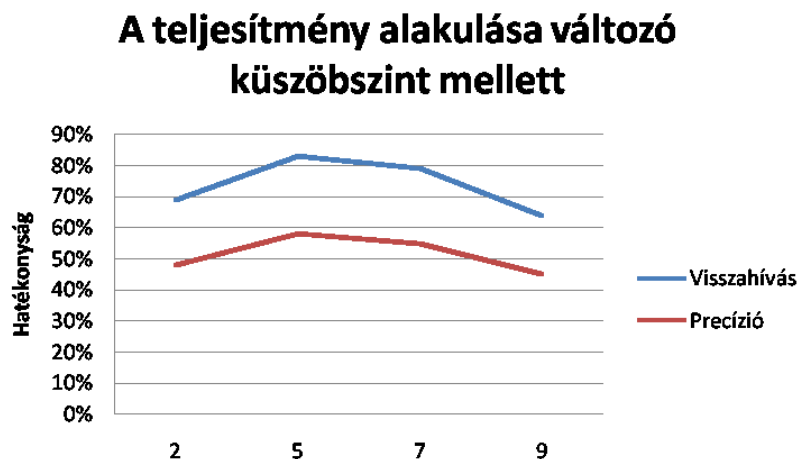
Az Object Databank adatbázis esetén a benne előforduló alakzatok sokfélesége miatt és a könnyebb átláthatóság érdekében az eredményeinket grafikonok segítségével mutatjuk be. Elsőként azt vizsgáltuk meg, hogy miként hat a teljesítmény alakulására, ha a küszöbértéket megtartjuk egy adott értéken, de a blokkméretet változtatjuk. Esetünkben a küszöb fix értékének kettőt adtunk, mert nagy blokkszám esetén – 20 felett már – értékelhetetlen, csupa nullából

álló leíróink lettek. Ugyanezen okból kifolyólag nem folytattuk a tesztelést 25 feletti blokk-számra. Megfigyeléseink szerint az 6.19. ábrán látható grafikon csúcspontja körül, tehát 25-ös blokkméret és 2-es küszöbparaméterek mellett érhető el a maximális teljesítmény.



6.19. ábra. A hatékonyság alakulása a blokkméret változásának függvényében

A további tesztelés során a küszöbszint változtatásával és a blokkméret rögzítésével próbáltuk ki az EHD leíró hatékonyságát. Blokkméretnek a fix 10-es értéket adtuk, majd 2-től kezdődően növeltük a küszöbértéket. Ezt a 6.20. ábrán látható grafikon szemlélteti.



6.20. ábra. A hatékonyság alakulása a küszöbszint változásának függvényében

A Flickr 160 adatbázis képeit az előfeldolgozási folyamat során 8 színűre redukáltuk, majd

Canny éldetektálást hajtottunk végre. Az előfeldolgozást követően a 6.21. ábrán látható képeket keresve végeztünk méréseket.



6.21. ábra. Diadalív, kör, piramis, templom és görög stílusú épület képe, melyekkel a Flickr 160 adatbázissal végzett tesztekert lefolytattuk.

A tesztelés eredményeit az 6.1. és 6.2. táblázatban foglaltuk össze.

	5 képblokk		15 képblokk		30 képblokk	
	Precízió	Felidézés	Precízió	Felidézés	Precízió	Felidézés
Diadalív	55%	47%	50%	53%	60%	53%
Kör	30%	25%	35%	31%	60%	41%
Piramis	45%	41%	50%	47%	65%	56%
Templom	15%	31%	35%	31%	50%	41%
Görög stílusú épület	50%	38%	60%	50%	80%	63%
Átlagos hatékonyság	39%	36%	46%	43%	63%	51%

6.1. táblázat. EHD alapú mérési eredmény 5 értékű állandó küszöb esetén.

Megfigyelhetjük, hogy a módszer a blokkszám növelésével egyre pontosabb találatokat szolgáltatott (lásd a 6.1. táblázatot), ebben az esetben viszont növekszik a feldolgozási idő. Ha a felhasználó részletesebb rajzot készít, akkor a paraméterek állítása esetén nagyobb változások figyelhetők meg. A küszöb változtatása esetében nem fogalmazható meg egyértelmű eredmény (lásd a 6.2. táblázatot). Nyilvánvalóan ez a kép előfeldolgozottságának minőségétől is függ. Előfordulhat, hogy egyes fontos élek csak gyenge élként jelennek meg, így azokat véletlenül kiszűrhetjük. Az EHD módszernél az a gyakorlati tapasztalat, hogy ha egy rajz nem pontosan ott helyezkedik el, ahol az objektum, akkor ezt egyes esetekben rosszul kezeli. A legjobb eredményt 30-as méretű képblokkal és 5-ös értékű küszöbvel értük el.

	2-es értékű küszöb		5-ös értékű küszöb		7-es értékű küszöb	
	Precízió	Felidézés	Precízió	Felidézés	Precízió	Felidézés
Diadalív	45%	44%	60%	53%	60%	44%
Kör	35%	31%	50%	31%	20%	19%
Piramis	50%	44%	55%	47%	70%	44%
Templom	25%	31%	35%	31%	35%	34%
Görög stílusú épület	50%	53%	65%	44%	70%	56%
Átlagos hatékonyság	41%	41%	53%	41%	51%	39%

6.2. táblázat. EHD alapú mérési eredmény 25 értékű állandó képblokk esetén.

6.2.4. Teszt eredmények a HOG leíró használatakor

A HOG alapú visszakeresés használatakor a leírók elkészítésekor befolyásolhatjuk az ablakméretet. Minél nagyobb ez a szám, annál részletesebb leírást kapunk, viszont így veszítünk a módszer robusztusságából, csakúgy, mint az EHD esetében. Beállíthatjuk a vödörszámot is, mely meghatározza, hogy az éleket hány csoportba kívánjuk besorolni. Kiindulásként a szakirodalomban számos helyen használt paraméterezést vizsgáltuk meg [43], majd ettől mindkét irányban eltérve végeztünk méréseket. Ablak méretnek 3-at, vödörszámnak pedig 9-et ajánlanak [43, 110], így ezeket választottuk kiindulási pontnak.

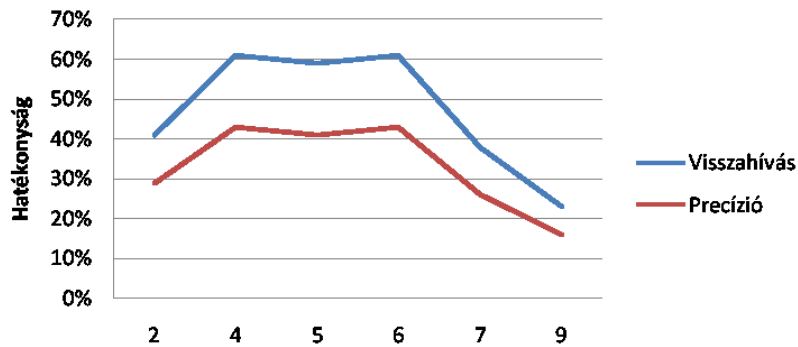
Az Object Databank adatbázison végezve a teszteléseket először a változó blokkszám hatásait teszteltük le rögzített 9-es vödörszám mellett. Az eredmények a 6.22. ábrán látható grafikonról olvashatóak le. Jól látható, hogy a 4-es, 5-ös illetve 6-os blokkméret mellett értük el a legjobb teljesítményt. A stagnálás valószínűleg a keresési eredmények pozitív, illetve negatív irányú kilengéseinek egyensúlyában keresendő, azonban 6-os blokkparaméter felett jelentős teljesítményromlás figyelhető meg.

Ahogy az EHD leírónál, úgy a HOG leíró használata esetén is leteszteltük a másik fő paraméter változásával járó teljesítmény-változásokat is. Ennél a tesztnél a blokkszámot fix 5-re választottuk, valamint a vödörszám változtatásával kísérleteztünk. A kapott eredményeket a 6.23. ábrán látható grafikon foglalja össze. Itt igazolódik be a [43]-ben is javasolt 9-es érték alkalmazhatósága.

A Flickr 160 adatbázison végezve a teszteléseket olyan eredményeket kaptunk, amelyek a [43]-ben ismertetett eredményekkel összemérhető. Eredményeinket a 6.3. és a 6.4. táblázatban foglaltunk össze.

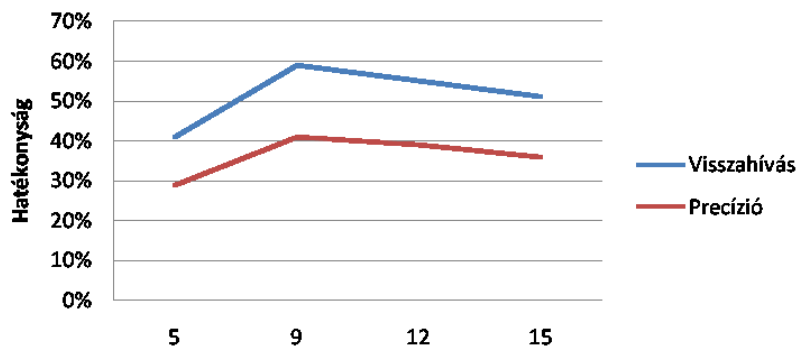
A táblázatok adataiból kiderül, hogy rögzített vödörszám mellett az ablakok részletesebb felosztásával növelhető a pontosság (lásd a 6.3. táblázatot). A leghatékonyabb kereséseket a

A teljesítmény alakulása változó blokkméret mellett



6.22. ábra. A hatékonyság alakulása a blokkok számának függvényében.

A teljesítmény alakulása változó vödorszám mellett



6.23. ábra. A hatékonyság alakulása a vödorszám változásának függvényében.

	3-as ablak		6-os ablak		10-es blokk	
	Precízió	Felidézés	Precízió	Felidézés	Precízió	Felidézés
Diadalív	30%	31%	35%	38%	65%	44%
Kör	65%	50%	80%	66%	80%	63%
Piramis	65%	56%	75%	66%	85%	69%
Templom	40%	41%	45%	41%	55%	50%
Görög stílusú épület	40%	25%	30%	31%	60%	56%
Átlagos hatékonyság	48%	41%	53%	48%	69%	56%

6.3. táblázat. HOG alapú mérési eredmény 9-es vödorszám esetén.

	5-ös vödörszám		9-es vödörszám		15-ös vödörszám	
	Precízió	Felidézés	Precízió	Felidézés	Precízió	Felidézés
Diadalív	50%	41%	65%	44%	45%	34%
Kör	65%	53%	80%	63%	90%	69%
Piramis	80%	66%	85%	69%	65%	47%
Templom	45%	41%	55%	50%	50%	50%
Görög stílusú épület	50%	44%	60%	56%	70%	56%
Átlagos hatékonyság	58%	49%	69%	56%	64%	51%

6.4. táblázat. HOG alapú mérési eredmény 10-es ablak méret esetén.

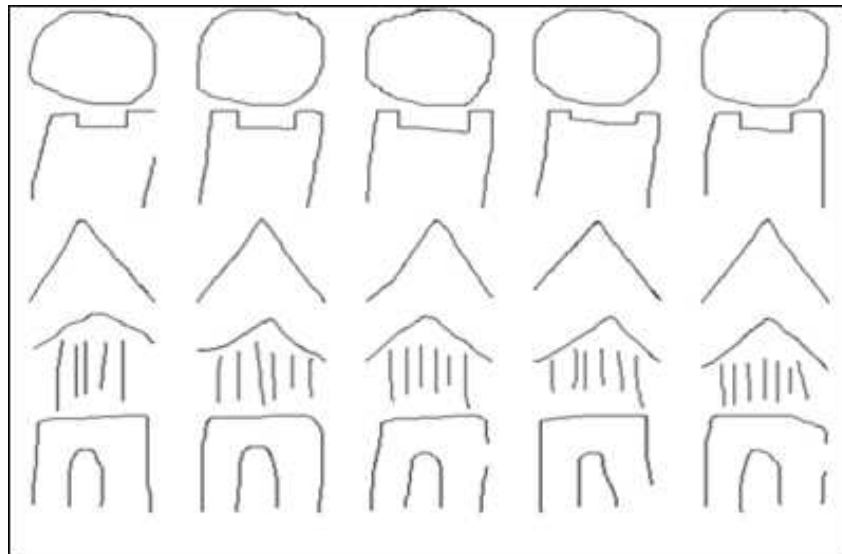
10-es paraméterezésű ablakméret esetén értük el. Fix ablakfelosztással és változó vödörszámmal történő kísérletek során pedig ugyanarra a következtetésre jutottunk, mint más szerző [43, 110] (lásd a 6.4. táblázatot). Ennek értelmében a módszer a 9-es vödörszám érték mellett működik a leghatékonyabban.

Általában elmondhatjuk, hogy az Object Databank adatbázisban szereplő képek esetén a EHD alapú visszakereséssel jobb eredményeket értünk el, mint az HOG alapúval. A FlickrR160 adatbázis esetén más a helyzet. Az előfeldolgozás komoly nehézségekbe ütközik a képek sokfélesége következtében. A képeket hasonlóan dolgoztuk fel, mint az EHD teszteknel. Az átfedő blokkok következtében egy-egy élszakasz az átlagolás következtében nem csak a saját blokkjára, hanem a szomszédjaira is hatással van. Tapasztalataink azt mutatják, hogy a HOG módszer sokkal jobban kezeli a sok kis élből, illetve a görbe élekből álló rajzokat. Ebből következően, a FlickrR160 képein szereplő nagyobb élmennyiség kedvez a HOG eljárásnak és indokoltá teszi annak használatát. A jelenség oka az átfedő blokkok közti átlagolásból ered. Olykor a rajz és a keresett objektum elhelyezkedése különbözik valamilyen irányban, azonban a kismértékű eltéréseket jól lehet kezelni ezzel a technikával. Ez nem mondható el az EHD esetében. A tesztek egyértelműen azt mutatják, hogy a HOG jobb módszernek bizonyult a FlickrR160 adatbázis esetében.

6.2.5. Összehasonlítás más rendszerrel

Elkészített rendszerünket egy konkurens megoldással összevetettük. A [43]-ben ismertetett rendszer a problémát hasonló irányból közelíti meg, mint ahogy mi tettük. Mi távolság transzformációs lépéssel egészítjük ki a leírást, ott pedig gradiens térképpel kiegészített HOG leírókat használtak, ahol legjobb paraméterezésnek 3-as ablak számot és 9-es vödörszámot adtak meg. Ezek mellett kísérleti célokból kipróbálták a HOG leíró használatát gradiens térkép készítése nélkül, valamint a SIFT módszert is. A tesztelést a FlickrR160 adatbázis segítségével végezték

el, ahol saját 25 elemű rajzadatbázisuk alapján kerestek (lásd a 6.24. ábrát). Mind egyes képre kiszámolták a precíziót. Az egyes eredményeket a 6.5. táblázatban mutatjuk be, ahol az utolsó két oszlopban láthatók a mi mérési eredményeink.



6.24. ábra. A [43]-ben használt szabadkézi rajzok.

Módszer	HOG (gradiens térképpel)	HOG (gradiens térkép nélkül)	SIFT	EHD (saját)	HOG (saját)
Átlagos precízió	54%	42%	41%	43%	44%

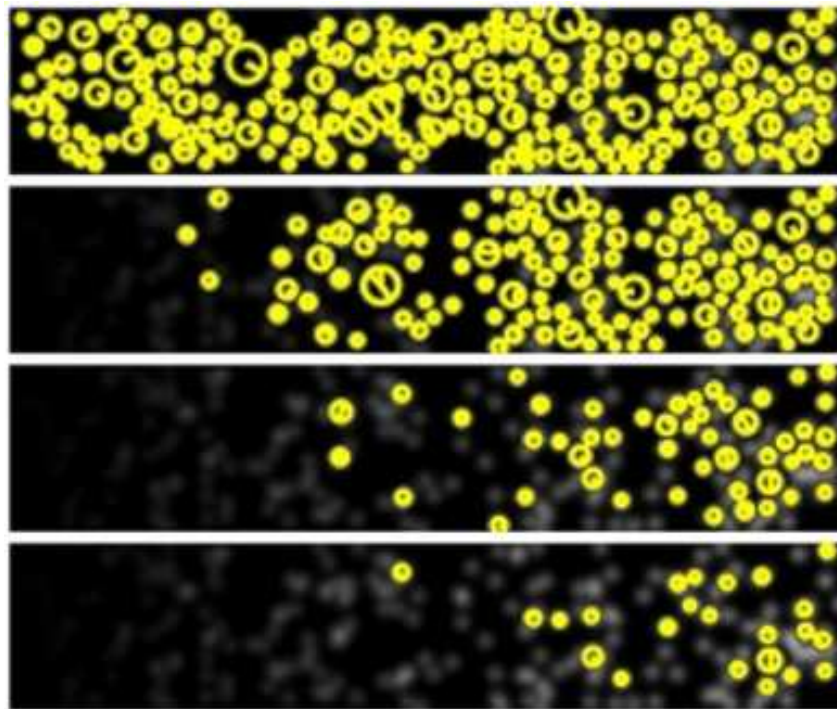
6.5. táblázat. Teszt alapú rendszereknél felhasznált módszerek teljesítménye.

6.3. Többszintű visszakeresés SIFT leíró használatával

Rendszerünkben egy többszintű visszakeresési lehetőséget teremtettünk meg azért, hogy egy kiválasztott eredménykép SIFT leírója alapján újra el lehet végezni egy pontosító keresést, így a felhasználói élményt és a pontosságot jelentősen meg tudtuk növelni. A keresés akkor hatékony és hasznos, ha olyan adatbázissal használjuk a programot, amelyik bizonyos objektumok előfordulására nézve nagyobb mértékben redundáns. Esetünkben ez mindegyik adatbázisra teljesül.

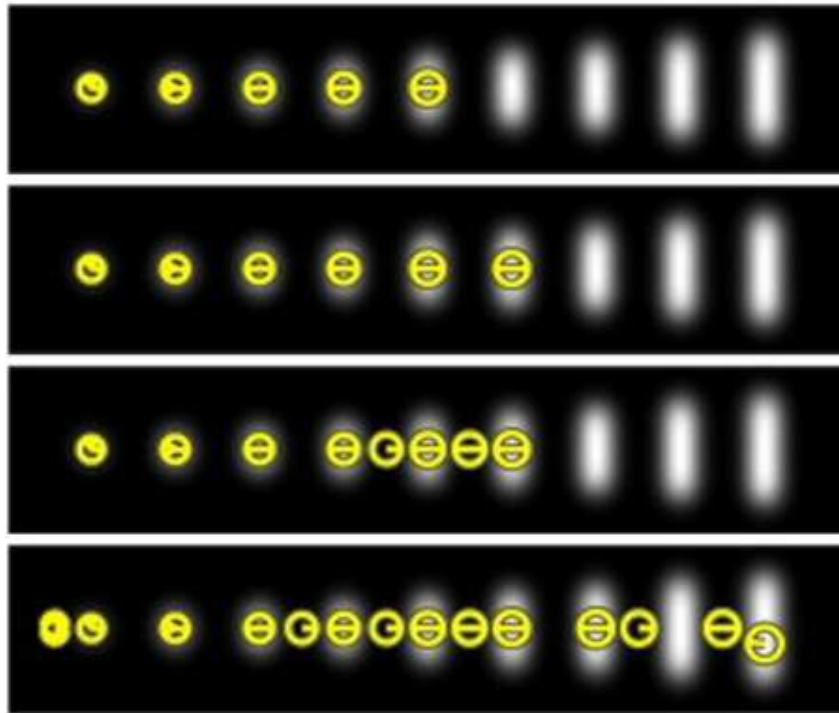
A SIFT implementáció három paraméter beállítását teszi lehetővé, ebből kettő a leíró vektorok elkészítését szabályozza, egy pedig a visszakeresést. A csúcsküszöb segítségével ki lehet szűrni a Gauss különbségtérben az abszolút értékben túl kicsi lokális szélsőértékeket. Ezzel

elérhetjük, hogy csak az igazán stabil és ez által reprezentatív pontok maradjanak meg további feldolgozásra (lásd a 6.25. ábrát). Az élküszöb paraméter segítségével a szélsőértékekhez tartozó görbületeknek adhatunk meg egy felső határt Gauss-különbség térben. Az érték növelésével egyre több élt vehetünk bele a szelekcióba a későbbi feldolgozáshoz (lásd a 6.26. ábrát). A harmadik paraméter a visszakeresés minőségét befolyásolja, egy küszöböt határoz meg a találatok jóságára.



6.25. ábra. A csúcsküszöb értékek növelésével egyre több halvány pont esik ki. Az értékek fentről lefelé: 0, 10, 20, 30 [123]

A következőkben megvizsgáljuk, milyen mértékben javítja a többszintű visszakeresési algoritmus a rendszer hatékonyságát. A SIFT leírók paraméterei a következők: csúcsküszöb=0,04, élküszöb=70, találatok jósága=0,2. Az egyes értékek heurisztikus próbálgatások eredményei. Bármely más irányba jelentősen elmozdulva rosszabb leírást és összehasonlítást kapunk. Ha egy olyan adatbázissal rendelkezünk, amelyben sok objektum több képen is előfordul, mint például az általunk használt tesztadatbázisok esetében, akkor a viszonylag gyenge visszakeresési eredményt pár kattintással fel lehet javítani (a folyamatot jól szemlélteti az 6.27., a 6.28. és a 6.29. ábra). Megvizsgáltuk az Object Databank és FlickrR160 segítségével a funkciónkat. A mérési eredményeket a 6.6. táblázat mutatja be. Látható, hogy módszerünk hatékonyan tudja javítani a visszakeresés sikerességét, ehhez persze megfelelő adatbázis kell. Ha nem is sokkal, de a módszer jobban működik valós környezetben készült képek esetén. A választás azért a



6.26. ábra. Él küszöbértékek: 3.5, 5, 7.5, 10 [123]

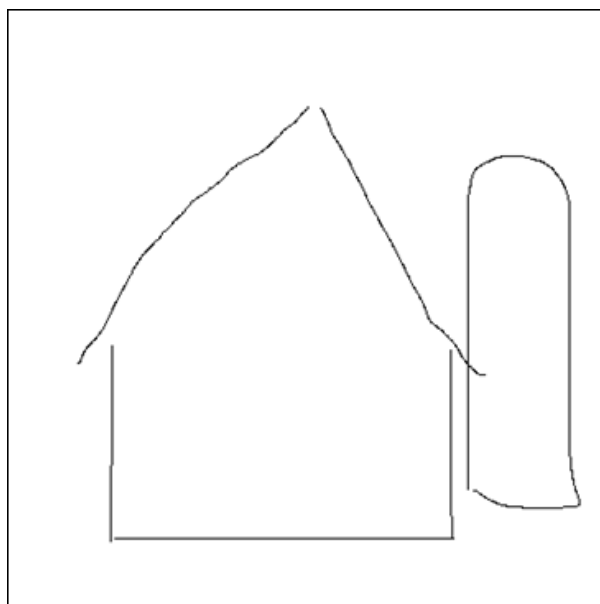
SIFT leíróra esett, mert forgatásra, eltolásra, a nézőpont változásaira és skálázásra invariáns bizonyos mértékig, előállítási ideje pedig gyors.

	Precízió az első kereséssel (EHD leíró használatával)	Precízió a második kereséssel (SIFT leíró használatával)
Farm	21%	86%
Kosár	28%	86%
Görög stílusú épület	60%	100%
Diadalív	45%	90%
Templom	40%	95%
Átlagos hatékonyság	39%	91%

6.6. táblázat. Mérési eredmények a többszintű, több ciklusú keresésnél

6.4. Összegzés

Elkészítettünk egy olyan tartalom alapú képviszakereső rendszert, mely az előfeldolgozott képeket indexeli HOG, illetve EHD módszerek felhasználásával, valamint vizsgálhatóvá teszi az különböző módszerek hatékonyságát. A rendszert kiegészítettük egy olyan végső fázis-



6.27. ábra. A felhasználó által rajzolt kiindulási skicc.



6.28. ábra. Az EHD alapú visszakeresés által szolgáltatott eredmény. A késsel kijelölt objektum alapján fogunk újra keresni az adatbázisban a SIFT leíró segítségével. A találati sorrend : balról jobbra, illetve fentről lefelé.



6.29. ábra. A SIFT használatát követően növekedik a találati pontosság. A találati sorrend: balról jobbra, illetve fentről lefelé.

sal, melynek során a találatok közül valamelyiket kiválasztva, SIFT algoritmus használatával tovább pontosítja a visszakeresési listát. Két fő szempontot vettünk figyelembe. A visszakeresési folyamat legyen a hagyományostól eltérő és nagymértékben interaktív. A módszernek robusztussága elengedhetetlen bizonyos mértékű zajokig, ami még egyszerű képek esetében előfordulhat. Implementáltunk és felhasználtunk több jellemző-vektor elkészítési algoritmust. A kezdeti próbálkozásaink során szembetűnt a probléma, hogy a rajzolt képet a színessel, sőt ennek él reprezentációjával módosítás nélkül nem lehet összehasonlítani. Megoldásként közbeiktattunk egy távolságtranszformációs lépést, amely nélkül nem működne a rendszerünk. Az egyszerű simítás-élkeresés alapú előfeldolgozó módszert továbbfejlesztettük, mely hasonló fontossággal bírt, mint az előző lépés. Rendszerünket kiegészítettük egy pontosító mechanizmussal, amely a SIFT algoritmus segítségével tovább növeli a releváns találatok számát. Úgy ítéljük meg, hogy a több szintű keresésnek fontos helye van egy tartalom alapú keresőrendszerben. Az eredmények szín szerinti csoportosítása pedig kiegészíti az alak alapú keresést, pótolja a színek vizsgálatának hiányát. Mérési adatainkat összevetettük a [43] publikációban közzétett méréssel (lásd a 6.5. táblázatot).

A teszteknel összevetettük az EHD és az általunk dinamikusan paraméterezzhető továbbfejlesztett HOG implementáció hatékonyságát, előállítási idejét több adatbázisra nézve. Tapasztalataink szerint a HOG felülmúlta az EHD alapú visszakeresést (lásd a 6.7. táblázatot). A helyzet azonban nem ilyen egyszerű. Az élhisztogram leíróval főként információban szegényebb,

kevés vonalból álló skiccekre lehet hatékonyabban keresni, míg a másik esetben a részletesebbekre kapunk jobb eredményeket. Ez a HOG átfedő ablakos megoldása miatt van. SIFT alapú többszintű keresési megoldással, az eredményképek alapján finomítottunk a találati listán. A visszakeresési válasz szín alapú kategorizálásával pedig nagyobb döntési lehetőséget adtunk a felhasználó számára, olyan módon, hogy nem egy statikus listával szembesül, hanem több csoport közül választhat.

	HOG	EHD
Átlagos precízió	51%	45%
Legjobb eredmény	69%	63%
Legrosszabb eredmény	16%	39%

6.7. táblázat. A HOG és EHD leírók rövid összehasonlítása.

Az eredmények azt mutatják, hogy módszerünk használata általános jellegű és egyszerű objektumokat tartalmazó ábrákra sikeresnek bizonyult, nem maradt el más hasonló rendszerek eredményeitől. Az elvégzett tesztek során kiderült, hogy általános beállítások nem léteznek. Automatizálás nélkül csak heurisztikus gondolatokkal próbálkozhatunk. Elmondhatjuk, hogy a CBIR rendszerek belefutottak a hatékony, automatikus szegmentáló algoritmusok hiányába, mint ahogy az a mi esetünkben is történt. Nem tudták a képet az adott szempont szerinti információtartalom alapján felbontani, mivel nem léteztek és nem is léteznek általánosan működő konkrét megoldások. Fontos tudni azt is, hogy a tartalom alapú visszakereső rendszerek miatt újra nagy hangsúly helyeződött a szín alapú képfeldolgozásra. Ez bizonyára kapcsolatban áll az emberi érzékeléssel.

Felmerülhet a kérdés, hogy miért van szükség skicc alapú keresőrendszerekre. A válasz egyszerű. A hagyományos adatkezelési módszerek számos téren már nem tudnak megküzdeni az adatok sokaságával. Új eljárásokra van szükség. A téma korszerű és igen fontos. Valamit felvázolok és képeket keresek, amelyeken valami hasonló van. Ez a gyakorlati problémamegoldások mind gyakoribb igénye. Mivel az ember alapvetően vizuális típus, erre a szempontra is hangsúlyt kell fektetni. Több különböző terület felől is igény mutatkozik képi tartalom alapú keresőrendszerekre, beleértve a rajz alapú keresést is. Megemlíthetjük a bűnüldözést (fantomképek, graffitik, tetoválások azonosítása), ahol a hatóságok munkáját támogathatnák a hatékonyság növelése érdekében. Akár még a múzeumok digitális adatbázisait is fel lehet ruházni interaktív keresőrendszerekkel. Képmegosztó portálokon pedig már fórumokat nyitnak, ahol már sokan várják a saját rendszerünkben használt hasonló technológiák megjelenését.

7. fejezet

Összegzés (Tézisek)

Az internet óriási mértékű elterjedésével, a digitális kamerák árának csökkenésével, valamint a számítógépek tároló kapacitásának növekedésével az utóbbi húsz évben jelentős mértékben megnövekedett az elérhető képek száma. A képek rendszerezett tárolásának és visszakereshetőségének igénye maga után vonta a képi adatbázisok elterjedését is.

A képi adatbázisokkal szemben elvárás, hogy a benne tárolt képeket hatékony módon tudjuk visszakeresni. Ehhez két lényeges funkcióval kell rendelkeznie ezen adatbázisoknak. Egyrészt az egyes képekhez jól jellemző indexeket kell generálni és tárolni, másrészt ezen indexek összehasonlításával egy keresés során a felhasználó igényeinek legmegfelelőbb eredményeket kell szolgáltatnia.

A képek indexelése leggyakrabban két módon történik. Az első és jelenleg legelterjedtebb a szöveges indexelés, amikor minden egyes képhez azt jól jellemző kulcsszavakat rendelünk hozzá, majd egy keresés során ezen kulcsszavakat hasonlítjuk össze. Ebben az esetben a kulcsszavakat általában emberi munkával lehet előállítani. Ez egyrészt maga után vonja a szubjektív döntéshozatalt, másrészt új jellemzők alapján történő kulcsszó előállítás csak nagyon nagy munkával valósítható meg.

A tartalom alapú képkereső rendszereknél a képet leíró indexeket, vagy tulajdonság vektorokat a számítógép állítja el automatizáltan a képen tárolt információk (pixelek intenzitása, szomszédsági viszonyok, stb.) felhasználásával. Ezen rendszerek esetén a kulcsszavak előállítása sokkal gyorsabb, mint az ember által előállított szöveges leírók használata esetén, valamint új igényekhez illeszkedő indexelés is egyszerűbben megvalósítható.

Az indexek előállítása előtt a képet úgy kell átalakítani, hogy abból hatékonyan lehessen az információkat kinyerni. Ez képfeldolgozási előfeldolgozó eljárások használatát igényli.

1. téziscsoport

A képfeldolgozó algoritmusok sok esetben előre megadott paraméterek alapján szolgáltatnak eredményt. Ezen paraméterek értéke viszont gyakran függ a képen tárolt információktól. Képi adatbázisok indexelésénél szükséges, hogy ezeket a paramétereket a képen tárolt információk függvényében automatizált módon tudjuk előállítani.

1.1. tézis: Eljárást dolgoztam ki, mellyel szegmentáló algoritmusok automatikus paraméterezése valósítható meg. Az eljárás alkalmazhatóságát tesztek igazolják. [99, 98, 97, 95]

1.2. tézis: Eljárást dolgoztam ki élkereső algoritmusok automatikus paraméterezésének hatékony megvalósítására. Az eljárást más hasonló módszerrel összehasonlítottam és ezen tesztek igazolják, hogy pontosabb paraméterezést sikerült ilyen módon megvalósítani. [99, 20]

2. téziscsoport

Az előfeldolgozás számos esetben megkívánja valamilyen simító, illetve zajcsökkentő eljárás használatát.

Simítást többféle módon is meg lehet valósítani. Leggyakoribb erre a kép szűrése például átlagoló, vagy Gauss maszkkal. Gyakran használt eljárás a kép Fourier transzformálása oly módon, hogy az előálló trigonometrikus tagokból csak az első pár tagot tartjuk meg. Hasonló módszer lehet a magasabb rendű szinguláris érték dekompozíció (HOSVD - High Order Singular Value Decomposition) használata. Ebben az esetben a képet három dimenziós tenzornak tekintjük a vizsgálati módszereknél ismertetett módon. Ezt a három dimenziós tenzort ortonormált függvények kompozíciójaként állítjuk elő. Ha ebből az előállításból is csak pár tagot tartunk meg, akkor a Fourier transzformációhoz hasonlóan szűrést tudunk megvalósítani, amelynek eredménye részletgazdagabb, így a képi indexek legyártására alkalmasabb eredményt szolgáltat.

2.1. tézis: Igazoltam, hogy HOSVD-alapú függvény approximációval az azonos számú ortonormált függvény megtartása esetén a transzformált képen végrehajtott simítás nagyobb részletgazdagságot eredményez mint ugyanolyan számú trigonometrikus függvény megtartása esetén a Fourier-alapú approximáció. [88]

Ennek az eljárásnak másik előnye, hogy nem jelennek meg a Fourier-alapú közelítésnél előálló ciklikus hullámok, melyek a Fourier sor alaptulajdonságai miatt jelennek meg a képeken.

3. téziscsoport

Képi adatbázisokban történő keresésnél az egyes képekhez hozzárendelt indexeket hasonlítjuk össze valamilyen hasonlósági mérték alkalmazásával. Szín alapú összehasonlítás esetén a figyelembe vett index gyakran a kép színhisztogramja. A színhisztogramok előállításánál a vödrökre osztás miatt információvesztés következik be. Az információ veszteségét a gyakran használt távolsági- és hasonlósági mértékek közül a kvadratus távolság használatával lehet kompenzálni.

3.1. tézis: Hisztogramok kvadratus távolságánál olyan új súlyokat vezettem be, melyek a keresés által szolgáltatott releváns találatok számát növelték a nem kvadratus távolságot használó, illetve más súlyozású kvadratus távolságot alkalmazó eljárásokhoz képest. [94, 91]

4. téziscsoport

A tartalom alapú képkeresés egy speciális területe a szabadkézi vonalrajzok (skiccek) alapján történő keresés. Ezen rendszereknél a felhasználó által készített szabadkézi rajzhoz hasonló alakú objektumokat keresünk képeken. A kereséshez szükséges, hogy a képeken található tárgyakat úgy alakítsuk át, hogy azok összemérhetőek legyenek a skiccekkel.

4.1. tézis: Kifejlesztettem egy olyan előfeldolgozó eljárást, mely skicc-alapú kereső rendszer esetén valós körülmények között készített képeket úgy alakít át, hogy azok összemérhetőek szabadkézi vonalrajzokkal. A kifejlesztett módszert más hasonló rendszerekkel azonos tesztkörülmények között összemértem. A mérési eredmények igazolták, hogy az előfeldolgozó eljárásom eredményesebben alkalmazható, mint más eljárások. [115]

A tartalom alapú képkereső rendszerekben a megtalált képeknek a felhasználó igényeihez minél jobban igazodniuk kell. Ennek érdekében visszacsatolási lehetősége is van sok esetben a felhasználónak, ahol megadhatja, hogy a megtalált képek közül melyeket tekinti relevánsnak. A rendszer ezen visszacsatolás alapján a felhasználó igényeihez jobban illeszkedő találatokat állíthat elő.

4.2. tézis: Skicc alapú kereső rendszerbe beépítettem egy olyan felhasználói releváns visszacsatolási lehetőséget, mely az ún. SIFT-leíró használatával jelentős mértékben növeli a keresések eredményességét. [115]

7.1. Az eredmények hasznosítása, továbbfejlesztési lehetőségek

A szegmentáló és élkereső algoritmusok automatikus paraméterezését megvalósító algoritmus nem csak képi adatbázisok indexelése esetén használható, hanem más esetekben is, amikor a felhasználónak nincs lehetősége megállapítani az adott képhez legjobban illeszkedő paraméter értéket. Az algoritmus hátránya viszont, hogy jelentős futási időt igényel, ami a képi adatbázisok indexelésénél nem probléma, más esetekben viszont problémát jelenthet.

A HOSVD módszer a képfeldolgozás területén széles körben alkalmazható lehet, így pl. simításra, képek nagyítása esetén interpolációs technikaként, valamint élek detektálására is. A képi adatbázisok esetében megvizsgálandó, hogy a HOSVD eljárást követően előálló képre jellemző ortonormált függvények a kép indexelésére alkalmazhatók e.

Hasonlóan nyitott kérdés még, hogy a HOSVD alapú eljárás lényegkiemelésre, az információ tömörítésére, valamint a képek osztályozására hatékonyan alkalmazható módszer e.

A kidolgozott súlyozott kvadratikus távolság használatának továbbfejlesztési lehetősége lehet, ha az alkalmazott színtérben egyenletes eloszlástól eltérő színeloszlást feltételezve az egyes irányokban más-más súlyokat alkalmazunk.

A skicc alapú rendszerben megvalósított releváns visszacsatolás SIFT leíró használatával nem csak skicc alapú, hanem általános tartalom alapú kereső rendszerekben is alkalmazható lehet, de ezzel kapcsolatban kísérleteket még nem végeztem.

Publikációs és hivatkozási lista

Az alábbiakban felsorolom a disszertációban hivatkozott releváns saját publikációimat, valamint az azokra érkezett független hivatkozásokat.

Folyóiratcikk

- A1. Sergyán, Sz.,** Csink, L.: Automatic Parametrisation of Image Processing Algorithms. *SCIENTIFIC BULLETIN of "Politechnica" University of Timisoara*, Vol. 54, No. 1, 2009, pp. 53–58., ISSN 1224-600X
- A2. Sergyán, Sz.:** A New Approach of Face Detection-based Classification of Image Databases. *Acta Polytechnica Hungarica*, Vol. 6, No. 1, 2009, pp. 175–184., ISSN 1785-8860
- A2c1.** Guo, L., Liao, Y., Luo, D., Liao, H.: Face image classification using appearance and texture features. *International Conference on Computer Application and System Modeling (ICCASM)*, Vol. 3, 22–24 October, 2010, pp. V3-476–V3-480.,

Külföldi nemzetközi konferencia

- B1.** Rövid, A., Rudas, I.J., **Sergyán, Sz.,** Szeidl, L.: HOSVD Based Image Processing Techniques. In: *Proc. of 10th WSEAS International Conference on Artificial Intelligence, Knowledge Engineering and Data Bases, Cambridge, UK, February 20–22, 2011*, pp. 297–302., ISBN: 978-960-474-273-8
- B2.** Szántó, B., Pozsegovics, P., Vámosy, Z., **Sergyán, Sz.:** Sketch4Match – Content-based Image Retrieval System Using Sketches. In: *Proc. of 9th IEEE International Symposium on Applied Machine Intelligence and Informatics, Smolenice, Slovakia, January 27-29, 2011*, pp. 183-188., ISBN 978-1-4244-7428-8, IEEE Catalog Number: CFP1108E-CDR

- B3. Sergyán, Sz.:** Content-Based Image Retrieval Using Automatically Determined Color Regions of Images. *In: Proc. of 7th International Symposium on Applied Machine Intelligence and Informatics, Herl'any, Slovakia, January 30-31, 2009*, pp. 41-45., ISBN 978-1-4244-3802-9, IEEE Catalog Number: CFP0908E-CDR
- B4. Sergyán, Sz.:** Classification of Image Databases Using Face Detection. *In: Proc. of 6th International Symposium on Intelligent Systems and Informatics, Subotica, Serbia, September 26-27, 2008*, ISBN 978-1-4244-2407-8, IEEE Catalog Number: CFP0884C-CDR
- B5. Sergyán, Sz.:** Color Histogram Features-based Image Classification in Content-based Image Retrieval Systems. *In: Proc. of 6th International Symposium on Applied Machine Intelligence and Informatics, Herl'any, Slovakia, January 21-22, 2008*, pp. 221-224., ISBN 978-1-4244-2106-0, IEEE Catalog Number: CFP0808E-CDR
- B5c1.** Stanley, R.J., De, S., Demner-Fushman, D., Antani, S., Thoma, G.R.: An Image Feature-based Approach to Automatically Find Images for Application to Clinical Decision Support. *Computerized Medical Imaging and Graphics*, In Press, Corrected Proof, 2010, ISSN 0895-6111
- B5c2.** Yang, H., Han, J., Cao, F.: Method of Image Retrieval Based on Bit-plane Histogram Features Vector. *Computer Engineering and Applications*, 46(21), 2010, pp. 165–167., DOI F10.3778/j.issn.1002-8331.2010.21.047
- B5c3.** Tieta, A.R.P., Aniati, M.A.: Image Feature Extraction and Recognition of Abstractionism and Realism Style of Indonesian Paintings. *2nd Intenational Conference on Advances in Computing, Control, and Telecommunication Technologies, Jakarta, Indonesia*, December 2–3, 2010, pp. 149–152., ISBN: 978-0-7695-4269-0
- B5c4.** Zhu, Z., Brilakis, I.: Parameter Optimization for Automated Concrete Detection in Image Data. *Autimation in Construction*, Vol. 19, Issue 7, November 2010, pp. 944–953., ISSN 0926-5805
- B5c5.** Jyothi, B., Madhavee, L.Y., Reddy, V.S.K.: Medical Image Retrieval using Multiple Features. *Advances in Computational Sciences and Technology*, Vol. 3, No. 3, 2010, pp. 387–396., ISSN 0973-6107
- B5c6.** Singh, S.: RGB Color Histogram Feature based Image Classification: An Application of Rough Reasoning. *Proceedings of the First International Conference on Intelligent Human Computer Interaction*, 2009, pp. 102–112., DOI 10.1007/978-81-8489-201-1_8

- B5c7.** Abenius, T.: Classification of Cell Images Using MPEG-7-included Descriptors and Support Vector Machines in Cell Morphology. *Thesis for a Diploma in Computer Science, Department of Computer Science, Faculty of Science, Lund University, December 12., 2008*
- B6.** Csink, L., **Sergyán, Sz.:** Automatic Parametrization of Edge Detection Algorithms. In: *Proc. of 5th International Symposium on Intelligent Systems and Informatics, Subotica, Serbia, August 24-25, 2007, pp. 119-121., ISBN 987-86-7031-131-3*
- B7.** **Sergyán, Sz.,** Csink, L.: Automatic Parametrization of Region Finding Algorithms in Gray Images. In: *Proc. of 4th International Symposium on Applied Computational Intelligence and Informatics, Timisoara, Romania, May 17-18, 2007, pp. 199-202., ISBN 1-4244-1234-X, IEEE Catalog Number: 07EX1788*
- B8.** **Sergyán, Sz.:** Color Content-based Image Classification. In: *Proc. of 5th Slovakian-Hungarian Joint Symposium on Applied Machine Intelligence and Informatics, Poprad, Slovakia, January 25-26, 2007, pp. 427-434., ISBN 978-963-7154-56-0*
- B8c1.** Balamurugan, P., Rajesh, R.: Classification of Greenery and Non-Greenery Images using Forward Color Coherence Vector (FCCV) and Guided Color Features. *ICGST International Conference on Artificial Intelligence and Machine Learning, Dubai, United Arab Emirates, April 12-14, 2011, pp. 61–65., ISSN: 1687-4846*
- B8c2.** Shang, C., Barnes, D., Shen, Q.: Effective Feature Selection for Mars McMurdo Terrain Image Classification. *9th International Conference on Intelligent Systems Design and Applications, 2009, pp. 1419–1424., DOI 10.1109/ISDA.2009.105*
- B9.** **Sergyán, Sz.:** Content Based Image Retrieval in Database of Segmented Images. In: *Proc. of 4th Slovakian-Hungarian Joint Symposium on Applied Machine Intelligence, Herl'any, Slovakia, January 20-21, 2006, pp. 380-388., ISBN 963 7154 44 2*
- B10.** **Sergyán, Sz.,** Csink, L.: Consistency Check of Image Databases. In: *Proc. of 2nd Romanian-Hungarian Joint Symposium on Applied Computational Intelligence, Timisoara, Romania, May 12-14, 2005, pp. 201-206., ISBN 963 7154 39 6*
- B10c1.** Tick, J.: P-Graph-based Workflow Modelling. *Acta Politechnica Hungarica, Vol. 4, 2007, No. 1, pp. 75-88.*
- B10c2.** Tick, J.: Workflow Modelling Based on Process Graph. In: *Proc. of 5th Slovakian-Hungarian Joint Symposium on Applied Machine Intelligence and Informatics, Poprad, Slovakia, January 25-26, 2007, pp. 419-426., ISBN: 978-963-7154-56-0*

B11. Kiss, A., Németh, T., **Sergyán, Sz.**, Vámosy, Z., Csink, L.: Recognition of a Moving Object in a Stereo Environment Using a Content Based Image Database. *In: Proc. of 3rd Slovakian-Hungarian Joint Symposium on Applied Machine Intelligence, Herl'any, Slovakia, January 21-22, 2005*, pp. 65-74.

B11c1. Hermann, Gy.: Computer Controlled Calibration System for Dial Gauges. *In: Proc. of International Symposium on Logistics and Industrial Informatics, Wildau, Germany, September 13-15, 2007*, pp. 117-120., ISBN 978-1-4244-1441-3

B11c2. Hermann, Gy.: Calibration System for Dial Gauges. *In: Proc. of 4th International Symposium on Applied Computational Intelligence and Informatics, Timisoara, Romania, May 17-18, 2007*, pp. 41-44.

Belföldi nemzetközi konferencia

C1. **Sergyán, Sz.**: Special Distances of Image Color Histograms. *In: Proc. of 5th Joint Conference on Mathematics and Computer Science, Debrecen, June 9-12, 2004*, pp. 92.

Belföldi konferencia

D1. **Sergyán Sz.**, Csink L.: Kísérletek a szín-alapú tartomány felismerés terén. *Informatika a Felsőoktatásban 2005 Konferencia, Debrecen, 2005. aug. 24-26.*, ISBN 963 472 909 6

Irodalomjegyzék

- [1] A. Balakrishnan. *Introduction to Optimization Theory in a Hilbert Space*. Springer-Verlag, Heidelberg, New York, 1971.
- [2] A. Balakrishnan. *Applied Functional Analysis*. Springer-Verlag, New York, Heidelberg, 1976.
- [3] R. Balasubramani and V. Kannan. Efficient use of MPEG-7 color layout and edge histogram descriptors in cbir systems. *Global Journal of Computer Science and Technology*, 9(4):157–163, 2009.
- [4] P. Baranyi, L. Szeidl, and P. Várlaki. Numerical reconstruction of the HOSVD based canonical form of polytopic dynamic models. In *IEEE 10th International Conference on Intelligent Engineering Systems*, pages 196–201, London, United Kingdom, June 2006.
- [5] M. Bober. MPEG-7 visual shape descriptors. *IEEE Transactions on Circuits and Systems for Video Technology*, 11(6):716–719, June 2001. ISSN 1051-8215.
- [6] D. Cai, X. He, Z. Li, W.-Y. Ma, and J.-R. Wen. Hierarchical clustering of WWW image search results using visual, textual and link information. In *12th ACM International Conference on Multimedia*, pages 952–959, New York, NY, USA, 2004. ISBN 1-58113-893-8.
- [7] D. Cai, X. He, W.-Y. Ma, J.-R. Wen, and H. Zhang. Organizing WWW images based on the analysis of page layout and Web link structure. In *IEEE International Conference on Multimedia and Expo*, volume 1, pages 113–116, June 2004. ISBN 0-7803-8603-5.
- [8] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–698, 1986. ISSN 0162-8828.
- [9] C. Carson, S. Belongie, H. Greenspan, and J. Malik. Blobworld: Image segmentation using expectation-maximization and its application to image querying. In *IEEE Tran-*

- sactions on Pattern Analysis and Machine Intelligence*, volume 24, pages 1026–1038, 2002.
- [10] C. Carson, M. Thomas, S. Belongie, J. M. Hellerstein, and J. Malik. Blobworld: A system for region-based image indexing and retrieval. In D. P. Huijsmans and A. W. M. Smeulders, editors, *VISUAL'99*, LNCS 1614, pages 509–517. Springer-Verlag Berlin Heidelberg, 1999.
- [11] C.-H. Chang and C.-W. Ha. Sharp inequalities of singular values of smooth kernels. *Integral Equations and Operator Theory*, 35(1):20–27, March 1999.
- [12] E. Chang and S. Tong. SVM active support vector machine active learning for image retrieval. In *ACM International Multimedia Conference*, pages 107–118, October 2001.
- [13] S. Chang and S. Liu. Picture indexing and abstraction techniques for pictorial databases. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(4):475–483, 1984.
- [14] Y. Chen, J. Li, and J. Z. Wang. *Machine Learning and Statistical Modeling Approaches to Image Retrieval*. The Kluwer International Series on Information Retrieval. Kluwer Academic Publisher, Boston / Dordrecht / London, 2004.
- [15] Y. Chen, J. Wang, and R. Krovetz. An unsupervised learning approach to content-based image retrieval. In *IEEE International Symposium on Signal Processing and its Applications*, pages 197–200, July 2003.
- [16] S.-F. Cheng, W. Chen, and H. Sundaram. Semantic visual templates: Linking visual features to semantics. In *International Conference on Image Processing*, volume 3, pages 531–535, Chicago, IL, USA, October 1998. ISBN 0-8186-8821-1.
- [17] C.-Y. Chiu, H.-C. Lin, and S.-N. Yang. Texture retrieval with linguistic descriptors. In *IEEE Pacific Rim Conference on Multimedia*, pages 308–315, 2001.
- [18] D. Comaniciu and P. Meer. Robust analysis of feature spaces: Color image segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 750–755, June 1997. ISBN 0-8186-7822-4.
- [19] I. Cox, M. Miller, T. Minka, T. Papatomas, and P. Yianilos. The bayesian image retrieval system, PicHunter: Theory, implementation, and psychophysical experiments. *IEEE Transactions on Image Processing*, 9(1):20–37, 2000.

- [20] L. Csink and S. Sergyán. Automatic parametrization of edge detection algorithms. In *5th International Symposium on Intelligent Systems and Informatics*, pages 119–121, Subotica, Serbia, August 2007. ISBN 987-86-7031-131-3.
- [21] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 886–893, July 2005.
- [22] T. Deselaers, D. Keysers, and H. Ney. Features for image retrieval: An experimental comparison. *Journal Information Retrieval*, 11(2):77–107, November 2007.
- [23] J. Eakins and M. Graham. Content-based image retrieval. Technical report, University of Northumbria, Newcastle, 1999.
- [24] M. Eitz, K. Hildebrand, T. Boubekeur, and M. Alexa. Photosketch: A sketch based image query and compositing system. In *SIGGRAPH 2009: Talks*, SIGGRAPH '09, pages 60:1–60:1, New Orleans, Louisiana, 2009. ACM. ISBN 978-1-60558-834-6.
- [25] M. Eitz, K. Hildebrand, T. Boubekeur, and M. Alexa. An evaluation of descriptors for large-scale image retrieval from sketched feature lines. *Computers and Graphics*, 34:482–498, October 2010.
- [26] R. Fabbri, L. D. F. Costa, J. C. Torelli, and O. M. Bruno. 2D euclidean distance transform algorithms: A comparative survey. *ACM Computing Surveys*, 40(1):1–44, February 2008.
- [27] C. Faloutsos, R. Barber, M. Flickner, J. Hafner, W. Niblack, D. Petkovic, and W. Equitz. Efficient and effective querying by image content. *Journal of Intelligent Information Systems*, 3(3-4):231–262, 1994.
- [28] L. Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. *Computer Vision and Image Understanding*, 106(1):59–70, April 2007.
- [29] H. Feng and T.-S. Chua. A bootstrapping approach to annotating large image collection. In *Workshop on Multimedia Information Retrieval*, pages 55–62. ACM Multimedia, November 2001.
- [30] H. Feng, R. Shi, and T.-S. Chua. A bootstrapping framework for annotating and retrieving WWW images. In *12th ACM International Conference on Multimedia*, pages 960–967, New York, NY, USA, 2004. ISBN 1-58113-893-8.

- [31] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, page 264, June 2003. ISBN 0-7695-1900-8.
- [32] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Hiang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, and P. Yanker. Query by image and video content: The QBIC system. *IEEE Computer*, 28:23–32, 2002.
- [33] J.-M. Geusebroek, G. J. Burghouts, and A. W. Smeulders. The amsterdam library of object images. *International Journal of Computer Vision*, 61(1):103–112, 2005.
- [34] T. Gevers and A. Smeulders. Pictoseek: Combining color and shape invariant features for image retrieval. *IEEE Transactions on Image Processing*, 9(1):102–119, 2000.
- [35] R. Gonzalez and R. Woods. *Digital Image Processing*. Prentice Hall, Upper Saddle River, USA, 2002.
- [36] G.-D. Guo, A. Jain, W.-Y. Ma, and H.-J. Zhang. Learning similarity measure for natural image retrieval with relevance feed. *IEEE Transactions on Neural Networks*, 13(4):811–820, 2002.
- [37] A. Gupta and R. Jain. Visual information retrieval. *Communications of the ACM*, 40(5):70–79, 1997.
- [38] G. Györök. Embedded hybrid controller with programmable analog circuit. In *IEEE 14th International Conference on Intelligent Systems*, pages 59.1–59.4, May 2010.
- [39] R. Haralick. Statistical and structural approaches to texture. *Proceedings of the IEEE*, 67(5):786–804, May 1979. ISSN 0018-9219.
- [40] M. Hearst, S. Dumais, E. Osman, J. Platt, and B. Scholkopf. Support vector machines. *IEEE Intelligent Systems and their Applications*, 13(4):18–28, July/August 1998. ISSN 1094-7167.
- [41] A. Hoover, G. Jean-Baptiste, X. Jiang, P. Flynn, H. Bunke, D. Goldgof, K. Bowyer, D. Eggert, A. Fitzgibbon, and R. Fisher. An experimental comparison of range image segmentation algorithms. *IEEE Transactions on Pat*, 18(7):673–689, July 1996. ISSN 0162-8828.
- [42] P. Howarth and S. Rüger. Evaluation of texture features for content-based image retrieval. *Image and Video Retrieval*, 3115:326–334, 2004.

- [43] R. Hu, M. Barnard, and J. Collomosse. Gradient field descriptor for sketch based retrieval and localization. In *17th IEEE International Conference on Image Processing*, pages 1025–1028, Hong Kong, September 2010. ISSN 1522-4880.
- [44] A. Jain, J. Lee, and R. Jin. Sketch to photo matching: A feature-based approach. In *SPIE, Biometric Technology for Human Identification VII*, volume 7667, pages 766702–766702, 2010.
- [45] A. Jain, J. Lee, R. Jin, and N. Gregg. Content based image retrieval: An application to tattoo images. In *IEEE Conference on Image Processing*, pages 2745–2748, November 2009.
- [46] A. Jain, J. Lee, R. Jin, and N. Gregg. Graffiti-id: Matching retrieval of graffiti images. In *ACM MM, MiFor'09*, pages 1–6, 2009.
- [47] F. Jing, M. Li, L. Zhang, H.-J. Zhang, and B. Zhang. Learning in region-based image retrieval. In *International Conference on Image and Video Retrieval*, pages 206–215, 2003.
- [48] C. Jörgensen. *Image Retrieval: Theory and Research*. Scacecrow Press, 2003.
- [49] T. Kanungo and Q. Zheng. Estimating degradation model parameters using neighborhood pattern distributions: An optimization approach. *IEEE Transactions on Patt*, 26(4):520–524, April 2004. doi:10.1109/TPAMI.2004.1265867.
- [50] A. Khotanzad and Y. H. Hong. Invariant image recognition by Zernike moments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(5):489–497, 1990.
- [51] A. Khotanzad and J. H. Lu. Classification of invariant image representation using a neural network. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 38(6):1028–1038, 1990.
- [52] W. Koontz, P. Narendra, and K. Fukunaga. A graph-theoretic approach to nonparametric cluster analysis. *IEEE Transactions on Computers*, C-25(9):936–944, September 1976. ISSN 0018-9340.
- [53] S. Kulkarni and B. Verma. Fuzzy logic for texture queries in CBIR. In *International Conference on Computational Intelligence and Multimedia Applications*, pages 223–226, Xi'an, China, 2003.
- [54] S. Kullback. *Information Theory and Statistics*. Dover, 1978.

- [55] L. D. Lathauwer, B. D. Moor, and J. Vandewalle. A multilinear singular value decomposition. *SIAM Journal Matrix Anal. Appl.*, 21(4):1253–1278, 2000.
- [56] B. Li, E. Chang, and Y. Wu. Discovery of a perceptual distance function for measuring image similarity. *Multimedia Systems*, 8(6):512–522, 2003.
- [57] Y. Li, J. Bilmes, and L. Shapiro. Object class recognition using images of abstract regions. In *17th International Conference on Pattern Recognition*, volume 1, Cambridge, UK, August 2004. ISBN 0-7695-2128-2.
- [58] Y. Li, L. Shapiro, and J. Bilmes. A generative/discriminative learning algorithm for image classification. In *10th International Conference on Computer Vision*, volume 2, Beijing, China, October 2005. ISBN 0-7695-2334-X.
- [59] Y. Liu, D. Zhang, G. Lu, and W.-Y. Ma. Region-based image retrieval with perceptual colors. In *Pacific-Rim Multimedia Conference*, pages 931–938, December 2004.
- [60] Y. Liu, D. Zhang, G. Lu, and W.-Y. Ma. A survey of content-based image retrieval with high-level semantics. *Pattern Recognition*, 40(1):262–282, Januar 2007.
- [61] F. Long, H. Zhang, and D. Feng. *Multimedia Information Retrieval and Management*, chapter Fundamentals of Content-based Image Retrieval. Springer, Berlin, 2003.
- [62] D. Lowe. Object recognition from local scale-invariant features. In *IEEE International Conference on Computer Vision*, volume 2, page 1150, 1999.
- [63] D. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60:91–110, 2004.
- [64] J. Luo and A. Savakis. Indoor vs outdoor classification of consumer photographs using low-level and semantic features. In *International Conference on Image Processing*, volume 2, pages 745–748, October 2001.
- [65] W. Ma and B. Manjunath. NeTra: A toolbox for navigating large image databases. In *IEEE International Conference on Image Processing*, pages 568–571, 1997.
- [66] J. Matas. *Colour-Based Object Recognition*. PhD thesis, Department of Electronic and Electrical Engineering, University of Surrey, 1996.
- [67] S. Mehrotra, Y. Rui, M. Ortega-Binderberer, and T. Huang. Supporting content-based queries over images in mars. In *IEEE International Conference on Multimedia Computing and Systems*, pages 632–633, 1997.

- [68] B. M. Mehtre, M. S. Kankanhalli, and W. F. Lee. Shape measures for content-based image retrieval: A comparison. *Information Processing and Management*, 33(3):319–337, 1997.
- [69] V. Mezaris, I. Kompatsiaris, and M. Strintzis. An ontology approach to object-based image retri. In *International Conference on Image Processing*, volume 2, pages 511–514, 2003.
- [70] A. Mojsilovic and B. Rogowitz. Capturing image semantics with low-level descriptors. In *International Conference on Image Processing*, pages 18–21, Thessaloniki, Greece, October 2001. ISBN 0-7803-6725-1.
- [71] A. Natsev, R. Rastogi, and K. Shim. WALRUS: A similarity retrieval algorithm for image databases. *IEEE Transactions on K*, 16(3):301–316, March 2004. ISSN 1041-4347.
- [72] S. Nene, S. Nayar, and H. Murase. Columbia object image library (COIL-100). Technical report, Columbia University, February 1996. CUCS-006-96.
- [73] W. Niblack, R. Barber, W. Equitz, M. D. Flickner, E. H. Glasman, D. Petkovic, P. Yanker, C. Faloutsos, G. Taubin, and Y. Heights. Querying images by content, using color, texture, and shape. In *SPIE Conference on Storage and Retrieval for Image and Video Databases*, volume 1908, pages 173–187, 1993.
- [74] V. Ogle and M. Stonebacher. Chabot: Retrieval from a relational database of images. *IEEE Computer*, 28(9):40–48, 1995.
- [75] P. Ohanian and R. Dubes. Performance evaluation for four classes of textural features. *Pattern Recognition*, 25:819–833, 1992.
- [76] G. Pass, R. Zabih, and J. Miller. Comparing images using color coherence vectors. In *4th ACM International Conference on Multimedia*, pages 65–73, New York, USA, 1996. ISBN 0-89791-871-1.
- [77] D. Paulus, L. Csink, and H. Niemann. Color cluster rotation. In *International Conference on Image Processing*, volume 1, pages 161–165, Chicago, IL, USA, October 1998. ISBN 0-8186-8821-1.
- [78] A. Pentland, R. Picard, and S. Sclaroff. Photobook: Content-based manipulation for image databases. *International Journal of Computer Vision*, 18(3):233–254, 1996.

- [79] K. Plataniotis and A.N.Venetsanopoulos. *Color Image Processing and Applications*. Springer, Berlin, 2000.
- [80] W. Pratt. *Digital Image Processing*. Wiley, Hoboken, USA, 2007.
- [81] J. Puzicha, T. Hofmann, and J. Buhmann. Non-parametric similarity measures for unsupervised texture segmentation and image retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 267–272, 1997.
- [82] J. Ren, Y. Shen, and L. Guo. A novel image retrieval based on representative colors. In *Image and Vision Computing*, pages 102–107, November 2003.
- [83] Y. Rubner, C. Tomasi, and L. J. Gubas. The earth mover’s distance as a metric for image retrieval. *International Journal of Computer Vision*, 40(2):99–121, 2000.
- [84] Y. Rui and T. Huang. Optimizing learning in image retrieval. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 1236–1243, June 2000.
- [85] Y. Rui, T. Huang, and S. Mehrotra. Content-based image retrieval with relevance feedback in Mars. In *IEEE International Conference on Image Processing*, pages 815–818, 1997.
- [86] Y. Rui, T. Huang, M. Ortega, and S. Mehrotra. Relevance feedback: A power tool for interactive content-based image retrieval. *IEEE Transactions on Circuits and Systems for Video Technology*, 8(5):644–655, September 1998. ISSN 1051-8215.
- [87] Y. Rui, T. S. Huang, and S.-F. Chang. Image retrieval: Current techniques, promising directions, and open issues. *Journal of Visual Communication and Image Representation*, 10(1):39–62, March 1999.
- [88] A. Rövid, I. J. Rudas, S. Sergyán, and L. Szeidl. HOSVD based image processing techniques. In *10th WSEAS International Conference on Artificial Intelligence, Knowledge Engineering and Data Bases*, pages 297–302, Cambridge, UK, February 2011. ISBN 978-960-474-273-8.
- [89] R. E. Schapire and Y. Singer. Improved boosting algorithms using confidence-rated predictions. *Machine Learning*, 37(3):297–336, December 1999. ISSN 0885-6125.
- [90] E. Schmidt. Zur Theorie der linearen und nicht linearen Integralgleichungen Zweite Abhandlung Auflösung der allgemeinen linearen Integralgleichungen. *Mathematische Annalen*, 64(2):161–174, June 1907.

- [91] S. Sergyán. Special distances of image color histograms. In *5th Joint Conference on Mathematics and Computer Science*, page 92, Debrecen, Hungary, June 2004.
- [92] S. Sergyán. Color content-based image classification. In *5th Slovakian-Hungarian Joint Symposium on Applied Machine Intelligence and Informatics*, pages 427–434, Poprad, Slovakia, January 2007. ISBN 978-963-7154-56-0.
- [93] S. Sergyán. Classification of image databases using face detection. In *6th International Symposium on Intelligent Systems and Informatics*, Subotica, Serbia, September 2008. ISBN 978-1-4244-2407-8, IEEE Catalog Number: CFP-0884C-CDR.
- [94] S. Sergyán. Color histogram features-based image classification in content-based image retrieval systems. In *6th International Symposium on Applied Machine Intelligence and Informatics*, pages 221–224, Herl'any, Slovakia, January 2008. ISBN 978-1-4244-2106-0, IEEE Catalog Number: CFP0808E-CDR.
- [95] S. Sergyán. Content-based image retrieval using automatically determined color regions of images. In *7th International Symposium on Applied Machine Intelligence and Informatics*, pages 41–45, Herl'any, Slovakia, January 2009. ISBN 978-1-4244-3802-9, IEEE Catalog Number: CFP0908E-CDR.
- [96] S. Sergyán. A new approach of face detection-based classification of image databases. *Acta Polytechnica Hungarica*, 6(1):175–184, 2009. ISSN 1785-8860.
- [97] S. Sergyán and L. Csink. Kísérletek a szín-alapú tartomány felismerés terén. In *Informatika a Felsőoktatásban 2005 Konferencia*, Debrecen, Hungary, August 2005. ISBN 963-472-909-6.
- [98] S. Sergyán and L. Csink. Automatic parametrization of region finding algorithms in gray images. In *4th International Symposium on Applied Computational Intelligence and Informatics*, pages 199–202, Timisoara, Romania, May 2007. ISBN 1-4244-1234-X, IEEE Catalog Number: 07EX1788.
- [99] S. Sergyán and L. Csink. Automatic parametrization of image processing algorithms. *SCIENTIFIC BULLETIN of "Politechnica" University of Timisoara*, 54(1):53–58, 2009. ISSN 1224-600X.
- [100] I. Sethi and I. Coman. Mining association rules between low-level image features and high-level concepts. In *SPIE Data Mining and Knowledge Discovery*, volume 3, pages 279–290, 2001.

- [101] H. C. Shen and A. K. C. Wong. Generalized texture representation and metric. *Computer, Vision, Graphics, and Image Processing*, 23:187–206, 1983.
- [102] A. W. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-based image retrieval at the end of the early years. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12):1349–1380, December 2000. ISSN 0162-8828.
- [103] J. Smith and S. Chang. Visualseek: A fully automated content-based image query system. In *ACM Multimedia '96*, pages 97–98, 1996.
- [104] J. Smith and C.-S. Li. Decoding image semantics using composite region templates. In *IEEE Workshop on Content-Based Access of Image and Video Libraries*, pages 9–13, Santa Barbara, CA, USA, June 1998. ISBN 0-8186-8544-1.
- [105] M. Sonka, V. Hlavac, and R. Boyle. *Image Processing, Analysis, and Machine Vision*. Thomson, 2008.
- [106] D. Stan and I. Sethi. Mapping low-level image features to semantic concepts. In *SPIE Storage and Retrieval for Media Databases*, pages 172–179, 2001.
- [107] P. L. Stanchev. General image database model. *Visual Information and Information Systems*, 1614:29–36, 1999.
- [108] P. L. Stanchev, D. G. Jr., and B. Dimitrov. High level color similarity retrieval. *International Journal Information Theories and Applications*, 10(3):363–369, 2003.
- [109] M. Stricker and M. Orengo. Similarity of color images. In *SPIE Storage and Retrieval for Image and Video Databases*, volume 2185, pages 381–392, 1995.
- [110] F. Suard, A. Rakotomamonjy, A. Bensrhair, and A. Broggi. Pedestrian detection using infrared images and histogram of oriented gradients. In *IEEE Intelligent Vehicles Symposium*, pages 206–212, Tokyo, September 2006. ISBN 4-901122-86-X.
- [111] M. J. Swain and D. H. Ballard. Color indexing. *International Journal of Computer Vision*, 7(1):11–32, 1991.
- [112] L. Szeidl, P. Baranyi, Z. Petres, and P. Várlaki. Numerical reconstruction of the HOSVD based canonical form of polytopic dynamic models. In *3rd International Symposium on Computational Intelligence and Intelligent Informatics*, pages 111–116, Agadir, Morocco, 2007.

- [113] L. Szeidl, I. J. Rudas, A. Rövid, and P. Várlaki. HOSVD based method for surface data approximation and compression. In *12th International Conference on Intelligent Engineering Systems*, pages 197–202, Miami, Florida, February 2008. ISBN 978-1-4244-2083-4.
- [114] L. Szeidl and P. Várlaki. HOSVD based canonical form for polytopic models of dynamic systems. *Journal of Advanced Computational Intelligence and Intelligent Informatics*, 13(1):52–60, 2009. ISSN 1343-0130.
- [115] B. Szántó, P. Pozsegovics, Z. Vámosy, and S. Sergyán. Sketch4Match - Content-based image retrieval system using sketches. In *9th IEEE International Symposium on Applied Machine Intelligence and Informatics*, pages 183–188, Smolenice, Slovakia, January 2011. ISBN 978-1-4244-7428-8, IEEE Catalog Number: CFP1108E-CDR.
- [116] H. Tamura, S. Mori, and T. Yamawaki. Textural features corresponding to visual perception. *IEEE Transactions on Systems, Man and Cybernetics*, 8:460–472, 1978.
- [117] M. Tarr. Rotating objects to recognize them: A case study on the role of viewpoint dependency in the recognition of three-dimensional objects. *Psychonomic Bulletin and Review*, 2(1):55–82, March 1993. ISSN 1069-9384.
- [118] S. Tong and E. Chang. Support vector machine active learning for image retrieval. In *ACM International Conference on Multimedia*, pages 107–118, Ottawa, Canada, 2001.
- [119] C. Town and D. Sinclair. Content-based image retrieval using semantic visual categories. Technical report, Society for Manufacturing Engineers, 2001.
- [120] E. Trucco and A. Verri. *Introductory Techniques for 3-D Computer Vision*. Prentice Hall, Upper Saddle River, USA, 1998.
- [121] S. Umbaugh. *Computer Imaging – Digital Image Analysis and Processing*. CRC Press, 2005.
- [122] A. Vailaya, M. Figueiredo, A. Jain, and H. Zhang. Image classification for content-based indexing. *IEEE Transactions on Image P*, 10(1):117–130, 2001.
- [123] A. Vedaldi and B. Fulkerson. VLFEAT: An open and portable library of computer vision algorithms. In *International Conference on Multimedia*, pages 1469–1472, New York, NY, USA, 2010. ISBN 978-1-60558-933-6.

- [124] R. C. Veltkamp, M. Tanase, and D. Sent. *State-of-the-art in Content-based Image and Video Retrieval*, chapter Features in Content-based Image Retrieval Systems: A Survey, pages 97–124. Kluwer, 2001.
- [125] S. Wan, P. Prusinkiewicz, and S. Wong. Variance-based color image quantization for frame buffer display. *Color Research and Application*, 15(1):52–58, February 1990.
- [126] J. Wang, J. Li, and G. Wiederhold. SIMPLIcity: Semantics-sensitive integrated matching for picture libraries. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(9):947–963, 2001.
- [127] J. Wang, G. Wiederhold, O. Firschein, and X. Sha. Content-based image indexing and searching using daubechies’ wavelets. *International Journal on Digital Libraries*, 1(4):311–328, 1998.
- [128] J. Wang, W.-J. Yang, and R. Acharya. Color clustering techniques for color-content-based image retrieval from image databases. In *IEEE International Conference on Multimedia Computing and Systems*, pages 442–449, Ottawa, Canada, June 1997. ISBN 0-8186-7819-4.
- [129] Y. Whan, Q. Chen, and B. Zhang. Image enhancement based on equal area dualistic sub-image histogram equalization method. *IEEE Transactions on Consumer Electronics*, 45(1):68–75, February 1999. ISSN 0098-3063.
- [130] C. S. Won, D. K. Park, and S.-J. Park. Efficient use of MPEG-7 edge histogram descriptor. *ETRI Journal*, 24(1):23–30, February 2002.
- [131] Y. Yitzhaky and E. Peli. A method for objective edge detection evaluation and detector parameter selection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(8):1027–1033, August 2003.
- [132] H. Yu, M. Li, H.-J. Zhang, and J. Feng. Color texture moments for content-based image retrieval. In *International Conference on Image Processing*, volume 3, pages 929–932, June 2002. ISBN 0-7803-7622-6.
- [133] D. Zhang, A. Wong, M. Indrawan, and G. Lu. Content-based image retrieval using gabor texture features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 13–15, 2000.
- [134] L. Zhang, F. Liu, and B. Zhang. Support vector machine learning for image retrieval. In *International Conference on Image Processing*, pages 7–10, October 2001.

- [135] X. Zheng, D. Cai, X. He, W.-Y. Ma, and X. Lin. Locality preserving clustering for image database. In *12th ACM International Conference on Multimedia*, New York, 2004. ISBN 1-58113-893-8.
- [136] X. Zhou and T. Huang. CBIR: From low-level features to high-level semantics. In *SPIE, Image and Video Communications and Processing*, volume 3974, pages 426–431, San Jose, CA, January 2000.
- [137] X. S. Zhou and T. S. Huang. Relevance feedback in image retrieval: A comprehensive review. *Multimedia Systems*, 8(6):536–544, 2003.
- [138] Y. Zhuang, X. Liu, and Y. Pan. Apply semantic template to support content-based image retrieval. In *SPIE Storage and Retrieval for Media Databases*, pages 442–449, San Jose, California, USA, January 2000.