

BABEŞ-BOLYAI UNIVERSITY CLUJ-NAPOCA
FACULTY OF MATHEMATICS AND INFORMATICS
SPECIALIZATION: COMPUTER SCIENCE

License Thesis

Intelligent photo analyzer by composition rules

Abstract

Nowadays the smart phones have more and more professional cameras. This is the reason why we are disposed to record every moment of our life. But the good quality of these photos doesn't guarantee a high artistic value. It is fundamental to follow some composition rules in order to take aesthetic photos.

The purpose of this paper is to present a method that takes an input photo and crops out a smaller region that follows the composition rules. Firstly, we detect the salient regions and the leading lines, then we calculate the aesthetic score of the photo based on composition rules. Finally, changing the relative position of these objects we try to find and crop the most aesthetic section of this photo. The paper also presents an Android phone application, that uses this method to improve the aesthetic value of the photos.

This work is the result of my own activity. I have neither given nor received unauthorized assistance on this work.

JULY 2017

KÁLMÁN NOÉMI-ILDIKÓ

ADVISOR:
ASSIST PROF. DR. BODÓ ZALÁN

BABEŞ-BOLYAI UNIVERSITY CLUJ-NAPOCA
FACULTY OF MATHEMATICS AND INFORMATICS
SPECIALIZATION: COMPUTER SCIENCE

License Thesis

**Intelligent photo analyzer by composition
rules**



SCIENTIFIC SUPERVISOR:

ASSIST PROF. DR. BODÓ ZALÁN

STUDENT:

KÁLMÁN NOÉMI-ILDIKÓ

JULY 2017

UNIVERSITATEA BABEŞ-BOLYAI, CLUJ-NAPOCA
FACULTATEA DE MATEMATICĂ ȘI INFORMATICĂ
SPECIALIZAREA INFORMATICĂ

Lucrare de licență

**Analizarea inteligantă a fotografiilor după
reguli de compoziție**



CONDUCĂTOR ȘTIINȚIFIC:
LECTOR DR. BODÓ ZALÁN

ABSOLVENT:
KÁLMÁN NOÉMI-ILDIKÓ

IULIE 2017

BABEŞ-BOLYAI TUDOMÁNYEGYETEM KOLOZSVÁR
MATEMATIKA ÉS INFORMATIKA KAR
INFORMATIKA SZAK

Licensz-dolgozat

**Intelligens képelemzés kompozíciós
szabályok alapján**



TÉMAVEZETŐ:

DR. BODÓ ZALÁN,
EGYETEMI ADJUNKTUS

SZERZŐ:

KÁLMÁN NOÉMI-ILDIKÓ

2017 JÚLIUS

Tartalomjegyzék

1. Bevezető	3
2. Kompozíciós szabályok	4
3. Beépített szabályok alkalmazása	6
4. Az algoritmus felépítése, megvalósítás lépései	11
4.1. Felhasznált technológiák	11
4.2. Elemek detektálása	11
4.2.1. Célobjektumok azonosítása	11
4.2.2. Vonalak azonosítása	12
4.3. Az esztétikai érték kiszámítása	13
4.4. A legesztétikusabb képrészlet meghatározása	13
4.4.1. A genetikus algoritmus paraméterei	15
5. Az alkalmazás felépítése	17
5.1. Szerkezete	17
5.2. Felhasznált technológiák	17
5.3. Az Android kliens bemutatása	18
5.4. A szerver bemutatása	20
6. 5. Esettanulmányok, észrevételek	22
7. Következtetések, továbbfejlesztési lehetőségek	29

1. fejezet

Bevezető

A művészek a történelem során mindenkorra törekedtek, hogy esztétikát csempésszenek műveikbe. Az esztétikus alkotások hatnak az érzékeinkre, ezáltal érzelmeket idéznek elő. Egy ilyen mű megalkotása azonban kihívást jelenthet, ugyanis nincs egy egyetemesen elfogadott esztétikai értékrend. Az esztétika megítélése szubjektív, az aktuális érzelmektől és a személyes ízléstől is nagyban függ. Az előbb említett kihívások ellenére megalakult egy tudományos irányzat, a számítógépes esztétika. Ez arra öszponosít, hogy megjósolja, egy bizonyos műalkotás milyen érzelmeket vált ki, milyen hatással lesz az alanyra, e mellett célja, hogy erősítse, javítsa a kiváltott hatást. A számítógépes esztétika tehát olyan számítástechnikai eljárások összessége, melyek az emberi szemhez, fülhöz hasonlóan képesek jól alkalmazható esztétikai döntéseket hozni [Hoenig, 2005].

Ez az irányzat minden művészeti ággal kapcsolatba hozható, így a zene, irodalom és festészet mellett a fényképészettel is alkalmazva van. A dolgozatban a fényképészettel fellelhető esztétikai tulajdon-ságokkal foglalkozunk, azon kompozíciós szabályokra fektetve a hangsúlyt, melyek betartásával egy sokkal kellemesebb látványt nyújtó képet kapunk.

A kompozíciós szabályok olyan tanácsokat foglalnak magukba, melyek betartásával sokkal sikeresebb képeket lehet készíteni. Azonban az átlagember nem rendelkezik különösebb tapasztalatokkal a fényképész terén, nincs meg a megfelelő képzettsége, ennél fogva a kompozíciós szabályokat sem ismeri. Ellenben mindenki szereti megörökíteni a számára fontos eseményeket, utazásai során látott tájakat, gyakorlatilag minden, ami valamiért fontos számára. Hasznos lehet tehát egy olyan alkalmazás, mely képes azonosítani a kompozíciós szabályokat, javítani a kép esztétikai értékén, ezáltal segítséget nyújtani a felhasználónak. Itt jön képbe a számítógépes esztétika használata.

A dolgozatban vizsgáljuk ezen kompozíciós szabályokat, a legreprezentatívvabbakat beépítjük a rendszerbe, majd ezek felhasználásával egy bemeneti képnek meghatározzuk a számszerűsített esztétikai értékét. Ezután egy kellőképpen gyors módszert keresünk a bemeneti kép egy olyan részképének meghatározására, amely nagyobb esztétikai értékkel rendelkezik, tehát jobban betartja a kompozíciós szabályokat. Egy olyan Android okostelefon alkalmazást is bemutatunk, mely ezt a módszert használja, illetve az ehhez tartozó web service szervert. A dolgozat lezárásként esettanulmányokkal és észrevételekkel demonstráljuk az algoritmus működését, végül következtetéseket vonunk le ezzel kapcsolatban.

2. fejezet

Kompozíciós szabályok

Kompozíció alatt a fényképen található elemek pozícióját, ezek elrendezésének egyensúlyát értjük. E mellett magába foglalja a képek színezetét és textúráját is [Peterson]. Ezen szabályok betartása különbözteti meg az amatőr felvételeket az esztétikus képektől. Egy jól összeállított kép elérése érdekében nem árt előre eltervezni, hogy az elemek pontosan hova is fognak majd kerülni a kereten belül. Fontosságuk ellenére nehéz meghatározni a kompozíciós szabályokat. A művészeti lényege ugyanis a kreativitásban, egyediségen rejlik, éppen ezért nem kell a kompozíciós szabályokat sem kőbe véssett törvényként kezelní. Szabályok helyett tehát sokkal inkább nevezhetőek tanácsoknak, melyeket jó megfogadni, ha egy igazán sikeres és maradandó képet szeretnénk alkotni [Peterson]. Előfordulhat az is, hogy két szabály ellentmond egymásnak, ezért választani kell, melyik teszi előnyösebbé a képet.

A legalapvetőbb és legszéleskörűbben ismert a harmadolás szabály. A leggyakoribb hiba az amatőr fotózásban az, hogy minden elem a kép közepére kerül. Ellenben az emberi szem hajlamos jobban felfigyelni arra a képre, mely harmadokra van osztva, és a célobjektum a harmadvonalak metszetében vagy ezen vonalak mentén helyezkedik el [O'Carroll]. Előnyös például egy emberi alakot vagy fát a kép jobb vagy baloldali harmadába helyezni, illetve a látóhatárt az alsó vagy felső harmadvonal mentére pozicionálni. A legtöbb kamerában van már lehetőség harmadolt nézetben készíteni képeket, segítséget nyújtva az amatőr fényképészeknek.

Az előző szabállyal ellentmondóan néha igenis előnyös, ha a célpont a kép közepére kerül. A szimmetrikus összeállításokat érdemes a kép közepére helyezni [O'Carroll]. Az utak, vagy épületek, esetleg a vizen visszatükröződő objektumok fényképezésekor hasznos alkalmazni ezt a szabályt.

A harmadvonalak mellett az átlók is esztétikailag fontos szerepet játszanak. Egy átló menti célobjektum dinamikát kölcsönöz a képnak, míg az átlók környékén található vonalak a célobjektum felé irányítják a tekintetet. A vízszintes vagy függőleges vonalak stabilitást kölcsönöznek, míg az átlók dinamikus feszültséget [O'Carroll]. Ez egyedivé teheti a képet.

Az egyensúly fogalma is központi szerepet játszik a képösszeállítás harmóniájában. Egy kiegyensúlyozott képen az objektumok egyenlő távolságra vannak elosztva a kép középpontja körül. Egy nem megfelelőképpen kiegyensúlyozott kép zavart okozhat, túlzsfolt hatást kelthet [Peterson].

Közismert, hogy a fényképek két dimenzióban örökítik meg a minket körülvevő világot. Bizonyos technikákkal azonban háromdimenziós hatást is kelthetünk. Elérhetjük ezt a közeli és távoli objektumok szétválasztásával, úgy, hogy kihangsúlyozzuk a közelebbi objektumokat, míg a távolabbi, háttérben lévő elemeket kissé összemossuk. Így mélységérzetet lehet kelteni, illetve így a figyelem központjába a közeli

2. FEJEZET: KOMPOZÍCIÓS SZABÁLYOK

célobjektum kerül, nem pedig a színes, túlzsúfolt háttér. Ez a mélységellesség megvalósítható hagyományos módon, DSLR gépet használva, megfelelőképpen beállítva az objektív gyújtótávolságát, illetve az élesség távolságát. E mellett mélységellességet elérhetünk megfelelő szoftver használatával, úgy, hogy a háttérben levő elemeket elhomályosítjuk [István].

A háromdimenziós hatás keltésére alkalmas még a keret a keretben alkalmazása. Ilyenkor ideális egy ablakot, ajtót, boltívet vagy faágakat fotózni, úgy, hogy ezek körülöleljék a képet, mintegy keretet alkotva a célobjektum körül.

Megfigyelt tény, hogy az emberek természetes módon vonzódnak a mintákhoz és különböző textúrákhoz. Vonzóak a szem számára és harmóniát sugallnak [Rowse]. Az ismétlődő minták minden jó módszert biztosítanak a kellemes képek készítésére.

Az elemek számát tekintve a páratlanság szabálya alapjául az a megfigyelés szolgál, hogy az emberi szemnek kellemesebb látványt nyújt a páratlan számú objektumot tartalmazó kép. A páros számú elemet tartalmazó elbizonytalaníthatja a megfigyelőt, elvonja a figyelmet az objektumról, az alany nem tudja mire koncentráljon. A páratlan szám sokkal természetesebb hatást kelt.

Bizonyos esetben az előnyös, ha a célobjektum kitölti a teljes keretet, néha jobb hatást érünk el, ha hagyunk a célpont körül egy kis negatív teret. Azzal, hogy kitöljük a teljes teret segítjük a nézőt, hogy maximálisan a célontra összpontosítson, megfigyelje a részleteket. Ezzel ellentében az is nagyon látványos lehet, ha sok semleges helyet hagyunk a fő objektum körül [Peterson]. Ez is az objektumra irányítja a tekintetet, s egyben ki is emeli a háttérből.

Ezen kompozíciós szabályok egytől egyig elősegítik az esztétikus képek készítését. A kihívás abban rejlik, hogy megtaláljuk az egyensúlyt a különböző szabályok alkalmazásában. Olyan szabályokat kellett tehát keresnünk, amelyek nem mondanak ellent egymásnak. Dolgozatunkban a harmadolás, vizuális egyensúly és átló dominancia szabályok kerültek alkalmazásra, ezek összetételével próbáltuk megfelelően számszerűsíteni egy kép esztétikai értékét.

3. fejezet

Beépített szabályok alkalmazása

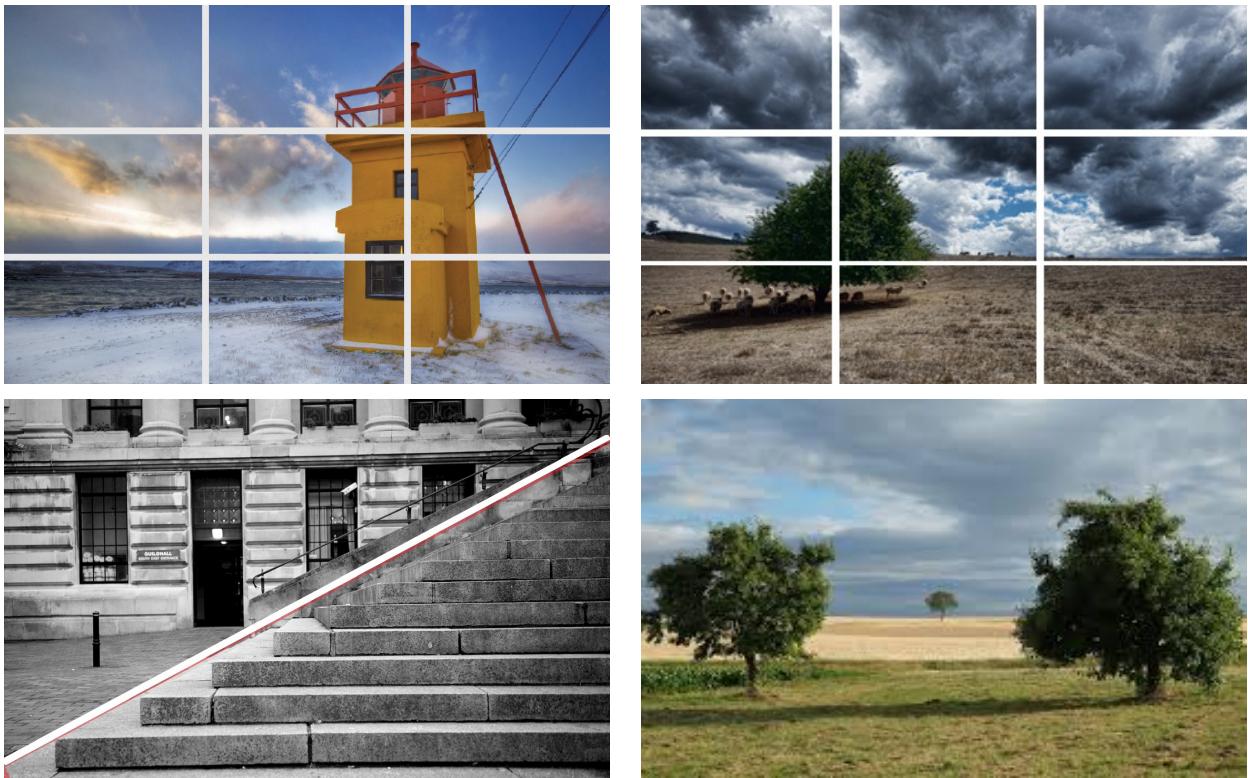
Ahhoz, hogy egy képnek számszerűsíteni lehessen az esztétikai értékét, fontos az előbb említett szabályok közül néhányat kiválasztani. Mivel egyes szabályok ellentmondanak egymásnak, olyan szabálycsoportot kellett kiemelnünk ezen halmazból, melyek nem csak jól megférnek egymás mellett, de együttesen egy sokkal esztétikusabb, kellemesebb képet adnak.

A képek esztétikájának megfelelő tükrözéséhez ezen kompozíciós szabályok közül háromat választottunk ki: a harmadolás szabályt, az átlók dominanciáját, illetve a vizuális egyensúlyt [6]. A harmadolás a legalapvetőbb szabály, minden sikeres műalkotásban fellelhető az alkalmazása, ezért ez az általunk kiválasztott csoportba is bekerült. Az átlós vonalak megfelelően irányítják a tekintetet, ezzel hatást gyakorolva az emberi érzékekre, a fő célpontra terelik a figyelmet, így a néző nem érezheti semmit mondónak a képet. Emellett az emberi lélek igényli az egyensúlyt, így a képekben is hajlamos az egyensúlyt részesíteni előnyben. A rendezetlen, túlzsúfolt, majd lappangó területeket tartalmazó képek zavart okozhatnak, a néző elveszítheti a kép mondandóját. Ezért esett a választás, az előbb említett két szabály mellett, a vizuális egyensúly beépítésére is.

A fejezetben látható képek a felhasznált szabályokat mutatják be. Az 3.1-es ábra első képe a harmadolás szabályt reprezentálja. Jól látható, hogy a világítótorony a jobboldali harmadvonal mentén helyezkedik el, minden két súlypontot magába foglalja. A 3.1-es ábra második képén pedig a horizont az alsó harmadvonallal teljesen megegyezik. Az ábra bal alsó képe az átlódominanciát mutatja, ugyanis a lépcső szélei pontosan az átlót alkotják. Az utolsó, jobb alsó ábra pedig a vizuális egyensúly szabályát tartja be, a fák egyenlő távolságra helyezkedve el a kép középpontjától.

Ahhoz, hogy a kiválasztott szabályokat megfelelően alkalmazni tudjuk a képértékelésben, bizonyos metrikákat kellett felhasználnunk. A céljainknak legmegfelelőbbnek az L. L. Renjie, C. L. Wolf és D. Cohen-Or által kidolgozott metrikák [Ligang Liu, 2010] bizonyultak. Kisebb változtatásokkal ezen metrikákat implementálva alakult ki az esztétikai értéket számszerűsítő alkalmazás.

3. FEJEZET: BEÉPÍTETT SZABÁLYOK ALKALMAZÁSA



3.1. ábra. Az általunk felhasznált kompozíciós szabályok. A felső kettő a harmadolás, a bal alsó az átlódominanciát, míg az utolsó a vizuális egyensúlyt mutatja be.

forrásokat betenni!!!!!!!!!!!!!!

A képhez rendelt esztétikai értéket a célobjektumok és fő vonalak pozíciója valamint térbeli felépítése alapján határoztuk meg. Ehhez azonban először azonosítanunk kellett ezen számottevő objektumokat. A metrikák implementációját tehát a képen fellelhető objektumok detektálása kellett megelőzze, melyhez a jól ismert algoritmusokat alkalmaztuk.

A célobjektumok azonosításához előbb egy sor képfeldolgozó algoritmust alkalmaztunk, mely megfelelően összemossa a képet, ezzel eltüntetve a felesleges éleket, fekete fehérré alakítottuk, majd megfelelő küszöbértéket alkalmazva 0 és 1 közé redukáltuk a pixelek értékeit. Ezután egy Canny operátor [Can] alkalmazva azonosítottuk a képen található fő kontúrokat. Ha mindez megtörtént, a kontúrokat magukba foglaló téglalapok képezik a kép objektumait. Ez azonban igen sok bekeretezett elemet eredményez. Sok közülük egymással átfedésbe is kerülhet, ezáltal egy objektumot akár több téglalappal is lefedhetünk, ami felesleges. Ennek elkerülése érdekében az egymással átfedésben levő téglalapokat egybeolvasszuk, így kevesebb darab, nagyobb területű célobjektumot kapva eredményül.

A kép túlzott összemosása miatt fontos objektumokat figyelmen kívül hagyhatunk, ami gondot okozhat. Ilyen gyakran előfordul az arcokat tartalmazó képek esetén, annak ellenére, hogy az emberi arc fontos szerepet játszik az elkészített képekben. Az ilyen hibák elkerülése végett a képen egy arcdetektálást is végeztünk, hogy ezek mindenkorán bekerüljenek a talált objektumok közé. A felfedezett arcokat

3. FEJEZET: BEÉPÍTETT SZABÁLYOK ALKALMAZÁSA

nagyobb súllyal ellátva adtuk hozzá a főobjektumok listájához.

A vonalak azonosítása is képfeldolgozó algoritmusok alkalmazásával valósítható meg. Első körben Canny éldetektáló operátort [Ope] használtunk, úgy, hogy előtte nem mostuk különösebben össze a képet. A vonalak keresésekor fontos, hogy minél elesebb képpel dolgozzunk, ugyanis összemosás esetén értékes információt veszíthetünk, ezért előfordulhat, hogy bizonyos vonalak nem lesznek megtalálva. E lépés után egy Hough-féle vonaldetektáló algoritmust alkalmaztuk. Ezt megfelelően paraméterezve eredményül meg is kaptuk a képben fellelhető vonalak halmazát. Ez azonban igen sok vonalat hajlamos magába foglalni, ami az értékelő algoritmusra nézve nem előnyös, túl sok számítást vonhat maga után, ami időigényessé teheti az egész folyamatot. Ennek elkerülése érdekében a vonalak közül csak az első n darab, legnagyobb súllyal rendelkező vonalat tartottuk meg. Egy vonal minél hosszabb, annál nagyobb a súlya.

Az általános vonalak mellett számon kellett tartanunk az átlóhoz közel eső vonalakat is. Az átló közeliséget nem a vonal pozíciójára értjük, hanem arra, hogy mennyire áll az átlóhoz hasonló szögben. Az átlók szögéhez viszonyítva 10o-os eltérést engedünk meg. Az átlós vonalak listáját az átlódominancia szabály vizsgálata miatt kellett eltárolnunk.

Ismerve a főobjektumokat és vonalakat, egy olyan 0 és 1 közötti értéket definiálhatunk, mely kellőképpen mutatja, hogy egy kép milyen szinten tartja be a fent említett szabályokat.

A dolgozatban használt szimbólumok jelentései a 3.2 ábrán fel vannak tüntetve. Az algoritmusainkban a távolság alatt a hagyományos két pont közötti távolság helyett minden a normalizált Manhattan távolságot értjük, melyet a következőképpen definiálhatunk : $d_M((x_1, y_1), (x_2, y_2)) = |x_1 - x_2|/w + |y_1 - y_2|/h$, illetve $d_{LM}(L, M)$ nem más, mint két vonal, L és M közti távolság. Két vonal közötti távolságot úgy számoljuk ki, hogy L két végpontjától kiszámítjuk az M legközelebbi pontjához mért Manhattan távolságát, majd ezt átlagoljuk.

Szimbólum	Jelentése
w, h	a kép szélessége és hosszúsága
C	a kép középpontja
$G_i, i = 1, 2, 3, 4$	a négy súlypont
$R_i, i = 1, 2, 3, 4$	a négy harmadvonallal
$S_i, i = 1, 2, \dots, n$	a detektált objektumok
$C(S_i), I(S_i)$	az objektum középpontja és súlya
$r(S_i)$	objektum méret - képméret arány
Q_1, Q_2	az átlók
$L_i, i = 1, 2, \dots, n$	a detektált vonalak
X	átłós vonalak halmaza
\bar{X}	nem átlós vonalak halmaza
$I(L_i)$	a vonal súlya
d_M	normalizált Manhattan távolság
d_{LM}	két vonal közötti távolság

3.2. ábra. A képletekben használt szimbólumok

3. FEJEZET: BEÉPÍTETT SZABÁLYOK ALKALMAZÁSA

$I(S_i)$ alatt az aktuális, i-edik objektum súlyát értjük, ami nem más, mint az objektum területe, míg $I(L_i)$ az i-edik vonal súlya, azaz ennek a hossza. \bar{X} az összes átlagos vonalat jelöli, míg X az összes átlós vonalat, azaz azon vonalakat melyek legfeljebb 10° -ban térnek el a Q_1 és Q_2 által jelölt átlók valamelyikétől.

A harmadolás szabály (Rule of Thirds, RT) kiszámítása két fő komponensből tevődik össze : E_{Point} a célobjektumoknak a súlypontoktól mért távolságát méri, míg az E_{Line} a vonalaknak a harmadvonalaktól mért távolságát tükrözi. A képletben szereplő γ_{Point} és γ_{Line} súlyok. Az összképlet ezen komponensekből a következő képpen épül fel [Ligang Liu, 2010]:

$$E_{RT} = \gamma_{Point} E_{Point} + \gamma_{Line} E_{Line}.$$

A pont értéke az összes célobjektumnak :

$$E_{Point} = \frac{1}{\sum_i I(S_i)} \sum_i I(S_i) e^{-\frac{D^2 S_i}{2\sigma_2}},$$

ahol $D(S_i)$ az aktuális célobjektumnak a négy, súlyponthoz mért minimális távolságát jelenti, $D(S_i) = \min_{j=1,2,3,4} d_M(C(S_i), G_j)$, illetve σ_2 konstans egyenlő 0.17

A vonal értéke az összes megtalált, nem átlós vonalnak :

$$E_{Line} = \frac{1}{\sum_{i \in \bar{X}} I(L_i)} \sum_{i \in \bar{X}} I(L_i) e^{-\frac{D_R^2 L_i}{2\sigma_1}},$$

ahol $D_R(L_i)$ a minimum vonaltávolság az aktuális vonaltól a négy harmadvonalhoz mérve, $D_R(L_i) = \min_{j=1,2,3,4} d_L(L_i, R_j)$ illetve a σ_1 konstans egyenlő 0.17 .

Kísérletek azt bizonyították, hogy a vonalak nagyobb szerepet játszanak az esztétikai érték kialakulásában, ezért nálunk is nagyobb súlyt kap a vonalérték. Ennek megfelelően a képletben szereplő $\gamma_{Point} = 1/3$, míg a $\gamma_{Line} = 2/3$ -nak választottuk meg.

A vizuális egyensúly (Visual Balance, VB) -ról akkor beszélhetünk, ha a célobjektumok összességének középpontja közel található a képnek a középpontjához. Ennek megfelelően a vizuális egyensúly értékének kiszámítása [Ligang Liu, 2010]:

$$E_{VB} = e^{-\frac{d_{VB}^2}{2\sigma_3}},$$

ahol $d_{VB} = d_M(C, \frac{1}{\sum_i I(S_i)} \sum_i I(S_i) C(S_i))$, $s\sigma_3 = 0.2$.

Az átlódominancia szabály (Diagonal Dominance, DD) értékének kiszámítása a vonalérték kiszámításához teljesen hasonló módon történik, annyi különbséggel, hogy itt csak az átlós vonalakkal dolgozunk. Ennek megfelelően [Ligang Liu, 2010]:

$$E_{DD} = \frac{1}{\sum_{i \in X} I(L_i)} \sum_{i \in X} I(L_i) e^{-\frac{D_Q^2 L_i}{2\sigma_2}},$$

3. FEJEZET: BEÉPÍTETT SZABÁLYOK ALKALMAZÁSA

ahol $D_Q(L_i) = \min(d_L(L_i, Q_1), d_L(L_i, Q_2))$.

Az esztétikai érték függvény (Aesthetic score function, A) a fennebb említett szabályok által kiszámított értékek kombinációja [Ligang Liu, 2010]:

$$E_A = \frac{\varpi_{RT} E_{RT} + \varpi_{VB} E_{VB} + \varpi_{DD} E_{DD}}{\varpi_{RT} + \varpi_{VB} + \varpi_{DD}},$$

ahol $\varpi_{RT} = 1.0$, $\varpi_{VB} = 0.3$, $\varpi_{DD} = 1.0$.

A célobjektum mérete (Salient-region Sizes, SS) is fontos tényező az összetett esztétikai érték számolásában. Olyan esetben, amikor egy képen csak egy szabályt ismerünk fel, a fent említett összefüggés nem írja le elég pontosan az adott kép esztétikai értékét. Előfordulhat, hogy az adott kép több részképe megegyező esztétikai értéket kap, ezáltal nem lehet objektíven eldönteni azt, hogy melyik a legsikeresebb ezen képrészletek közül. Példának okáért vegyük azt az esetet, amikor a kép csak egy célobjektumot tartalmaz, akkor azt a négy súlypontra helyezve ugyanazt az esztétikai értéket kapnánk. Ez gondot okozhat. Ennek elkerülése érdekében vezettük be a célobjektum vagy célobjektumok méretét, mint értékelési szempontot.

Esztétikus képek tanulmányozása során észrevehető, hogy a célobjektumok három csoportba sorolhatók : kis, közepes és nagy méretűek. A dolgozatban előnyben részesítjük azon objektumokat, melyek ezt az eloszlást követik. A három méretnek az alábbi arányok felelnek meg: $r_1 = 0.1$, $r_2 = 0.56$ és $r_3 = 0.82$. Az átlunk használt méret-függvény az ehhez konvergáló arányokat támogatja [Ligang Liu, 2010]:

$$E_{SS} = \sum_i \max_{j=1,2,3} e^{-\frac{(r(S_i) - r_j)^2}{2\tau_j}},$$

ahol $\tau_1 = 0.07$, $\tau_2 = 0.2$, $s\tau_3 = 0.16$, illetve $r_1 = 0.1$, $r_2 = 0.56$, $s r_3 = 0.82$.

Az összetett esztétikai érték (Combined Aesthetic Score) a kiszámított E_A és E_{SS} értékekből tevődik össze [Ligang Liu, 2010]:

$$E = (1 - \omega_{SS})E_A + \omega_{SS}E_{SS},$$

ahol $\omega_{SS} = 0.08$, ez az érték azért olyan kicsi, mivel a hangsúly az esztétikai értéken van, a célobjektum mérete csak másodlagos, a határozatlan esetek elkerülése végett lett beépítve a rendszerbe.

Elvégezve az objektumokat azonosító lépéseket, illetve alkalmazva a fent említett képleteket, eredményül egy 0.0 és 1.0 közötti értéket kapunk. Optimális esetben a kép esztétikai értéke 1.0 lenne, ez a gyakorlatban 100% -nak felelné meg.

4. fejezet

Az algoritmus felépítése, megvalósítás lépései

4.1. Felhasznált technológiák

Az algoritmus implementálásához a Java programozási nyelvet választottuk, míg a képelemző algoritmusok használatához az OpenCV nyílt forráskódú könyvtárat. Ez a könyvtárcsomag teljesen szabadon felhasználható. Az OpenCV használatára azért esett a választás, mivel ez az alapvető képfeldolgozási algoritmusokat magába foglalja, nagyon gyors és hatékony megoldásokat biztosítva. Alkalmazásunk sok számítást kell végezzen, ezért szükségeltetett a performanciát szem előtt tartani. Bár a könyvtárcsomagot eredetileg C/C++ -ban írták, készítettek hozzá úgynevezett wappereket, melyek segítségével más nyelvekben is használható, többek között Java-ban is.

Algoritmusunk három fő részre bontható: az első a képen található elemek azonosítása; a második az esztétika számszerűsítése, a szabályoknak megfelelő metrikák implementációja; a harmadik pedig a legesztétikusabb képrészlet megtalálása.

4.2. Elemek detektálása

A képen található elemek azonosítása a célobjektumok, arcok, illetve vonalak detektálását takarja. Ezekre szükségünk van a második lépésnél, az esztétika számszerűsítésénél. Az elemek azonosítása a legidőigényesebb, ezért ezt a lépést az elemzés során célszerű csak egyszer elvégezni. Az azonosított objektumokat és vonalakat listákban tároljuk, hogy az elemzés során ezt bármikor használni tudjuk.

4.2.1. Célobjektumok azonosítása

A célobjektumok azonosításában első lépésként kissé össze kellett mosnunk a képet, ezáltal a felesleges élektől megszabadulni, míg a fontosakat hangsúlyosabbá tenni. Erre a célra a piramis átlagoló szűrőt [10] alkalmaztuk. Az algoritmus egy maxlevel+1 szintet tartalmazó Gauss piramist épít, az összemosó algoritmust mindenkorán a kisebb szinteken alkalmazva először, majd az eredményeket átadva a magasabb szinteknek. Itt újból alkalmazva van az összemosás, de csak azon pixelekre, melyek adott határnál jobban eltérnek a nekik megfelelő alacsonyabb szintű pixelektől. Ez a színes területek körvonalait élesebbé teszi, a kevésbé hangsúlyos éleket pedig teljesen eltünteti. Az algoritmus paramétereit próbáltuk úgy beállítani, hogy a leginkább helyes eredményt adja: az adott pixel körüli vizsgálandó pixelek számát a

4. FEJEZET: AZ ALGORITMUS FELÉPÍTÉSE, MEGVALÓSÍTÁS LÉPÉSEI

kép területének 3%-val tettük egyenlővé, a pixelérték küszöböt 100-ra, míg a piramis szintjeinek számát 2-re állítottuk.

Következő lépésben a képet fekete-fehérré alakítottuk, majd adaptív küszöbértéket (adaptive threshold) [Thr] alkalmazva 0 és 1 közé szorítottuk a pixelek értékét. Mivel a kép különböző pontjai más-más megvilágításban lehetnek, nem lenne célravezető egy globális küszöbértéket használni. Ezért esett a választás az adaptive threshold alkalmazására, ahol kisebb régióknak saját küszöbértéket számolunk, így különböző helyeken más-más küszöbértékkel dolgozunk, mely a fényviszonyuktól függetlenül jobb eredményeket ad.

Ezután az élek felfedezése következik. Erre a lépésre a legalkalmasabbnak a Canny-féle éldetektálás [Can] bizonyult. Ez egy sor algoritmusnak az alkalmazása, mely magába foglalja a kép összemosását, a zavarok kiszűrését, konvolúciós maszk alkalmazását, a nem éleket képező pixelek kiszűrését, illetve utolsó lépésként, két küszöbérték alkalmazását : ha az alsónál kisebb értékűek a pixelek, akkor nem képeznek élt, ha a felsőnél nagyobbak, akkor élt képeznek, illetve ha a kettő között vannak, csak akkor képeznek élt, ha van olyan szomszédos pixelük, mely élhez tartozik. Észrevehető, hogy a kezdetekkor már alkalmaztunk egy összemosást és a Canny-féle algoritmusban is található egy. Azért volt szükséges egy összemosást alkalmazni előre, mivel a Canny algoritmusban nincs megadva a választás lehetősége: Gauss-féle összemosást használ [Canny, 1986], nekünk azonban a piramis átlagoló a megfelelőbb. E lépés eredményeképp a kimeneti kép már csak a kontúrokat tartalmazza.

Ezután már csak külön kellett választani ezen kontúrokat egy pontok halmazát tartalmazó listába, és az adott pontralmaзов egy-egy téglalapba helyezhetőek. Ez még nem jelenti az algoritmus végét, ugyanis az így kapott eredményben egy objektumot több kisebb résztéglalap reprezentálhat. Ezt célszerű volna egy téglalapba tömöríteni. Az egyetlen hátramaradt lépés tehát az egymást metsző vagy átfedő téglalapok egyesítése volt. Az így kapott nagyobb területű téglalapok már a célobjektumokat reprezentálják.

A kép színkombinációjából adóthat olyan probléma, hogy az emberi arc nem lesz célobjektumként azonosítva. Ez probléma, ugyanis az emberi arcok fontos szerepet játszanak a képeken. Annak érdekében, hogy az emberi arcok mindenkorábban bekerüljenek a célobjektumok közé, az eredeti képen alkalmaztuk az OpenCV által kínált arcfelismerést, így az már nem fordulhat elő, hogy az arcok kimerülnek a célobjektumok listájából.

4.2.2. Vonalak azonosítása

A vonalak azonosítása esetén nincs szükség a kép nagy mértékű összemosására. Itt fontosak az eredeti élek, kockázatos volna eltüntetni egyes éleket, hiszen ezzel akár fontos vonalakat is elveszíthetnénk. A vonalkereső algoritmusunk egy Canny-féle élkereséssel kezdődik, ahol a kis és nagy küszöbérték rendre 100 és 200.

A megtalált élek közül a vonalakat egy valószínűségen alapuló Hough-féle vonaltranszformációval kapjuk meg. A Hough-féle transzformációban a vonalak polárkoordinátában vannak leírva. Így egy vonal a következőképpen adható meg: $y = (-\cos \theta / \sin \theta)x + (r / \sin \theta)$, ahonnan egyszerűen kifejezhető : $r = x \cos \theta + y \sin \theta$. Tehát egy (x_0, y_0) ponton áthaladó egyenesek halmazát a (r_θ, θ) párosok képezik.

4. FEJEZET: AZ ALGORITMUS FELÉPÍTÉSE, MEGVALÓSÍTÁS LÉPÉSEI

Algoritmusunkban az $r = 1$ pixel és a $\theta = 1^\circ$, illetve a vonalhoz szükséges minimális metszéspontok száma, a minimális vonalhossz és a maximális vonal közti szakadás a fénykép méreteitől függ. A minimális metszéspontok számának a szélesség és magasság minimumának a 10% -át vettük, a minimális vonalhosszt az átlónak 25% -a, illetve a maximális vonal közti szakadást az átlónak 4% -a képezi. A Hough-féle vonaldetektálás az előbb említett paramétereket használva egy vonallistát ad eredményül. Ez azonban egyes képeknél igen sok, felesleges vonalat is tartalmaz. Éppen ezért a talált vonalaknak csak kis százalékával foglalkoztunk. Kiválasztottunk maximálisan 6 darab legértékesebb vonalat. Egy vonal annál értékesebbnek számít, minél nagyobb a hossza.

A átlódominancia szabály vizsgálatához külön számon kellett tartanunk az átlóközeli vonalakat is. Ennek érdekében kiszámoltuk az átlók szögét, és a kapott vonallistából kerestük azon vonalakat, melyek ezekhez viszonyítva egy bizonyos küszöbön belül helyezkednek el. Ezt a küszöbértéket 10° -ra állítottuk be. Sorra ellenőrizve a vonalakat, a határon belül esőket hozzáadtuk az átlókat tartalmazó listához.

4.3. Az esztétikai érték kiszámítása

Az adott kép esztétikai értékének kiszámításához a fentebb említett metrikákat szükséges implementálni. Az algoritmusunk során nem csupán az eredeti, teljes nagyságú képnek kellett kiszámítani az esztétikai értékét, hanem a javasolt képrészletnek is. Felmerül tehát a kérdés, hogy minden ilyen képrészletre érdemes-e újból megkeresni a célobjektumokat és vonalakat. Performancia szempontjából semmiképp sem. Az objektumok megkeresése időigényes folyamat, ezért nem lenne célravezető minden alkalommal újból es újból azonosítani ezeket. Pontosan ezért volt hasznos az előző lépésekben minden megtalált objektumot listákba elmenteni. Ilyen módon ezeket a későbbiek során bármikor hasznosítani lehet.

Egy képrészlet esetén előfordulhat, hogy bizonyos objektumok vagy vonalak teljesen, vagy csak részben kiesnek a keretből. Ilyen esetben nem dolgozhatunk az eredeti objektumlistával. Az ilyen célobjektumok esetén meg kell keresnünk azt a téglalapot, mely a célobjektum és a keret metszete, ez lesz az aktuális képrészletben használt objektum. Ha egy adott objektum teljesen a kereten kívülre esik, azt nem vesszük számításba az adott értékszámolásban. A kilógó vonalak esetén hasonlóképpen járunk el. Az a vonalrészlet lesz az aktuálisan használt vonal, mely belőg a képbe, vagy adott esetben, ha a vonal teljesen a kereten kívül esik, akkor ezt nem használjuk az értékszámolásban. Egy másik opció az lett volna, hogy ha egy objektum bizonyos része kiesik a keretből, akkor az a keret érvénytelennek tekinthető legyen, vagy az adott objektumot ne tekintsük célobjektumnak. Ezt a lehetőséget azonban elvetettük, ugyanis megfigyeltük, hogy egy kép lehet esztétikusabb, művészibb akkor, ha az adott objektumnak csak egy része található az aktuális keretben.

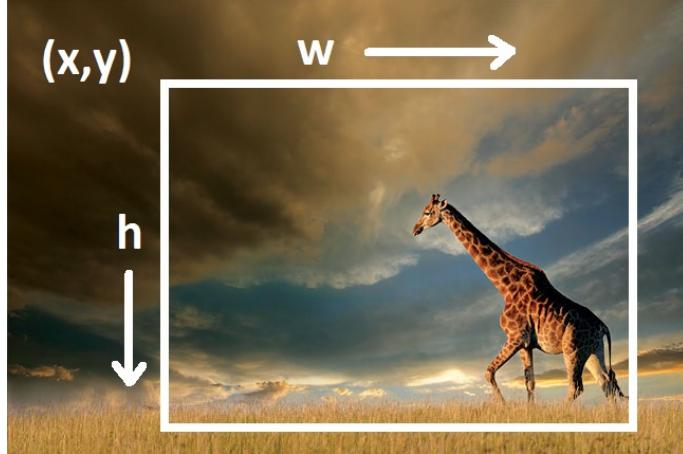
4.4. A legesztétikusabb képrészlet meghatározása

A dolgozatban tárgyalt algoritmus utolsó lépését a legesztétikusabb képrészlet meghatározása képezi. Ezt a képoptimalizációs lépést keresési algoritmussal valósítjuk meg. Egy kép azonban nagyon sok lehetséges részképpel rendelkezhet. Egy ilyen lehetséges képrészletet 4 paraméterrel írhatunk le: a képrészlet

4. FEJEZET: AZ ALGORITMUS FELÉPÍTÉSE, MEGVALÓSÍTÁS LÉPÉSEI

bal felső sarkának x és y koordinátája, illetve a képrészlet szélessége és hosszúsága, (x_i, y_i, w_i, h_i) , ahogyan azt a 4.1 ábra is mutatja.

4.1. ábra. A képrészlet paraméterei



Egy kép esetén tehát $x_i \in [0, w]$, $y_i \in [0, h]$, $w_i \in [0, w]$ és $h_i \in [0, h]$ intervallumok jöhetnek szóba, ami nagyon nagy számú lehetséges kombinációt adna. Ez egy 480x320-as kép esetén, ami nem számít nagy felbontásúnak, $480^2 * 32^2 \approx 32\text{milliard}$ volna. Ez nagyon hosszú időt venne igénybe, ugyanis ennyiszer kellene az értékelő algoritmust végrehajtani.

A képoptimalizációs algoritmus első lépéseként tehát csökkentenünk kellett a lehetséges esetek számát. Ezt a legegyszerűbben az intervallumok méretének csökkentésével érhetjük el. A kép minőségének megőrzése is fontos szempont, ezért a képet legfeljebb az eredeti területének negyedére csökkentjük. Ez nagyon kis felbontású képek esetén már kellemetlen hatást okozhat, azonban algoritmusunkban feltételezzük a nagy felbontású képek használatát. Az algoritmusban használt intervallumok ennek megfelelően a következőképpen változnak : $x_i \in [0, w/2]$, $y_i \in [0, h/2]$ illetve $w_i \in [w/2, w]$. Szintén segítségünkre váthat az, ha megtartjuk a kép eredeti arányait. Ennek megfelelően elegendő a kép szélességét és a szélesség/hosszúság arányt ismernünk, ebből ugyanis egyszerűen kiszámítható a hosszúság.

Az intervallumok csökkentésével a lehetséges kombinációk számát is nagyon lefaragtuk, azonban ez még mindig nem elég. A lehetséges részképek száma még mindig túl nagy ahhoz, hogy ezeknek sorra kiszámoljuk az esztétikai értékét. Szükség van tehát egy olyan, eléggé gyors módszerre, mely megoldja az optimizációs problémákat. Erre legalkalmasabbnak a genetikus algoritmus használatát találtuk.

A genetikus algoritmusok olyan keresőalgoritmusok, melyek a genetikára és a természetes szelekcióra épülnek. Általában optimális vagy az optimálishez közel álló megoldások megtalálására használják, olyan problémák esetén melyeket nagyon hosszú időbe telne megoldani a hagyományos programozási algoritmusokkal. Az optimizációs problémák esetén az a kihívás, hogy megtaláljuk azon bemeneti értékeket, melyre a legjobb étéket kapjuk [Wei Luo, 2011]. Esetünkben keressük azon (x_i, y_i, w_i) számhármat, mely a legnagyobb esztétikai értékkel rendelkezik. Ez a számhármas reprezentál tehát egy egyedet. Ilyen egyedekből építjük fel a populációt, ezen hajtjuk végre a genetikus algoritmus függvényeit: a kiválasztást, a keresztezést és a mutációt.

4. FEJEZET: AZ ALGORITMUS FELÉPÍTÉSE, MEGVALÓSÍTÁS LÉPÉSEI

A genetikus algoritmusok problémáról problémára változhatnak, de a következő részeket minden tartalmazniuk kell: inicializáció, szelekció, szaporítás és leállás. Az inicializáció során a kezdeti populációt generáljuk. Ennek mérete feladattól függően változó. Általában a egyedek a keresési téren belül egyenletesen oszlanak el. Néha hasznosabb a jónak vélt egyedekkel inicializálni a kezdeti populációt, ez viszont a globális helyett lokális optimum megtalálásához vezethet [Michalewicz, 1992].

Minden generációban az egyedek egy része kiválasztásra kerül. Ezek általában a fitnesz függvény értéke szerint lesznek kiválasztva. Bizonyos genetikus algoritmus változatokban minden egyed kiértékkelésre kerül, s a legjobb értékűek lesznek szelektálva. Ez a teljes folyamat azonban túlságosan hosszú lehet, így előnyössé válik a véletlen egyed kiválasztása.

Az újabb egyedek bevezetése kereszteződéssel, illetve mutáció műveletteset lehetséges. A kereszteződés esetén két meglévő egyedből hozunk létre egy újat, míg a mutáció esetén egy egyedet módosítva kapunk újabbat.

Végül, a genetikus algoritmusok egy leállási feltétel teljesülésig futnak. Ez a feltétel lehet egy előre rögzített maximális generációszám, illetve a maximális fitneszérték elérése.

4.4.1. A genetikus algoritmus paraméterei

Algoritmusunkban a kezdőpopuláció létszámanak 200-at választottunk, hogy minél nagyobb valószínűséggel kapjuk már első körben egy, az eredetinél nagyobb esztétikai értékkal rendelkező képet, illetve hogy minél változatosabb legyen a kezdeti populációink. Az algoritmus evolúciójának leállási feltételeinek 500 generációt állítottunk be. Természetesen az algoritmus akkor is leállási ponthoz érkezne, ha megtalálná a maximálisan esztétikus képet, amelynek értéke 1.0 lenne. Kísérletek elvégzése után, illetve figyelembe véve, hogy nincs olyan kép mely minden szabálynak tökéletesen megfelelne, biztosan elmondható, hogy egy kép esetén sem fogunk maximális értéket kapni. Éppen ezért volt szükséges egy leállási feltételnek megfelelő maximális generációszámot megadnunk.

A genetikus algoritmusban bajnokság (tournament) szelekciót alkalmaztunk. Ez a szülők megválasztási módja. Szülőválasztáskor tehát véletlenszerűen kiválasztunk 20 darab egyedet, ezek közül megkeressük a legnagyobb esztétikai értékkel rendelkezőt, ez lesz az egyik szülő. A másik szülő választásánál is hasonlóképpen járunk el. E két szülőt megfelelően keresztezzük : mindhárom gén, tulajdonság esetén (x , y és szélesség) generálunk egy számot. Ha ez a szám kisebb mint a 0.5 - os küszöbérték, akkor az első, különben a második egyed tulajdonságát örököli meg a gyermek. Ha a generálás során a kapott keret kilógna a képből, akkor addig választjuk újra a szülőket, majd keresztezzük, amíg a kapott keret teljesen az eredeti képen belül nem lesz.

A mutáció során hasonlóképpen járunk el. Véletlenszerűen választunk egy értéket. Ha ez kisebb mint a mutációs küszöbhatár, akkor a megfelelő gént újrageneráljuk. Ha a kapott mutált egyed nem található a képen belül, akkor addig generáljuk újra, míg olyan keretet nem kapunk, mely teljesen az eredeti képen belül helyezkedik el. Az általunk használt küszöbhatár 0.3. A mutáció beépítése azért fontos, hogy elkerüljük a lokális optimumok megtalálását, célunk ugyanis az összes lehetőség közül a legjobbat, azaz a globális optimumot megtalálni.

4. FEJEZET: AZ ALGORITMUS FELÉPÍTÉSE, MEGVALÓSÍTÁS LÉPÉSEI

Fontos megemlíteni, hogy a genetikus algoritmusban elitista szelekciót alkalmaztunk tehát előnyben részesítettük az eliteket, azon egyedeiket, melyek nagy esztétikai értékkel rendelkeznek. minden generációban kiválasztottuk tehát a legesztétikusabb részképet, majd első lépésként ezt hozzá is adtuk a következő generáció egyedeihez. Ezen módszerrel elérve a maximális generációs számot és kiválasztva a legesztétikusabb képrészletet, ha nem is a legoptimálisabb, de jelentősen jó megoldást kaptunk viszonylagosan rövid idő alatt.

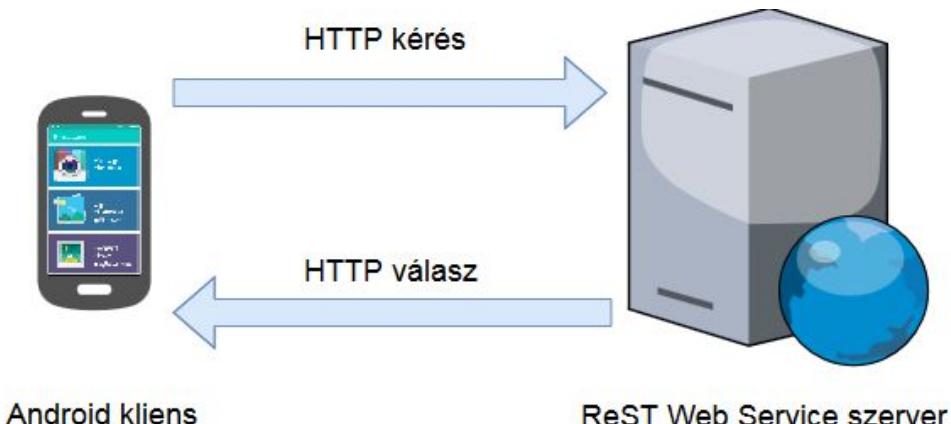
5. fejezet

Az alkalmazás felépítése

Annak érdekében, hogy algoritmusunk könnyen felhasználható és alkalmazható legyen a minden nap életben, szükségessé vált egy barátságos felhasználói felületet készítenünk hozzá. Erre a legalklamsabbnak egy telefonos alkalmazás tűnt. Így a felhasználó a kép készítése után rögtön elemezheti is a képet, s ha az nem bizonyul elég sikeresnek, az alkalmazás segítségével optimizálhatja, s így már egy kész remekművel dicsekedhet.

Mivel az algoritmusunk meglehetősen erőforrásigényes, nem bízhattuk ezt a munkát a felhasználók okostelefonjára. Ennek érdekében esett a választás a ReST alapú kommunikáció alkalmazására. Ezért egy szerver alkalmazás végzi a képelemző, képertékelő feladatokat, míg a kliens csak kéréseket küld a szerver felé, illetve feldolgozza, és megjeleníti a kapott választ.

5.1. Szerkezete



5.1. ábra. Az alkalmazás szerkezete

5.2. Felhasznált technológiák

A kliens alkalmazást Android okostelefonokra terveztük, ugyanis széleskörűen elterjedt, egyben ez rendelkezik a legnagyobb felhasználói bázissal. Fontos szempont volt tehát az, hogy minél több felhasználó számára elérhető legyen ez az alkalmazás. Mivel a képelemző és feldolgozó algoritmusok igencsak

5. FEJEZET: AZ ALKALMAZÁS FELÉPÍTÉSE

erőforrásigényesek, nem bízhattuk ezen feladatokat az okostelefonokra. Éppen ezért szükségessé vált ezen számításokat egy szerverre bízni. Ezt a szerveralkalmazást valamelyen módon el kellett érnünk a kliens oldalról is. Választásunk a ReST alapú kommunikációra esett, ennek érdekében a kliens HTTP kéréseket küld a szervernek, melyekre szinten HTTP válaszokat vár. A kommunikációra kliens oldalon a Retrofit-ot használtuk, mely egy kliens Android és Java számára. Ez nagyban megkönnyíti a JSON formátumú üzenetek köldését és fogadását ReST alapú web service-eken keresztül. A Retrofit-on belül OkHttp könyvtár van használva a HTTP kérések küldésére.

Mivel az algoritmusunk Java programozási nyelvben íródott a szerver alkalmazás megírására is ezt a nyelvet válsztottuk. A web service-ek megalkotására a JAX-RS specifikációt belül a Jersey implementációt alkalmaztuk. A JAX-RS annotációkat használ a Java osztályok ReST függőségeinek megadásához. A Jersey implementáció pedig egy könyvtárat biztosít a Restful web service-ek implementálására egy Java servlet konténerben. Alkalmazásunk esetében az Apache Tomcat servlet konténerre esett a választás.

5.3. Az Android kliens bemutatása

A kliens alkalmazásunk három activity-ből áll: MainActivity, SelectedImageActivity és GalleryActivity. E három oldal funkcionalitásaiiból épül fel a képelemző applikációnk.

Az alkalmazás beindításakor a MainActivity jelenik meg legelőször. Itt a felhasználó eldöntheti, mit is szeretne kezdeni az applikációval. Három lehetőséget kínál az activity: új fénykép készítését, meglevő fénykép kiválasztását a galériából, illetve a már analizált és esetlegesen optimalizált képek galéria-szerű megjelenítését. Ezek az opciók egymás alatt találhatóak és LinearLayout alkalmazásával lettek elhelyezve. Egy ilyen választási lehetőség megjelölés szempontjából 2 részből áll: baloldalt egy, az esemény számára reprezentatív kép, jobboldalt pedig egy leírás. Ez a két komponens szintén LinearLayout-al lett elhelyezve. Ezeket a képeket igyekeztünk úgy megválasztani, hogy a felhasználó számára a leírás elolvasása nélkül s egyértelművé váljon, mi is fog történni, ha az adott opciót választja. Ha az elsőre, azaz a fénykép készítésére esett a választás, akkor betöltődik az alapértelmezett kamera applikáció, ahol a kép készítése után ezt el tudjuk fogani, vagy újabb képet tudunk készteni. Ha elfogadtuk a képet, az applikáció a SelectedImageActivity-hez navigál. Ha a második, azaz a kép válogatása lett kiválasztva, akkor a Gallery applikáció töltődik be, itt a telefonon levő összes kép közül tudunk válogatni, ha pedig kiválasztottunk egyet, akkor szintén a SelectedImageActivity-hez leszünk irányítva. Ezzel ellentétben, ha az utolsó opcióra esett a választás, akkor a GalleryActivity töltődik be. Mivel az activity-k közötti váltás intent-ek segítségével történik, és nekünk szükségünk van a kiválasztott vagy készített képre a SelectedImageActivity-ben is, ezért ezen képek Uri-ját extra információként meg kell adjuk a navigáló intent-ekbe. Ehhez azonban a frissen készített képet el is kell mentenünk. Az applikáció könyvtárán belül létrehoztunk ezért egy Pictures könyvtárat, melybe úgy a frissen készített, mint a kiválasztott képet elmentjük, annak érdekében, hogy majd galéria-szerűen meg tudjuk jeleníteni a későbbiekben. A képek mentése esetén egy *JPEG_timestamp* formátumú elnevezést alkalmaztunk, így biztosan nem lesz két kép ugyanazzal a névvel ellátva.

A SelectedImageActivity feladata, hogy megjelenítse a kiválasztott vagy éppen frissen készített ké-

5. FEJEZET: AZ ALKALMAZÁS FELÉPÍTÉSE

pet egy ImageView-ban. E mellett fontos szerepet játszik az activity-ben a jobb alsó sarokban található *ANALYZE* feliratú gomb, melyre kattintva a kép el lesz küldve a szerver számára, ahol megkezdőhet ennek az értékelése. E két elem, azaz az ImageView és a gomb egy CoordinatorLayout-ban van elhelyezve. Az activity ActionBar-ja nincs ellátva névvel, helyette rendelkezik egy vissza gombbal, ami valójában egy balra mutató nyíl, és erre kattintva a MainActivity-hez irányít vissza minket az applikáció. Az ActionBar jobb oldalán pedig található egy *ANALYZE* gomb, egy megosztó gomb, illetve egy beállítások gomb. Az *ANALYZE* feliratú gomb ugyanazzal a funkcionálitással van ellátva, mint a jobb alsó sarokban található gomb. Néhány felhasználó ezt a funkcionálitást is az ActionBar-ban keresné, éppen ezért került ide is be. A megosztásra kattintva egy *ACTION_SEND* típusú intent jön létre, melybe extra információként bekerül az aktuális kép uri-ja, így a képet könnyedén meg lehet osztani más applikációkkal, többek között üzenetként elküldeni más felhasználónak, megosztani a Facebook-on, vagy éppen elmenteni a to-do listánkba. Végül, az ActionBar utolsó eleme a beállításokért felelős. Erre, a fogaskerékkel jelzett opcióra kattintva beállíthatjuk a szerver IP címét. Ekkor megjelenik egy AlertDialog, ami egy címmel, rövid leírással és egy EditText-el rendelkezik. Az EditText alapértelmezetten az aktuálisan beállított IP címmel van kitöltve. Ezt egyszerűen módosíthatjuk, fontos azonban, hogy valodi IP cím formátumot adjunk meg. Ha mégis hibás a megadott formátum, akkor megjelenik egy másik AlertDialog, ami közli a felhasználóval, hogy helytelen a megadott IP cím. Itt az OK-ra kattintva, visszakerülünk az IP beállításokat tartalmazó AlertDialogba, ahol ismét megadhatjuk a kívánt címet.

A kép analizálására kattintva előbb ellenőrizzük az internetkapcsolatot, hiszen hogyha nincs internetkapcsolat, akkor teljesen feleslegesen próbálnánk meg HTTP kérést küldeni. Mivel a Post kérésben kell a szerver az egész képelemző műveletet elvégezze, mely időigényes, a válaszra igen hosszú ideig kellene várjon a kliens. Ez nagy valószínűséggel túllépné a kliens kérése által alapértelmezetten beállított várakozási időt, s ennek megfelelően TimeoutException keletkezne, s a kérés nem kapná meg a választ. Ennek elkerülése érdekében a timeout-ot átállítottuk másfél percre. Tapasztalataink alapján ez idő alatt a kérésre biztosan érkezik válasz is. Azonban így minden kérés esetén másfél percet kellene várunk, s csak azután vennénk észre, hogy valami gond történt. Ez felesleges várakozási idő lenne abban az esetben, ha például a szerver éppen leállt, vagy a felhasználó nem a szerver IP címét adta meg, s így nem tudja elérni a szerverünket a kérés. Emiatt vált szükségessé egy másik kérés létrehozása, mely nem rendelkezik meghosszabbított timeout-al. Ez egy HTTP Get, melyben egyszerűen csak egy kérést küldünk a a szervernek, s ha ettől megkapjuk a választ, akkor tudjuk, hogy az IP cím is helyes, és a szerver is fut, azaz küldhetjük a képelemző kérést a szervernek. Miközben a kliens a válaszra várakozik, az applikációban megjelenik egy ProgressDialog, mely jelzi, hogy mi történik éppen, mire kell várnai. A képoptimalizálás válasza egy JSON formátumú üzenet, mely a kivágott, maximális esztétikai értékkel rendelkező képrészlet koordinátáit, eredeti és optimalizált értékét tartalmazza. Ezen koordinátákat megkapva a kliens egy AlertDialog-ban megjeleníti a kép eredeti értékét, az optimalizált értékét, illetve a koordinátákat is. Az AlertDialog megkérdei a felhasználót, hogy szeretné-e alkalmazni a kivágást a képre. Ha pozitív válasz érkezik a felhasználótól, akkor az alkalmazás megjeleníti a kivágott képrészletet, s egyben el is menti az applikáció könyvtárába, *JPEG_timestamp_optimized* formátumú névvel.

Fontos megemlítenünk, hogy az IP cím elmentéséhez a singleton tervezési mintát alkalmaztuk. Erre

5. FEJEZET: AZ ALKALMAZÁS FELÉPÍTÉSE

azért volt szükség, mivel az applikáció futása során meg kell jegyezze minden objektumot, amelyet a program használ. Kellemetlen volna, ha az activity-k közötti navigálás során minden újra és újra be kellene állítani az IP-t. A singleton programtervezési minta egy objektumra korlátozza egy objektum példányainak számát, illetve fontos tulajdonsága, hogy ez a példány lehetőleg mindenhol elérhető legyen. Ennek megfelelően a létrehozott IPAddress példányunk is elérhető és állandó: addig ugyanazt az értéket tartalmazza, amíg azt a program egy adott pontján meg nem változtatjuk.

Amint már említettük, a kliens a Retrofit implementációt használja a HTTP kérések küldésére és fogadására. Ehhez azonban előbb létre kell hozni egy interfész, melyben definiáljuk a végpontokat. Ez az interfész Retrofit annotációkat használ a kérésekhez és paramétereikhez tartozó információk megadásához. A visszatérési érték minden Call<ResponseBody> típus. Applikációnk keretén belül ez az interfész a Service osztály által van megadva, mely értelemszerűen egy @GET és @POST annotációval ellátott függvényt tartalmaz.

5.4. A szerver bemutatása

A kliensnél bemutatott funkcionálisokból adódóan a webszerverünk csak két ReST útvonalat rendelkezik: egy Post kéréshez tartozó útvonal, melyre az elemezni kívánt képet küldjük, illetve egy Get kérésnek megfelelő útvonal, mely a szerver ébrenlétéit hivatott leellenőrizni. A Post kéréshez tartozó útvonal: <http://localhost:8084/PhotoAnalyzer/rest/file/upload> a ReST elvárásoknak megfelelően önmagában is beszédes, látható, hogy a PhotoAnalyzer szerverhez irányítjuk kérésünket, mely rest hívást vár el, és konkrétan egy file feltöltést végzünk. A <http://localhost:8084/PhotoAnalyzer/rest/alive> pedig a Get kérés útvonalra, melyből szintén látható a funkcionális, azaz a szerver ébrenlétének ellenőrzése.

A szerver ébrenlétének ellenőrzésekor egy paraméter nélküli Get kérés történik, mely egy egyszerű szöveges üzenetet ad válaszként. Ha a kliens megkapja ezt a választ, az azt jelenti, a szerver működik, tehát elköldheti a valódi kérést: a képet tartalmazó Post-ot. Ha valamilyen okból ezt a választ nem kapja meg a kliens, akkor egyértelműen valami gond történt menet közben: vagy megszakadt az internetkapcsolat, vagy rossz IP-cím lett beállítva a kliens által, vagy éppen leállt a szerver. Ilyen esetben a kliens értesíti a felhasználót a problémáról. Ennek a kérésnek a lekezelésére egy az IsAliceService nevű web service-t hoztunk létre szerver oldalon.

Az előbb említett Web Service mellett csak egyetlen létezik, az UploadFileService, mely a kép feltöltéséért, elemzéséért és a válasz visszaküldéséért felelős. Ez a service képezi a találkozási pontot a webszerver és maga az általunk kidolgozott algoritmus között. Miután a szerver sikeresen megkapta a feltöltött képet, ideiglenesen lementi egy erre létrekozott folderbe, a könnyebb hozzáférés végett. Ezután az Analyzer osztályt felhasználva megtörténik a képen található fő objektumok és vonalak detektálása, illetve az optimális képrészlet megkeresése a genetikus algoritmus segítségével. Észrevehető, hogy a teljes algoritmusnak ennek a Post kérésnek a keretén belül kell lefutnia. Ez azonban időigényes folyamat, pontosan ezért vált szükségessé megfelelő hosszúságú timeout beállítása a kliens oldalán. Mégis, hogy energiát és időt spórolhassunk, válaszként nem a kész, kivágott képet küldjük vissza, hanem csak ennek a paramétereit: a bal felső sarkának x és y koordinátáját, illetve a szélességet és hosszúságát. Végül, hogy

5. FEJEZET: AZ ALKALMAZÁS FELÉPÍTÉSE

a szerver se tartalmazzon fölösleges információkat, az előzőleg feltöltött, elmentett képeket töröljük.

6. fejezet

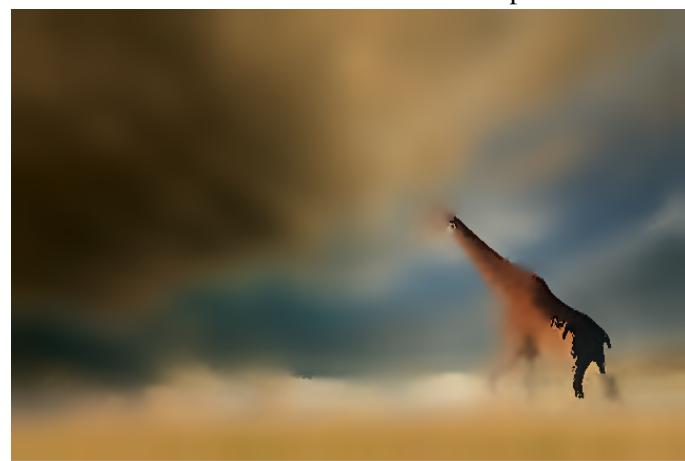
5. Esettanulmányok, észrevételek

A következőkben néhány konkrét eseten keresztül mutatjuk be az algoritmusunk működését, felhívva a figyelmet az algoritmus tulajdonságaira. Példáinkon keresztül bemutatjuk az objektumok azonosításának lépései, a szabályokat képviselő függvények értékeit, illetve a bemeneti és eredményül kapott képet is. Az első esettanulmány lépései:

6.1. ábra. A bemeneti kép



6.2. ábra. Az összesített kép



6. FEJEZET: 5. ESETTANULMÁNYOK, ÉSZREVÉTELEK

6.3. ábra. A kép threshold alkalmazás után



Az ábra az algoritmus különböző fázisaiban mutatja be a bemeneti a képet. A 6.2 ábra a kezdeti összemosás utáni állapotot örökíti meg, látszik, hogy a felesleges élek teljesen eltűntek, csak a zsíráf mentén maradtak meg láthatóak. A 6.3 ábra a fennmaradt éleket mutatja, ez alapján lettek azonosítva a 6.4. ábrán bekeretezett objektumok. A következő, 6.5 ábra az összemosás nélkül alkalmazott thresholdolást mutatja, itt látható, hogy sokkal több él megmaradt, amire szükségünk is van a vonalak azonosításához. Végül az utolsó, 6.6 ábrán a megtalált vonalak vannak feltüntetve. Bár az előző, 6.5 kép azt a látszatot kelti, hogy sokkal több vonal fedezhető fel, mégsemcsok vonal azonosítva. Ez annak köszönhető, hogy a többi vonaljelölt vagy túlságosan rövidnek bizonyult, vagy túl nagy szakadások darabolták fel őket.

6.4. ábra. A kép a detektált célobjektumokkal



6. FEJEZET: 5. ESETTANULMÁNYOK, ÉSZREVÉTELEK

6.5. ábra. Simán thresholdolt kép



6.6. ábra. A detektált vonalak



6.7. ábra. eredményül kapott, optimalizált kép



A kezdeti kép esztétikai értéke 0.666046 volt, ami azt jelenti, hogy a kép megközelítőleg 66% - ban tartotta be a kompozíciós szabályokat. Egész pontosan a harmadolás szabály értéke 0.770615, a vizuális egyensúlyé 0.421820, illetve az átlódominanciáé 0.947065 volt.

6. FEJEZET: 5. ESETTANULMÁNYOK, ÉSZREVÉTELEK

Ez a kapott érték nem nevezhető jónak. Szükséges tehát az optimalizációs algoritmussal megkeresni azon képrészletet, mely maximális eredményt szolgáltat. Az algoritmus eredményeképpen egy olyan képrészletet kaptunk, melynek bal felső sarkát a (191, 123) pont képezi illetve szélessége 405, míg hosszúsága 270 pixel. Ennek a képnak az esztétikai értéke jelentősen jobb, 0.954201, azaz megközelítőleg 95% - os. Észrevehető, hogy minden szabály értéke nőtt: a harmadolás szabály értéke 0.956310, a vizuális egyensúlyé 0.955622, illetve az átlódominanciáé 0.957842 lett. Végül a 6.7. ábra mutatja a kapott eredményt.

6.8. ábra. Túlzsúfolt képekre alkalmazva az algoritmust.

forrasokatbetenni!!!!!!!!!!!!!!
!!!!!!!!!!!!!!



Az algoritmus többszöri végrehajtása során észrevettük, hogy a túlzsúfolt, vagy túl közelí képek esetén a kép területének nagyon nagy százaléka a célobjektumot képezi. Ez annak köszönhető, hogy az algoritmus nagyon sok, összefüggő élet talál. Az ilyen képek nem lehetnek igazán esztétikusak. Nem előnyös ilyen túlzsúfolt képekre alkalmazni az algoritmust. A közeli képeket is érdemes elkerülni, ugyanis az algoritmus részlet kivágásával, azaz beközelítéssel próbálja feljavítani a képet, egy nagyon közeli kép esetén azonban már ezen nem lehet ilyen módszerrel optimalizálni.

6.9. ábra. Arcok detektálása



Az algoritmus fontos lépései, melyet a fent bemutatott esettanulmány keretein belül nem tudtunk demonstrálni: az arcdetektálás, illetve az átlóközeli vonalak azonosítása. Ahogyan a 6.9 ábra is mutatja,

6. FEJEZET: 5. ESETTANULMÁNYOK, ÉSZREVÉTELEK

az arcok nyilvántartása az objektumokéhoz teljesen hasonló módon történik: egy négyzetet rajzolhatunk az arc köré, ezen négyzeteket eltárolva kapjuk meg az arcok listáját. Az átlós vonalak azonosítása pedig a vonalak azonosításával egy időben történik, annyi különbséggel, hogy ezen vonalakat külön listában tároljuk el, ahogyan ezt a 6.10 ábra is mutatja.

6.10. ábra. Az eredeti kép, illetve az átlagos és átlóközeli vonalak, külön színnel jelölve.

forrasokatbetenni!!!!!!!!!!!!!!
!!!!!!



Egy másik fontos észrevétel, hogy a jó felépítésű, 0.8 feletti esztétikai értékkel rendelkező képek esetén az algoritmus hajlamos ezt az értéket optimálisan elfogadni. Nem minden esetben találja meg az ennél optimálisabb értékeket. Látványos javítást 0.6-0.7 - nél kisebb esztétikai értékkel rendelkező képek esetén érhetünk el.

Az algoritmus megírásában a legfőbb kihívást a megfelelő képfeldolgozási paraméterek megtalálása jelentette. Jelen paraméterek mellett is előfordulnak olyan képek, melyeken az algoritmus nem talál meg minden kiemelkedő területet. Ez természetesen befolyásolja az esztétikai érték kiszámolását is. Azonban nem lehet általánosan mindenre jól működő paramétereket találni. minden kép egyedi a maga módján, így ami az egyik kép esetében nagyon jó paraméterhalmaz, az egy másik kép esetén nagyon rossznak bizonyulhat. Célunk tehát egy általánosan elég jól működő paraméteregyüttes megtalálása volt.

6. FEJEZET: 5. ESETTANULMÁNYOK, ÉSZREVÉTELEK

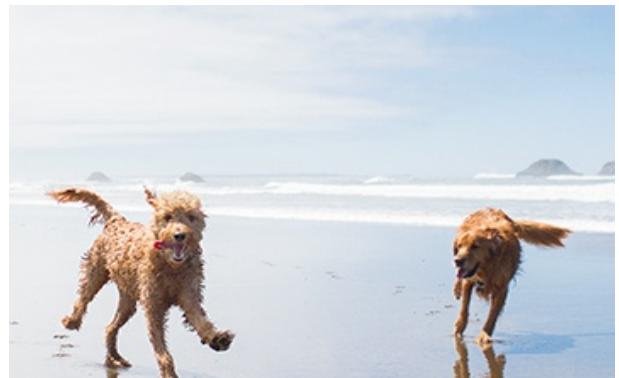
6.11. ábra. Az eredeti kép esztétikai értéke: 0.899226 , az optimalizált kép esztétikai értéke: 0.94805786

forrasokatbetenni!!!!!!!!!!!!!!



6.12. ábra. Az eredeti kép esztétikai értéke: 0.511398, az optimalizált kép esztétikai értéke: 0.861275 .

forrasokatbetenni!!!!!!!!!!!!!!



6.13. ábra. Az eredeti kép esztétikai értéke: 0.919799 , az optimalizált kép esztétikai értéke: 0.930881 .

forrasokatbetenni!!!!!!!!!!!!!!



6. FEJEZET: 5. ESETTANULMÁNYOK, ÉSZREVÉTELEK

6.14. ábra. Az eredeti kép esztétikai értéke: 0.899165, az optimalizált kép esztétikai értéke: 0.921564 .
[forrasokatbetenni!!!!!!](#)



A fenti ábrán a baloldali képek az eredetiek, míg a jobboldali képek az általunk ajánlott optimalizált változataik. Mindegyik kép alatt fel van tüntetve az általunk kiszámított esztétikai érték is. Látható, hogy egyes képek esetén jelentős optimalizációt sikerült elérnünk.

Az általunk használt algoritmus csak a szabályok betartását ellenőrzi, nincs beépítve semmilyen színtű kreativitás. Ezért mindenki eldöntheti, hogy a kapott képek valóban esztétikusabbak-e. Az azonban biztos, hogy a vizsgált szabályokat jobban betartják.

7. fejezet

Következtetések, továbbfejlesztési lehetőségek

Dolgozatunk jelen állapotában egy olyan algoritmust ír le, mely képes egy bemeneti képen azonosítani a kiemelkedő területeket és vonalakat, ezek felhasználásával esztétikai értéket számolni, illetve az adott képnek egy olyan részképet eredményül adni, mely optimális értékkel rendelkezik. Segítséget nyújt tehát az amatőr fényképészeknek a sikeresebb, esztétikusabb képek készítésében. Az eredményesebb felhasználás érdekében érdemes néhány továbbfejlesztési lehetőséget figyelembe venni.

Annak ellenére, hogy az algoritmus jelen állapotában sem nevezhető lassúnak, érdemes volna a számítások gyorsítására törekedni. A dolgozat elsődleges továbbfejlesztési lehetőségei közé sorolható tehát a performancia javítása. Mivel az algoritmus elég időigényes képfeldolgozási számításokat igényel, illetve az esztétikai értéket is több ezerszer számítjuk ki a genetikus algoritmusban, ezeket jó volna a lehető legrövidebb idő alatt elvégezni. Az objektumok, illetve vonalak azonosítását külön szálon futtatva, illetve az esztétikai értékszámolásban is párhuzamosítást alkalmazva csökkenthető volna az algoritmus futási ideje.

Szintén az algoritmus gyorsítását szolgálná a genetikus algoritmus továbbfejlesztése. Érdemes volna tehát különböző típusú genetikus algoritmusokat kipróbálni, közben figyelni arra, hogy milyen típusú genetikus operátorok esetén kapunk gyorsabb, vagy adott esetben jobb eredményt.

Egy másik továbbfejlesztési lehetőséggént megemlíthető egyéb kompozíciós szabályok bevezetése. Minél több szabályt sikerül beépíteni, annál pontosabban lehetne esztétikai értéket számolni. Ehhez azonban olyan szabályokat kell találnunk, melyek jól együttműködnek a már felhasználtakkal. A megfelelő szabályok megtalálása kihívást jelenthet. Érdemes volna tehát olyan szabályok és mérési metrikák után nézni, melyek növelnék a kép esztétikai megítélésének helyességét.

Irodalomjegyzék

Canny edge detection.

Opencv hivatalos weboldal.

Image thresholding.

Canny, J. *A computational approach to edge detection. IEEE Transactions on pattern analysis and machine intelligence.* Springer, 1986.

Hoenig, F. Defining computational aesthetics. comp. aesthetics in graphics, vis. and imaging, 2005. URL <https://pdfs.semanticscholar.org/53ff/fdf60dc8d2a70d8e731c21a5ddb042b959ee.pdf>.

István, D. Mélységélesség - képrajzolás és gyújtótávolság, a perspektívahatás alakítása. URL <http://www.fotovilag.hu/foto/objek-1/melysg.htm>.

Ligang Liu, e. a. Optimizing photo composition. *Computer Graphics Forum*, 29, 2010.

Michalewicz, Z. *Genetic algorithms + data structures = evolution programs.* Springer, 1992.

O'Carroll, B. Composition techniques that will improve your photos. URL <https://petapixel.com/2016/09/14/20-composition-techniques-will-improve-photos/>.

Peterson, D. Composition rules for photos that shine. URL <http://www.digital-photo-secrets.com/tip/3372/18-composition-rules-for-photos-that-shine/>.

Rowse, D. Elements of composition in photography. URL <https://digital-photography-school.com/5-elements-of-composition-in-photography/>.

Wei Luo, X. T., Xiaogang Wang. Content-based photo quality assessment. *Conference on Computer Vision (ICCV)*, 2011.