

SAPIENTIA – ERDÉLYI MAGYAR TUDOMÁNYEGYETEM
MAROSVÁSÁRHELYI MŰSZAKI és HUMÁNTUDOMÁNYOK KAR
VILLAMOSMÉRNÖKI TANSZÉK

Számítástechnika IV év

Kép szegmentáció B-Spline alapú szintfüggvényekkel



VEZETŐ TANÁR:

DR. SZILÁGYI LÁSZLÓ - DOCENS

SZERZŐ:

GÁBOR BERNÁT

2011

TARTALOMJEGYZÉK

TARTALOMJEGYZÉK	2
KIVONAT	3
1. ÁLTALÁNOS BEVEZETÉS	4
2. ELMÉLETI MEGALAPOZÁS	6
2.1. SZINTFÜGGVÉNY.....	6
2.2. ENERGIA KRITÉRIUM.....	6
2.3. B-SPLINE SZINTFÜGGVÉNY MODELL.....	7
2.4. ENERGIA KRITÉRIUM MINIMALIZÁLÁSA.....	8
2.5. DISZKRÉT FORMA	9
2.5.1. Diszkrét jellemző függvény	9
2.5.2. Gradiens számítás	10
2.6. IMPLEMENTÁCIÓS PROBLÉMÁK	10
2.6.1. Korlátos szintbeállító függvény	10
2.6.2. Első fokú gradiens módszer	12
3. MEGVALÓSÍTÁS	13
3.1. HASZNÁLT FÜGGVÉNYEK ÉS PARAMÉTEREK	13
3.2. DIAGRAMOK	15
3.2.1. Osztálydiagram	15
3.2.2. Adatfolyam.....	16
4. SZEGMENTÁCIÓS EREDMÉNYEK	17
4.1. SZIMULÁCIÓS KÉPEK	17
4.2. VALÓ VILÁGI KÉPEK.....	17
4.3. ORVOSI KÉPEK.....	18
5. KÖVETKEZTETÉS	19
IRODALOMJEGYZÉK	21



KIVONAT

1965-ben Moore egy tudományos írásában megfogalmazta a ma Moore törvényeként ismert kijelentést, mely szerint egy áramkörbe beintegrált alkotó elemek száma duplázódik évente. Az elektronika e látványos fejlődési ritmusát talán csak a felhasználási módszereknek és felhasználók számának sikerül túlszárnyalni. Nem meglepetés tehát, hogy a digitális kamerák és egyéb képalkotó rendszerek rohamos elterjedésével megnőtt az az igény is, hogy a begyűjtött képekről könnyen és gyorsan nyerjünk ki különböző ismereteket.

A képfeldolgozás azon területét mely formák, alakzatok megtalálásával és kinyerésével foglalkozik, szegmentációnak hívjuk. A szakirodalomban számos olyan módszer van leírva, mellyel egy képből ki lehet nyerni egy tárgy körvonalát. Jelen dolgozatba ezek közül egy B-Spline függvénnyel interpolált szintbeállító megközelítést mutatok be.

Ennek egyik fő tulajdonsága, hogy a szükséges transzformációkat és deriválásokat szűrők segítségével valósítjuk meg. Emiatt lehetséges gyors, ún. „batch” típusú utasításokkal való megvalósítás létrehozása. Implementálásra a C++ nyelvezetet használtam az OpenCV könyvtár segítségével. Az algoritmus lehetséges alkalmazási területei közül megemlíteném egy kéz, fej követését, avagy az orvosi képfeldolgozásban alkalmazható úgy mikroszkóp felvételen sejtek hely azonosítására, mint MRI és echókardiográf képeken szegmentációra.

Kulcsszavak: szegmentáció, szintbeállító függvény, B-Spline, energia minimalizálása, C++



1. ÁLTALÁNOS BEVEZETÉS

Századunk gyors elektronikai fejlődésével számos iparágban, szakterületen egyre gyakrabban bukik fel a probléma, hogy egy képi anyagon levő tárgyakat, formákat, alakzatokat azonosítani kell. Máskor ezek változását kell, hogy nyomon kövessük, hogy majd a gyűjtött adatok szerint valamilyen beavatkozást végezzünk el. Ez lehet egy konkrét elektronikai gép vezérlése egy ipari gép esetében, avagy egy emberi döntés az orvos részéről, ha echókardiográf képekről beszélünk.

A klasszikus szakirodalomban számos módszer létezik a képek homogén részeire való bontásra. Ilyen egy vágás alkalmazása intenzitás képeknél, Bezdek által leírt fuzzy osztályozás [1], Canny él detektálás [2], avagy az általános Hough transzformáció [3], hogy csak egy párat említsek. Az érdekelt olvasó ezek alapos áttanulmányozását megfigyelheti Sonka-Hlavac-Boyle könyvében [4].

Képeink habár számunkra diszkrét formában állnak rendelkezésre egy folytonos világból származnak. Hogy ezeket diszkrét formába tárolni tudjuk mintavételezésre és kvantifikációra szorulunk. A klasszikus módszerek egyik fő gondjuk, hogy e diszkrét modelleken alapulnak és így érzékenyek paramétereik jó megválasztására. Ez gondot akkor okozhat mikor komplex változó alakzatok és zajos képeken akarjuk őket alkalmazni.

Nemrég új kép szegmentációs modelleket írtak le melyek a variációs megközelítésen alapulnak. Ezek folytonos téren vannak definiálva és így matematikailag jól kitanulmányozottak mivel a folytonos analízis sokkal jobban ki van dolgozva, mint diszkrét párja. A két legismertebb modell: a Mumford-Shah [5] [6] és az aktív kontúr [7]. A Mumford-Shah a képet megkülönböztethető homogén régiókra bontja, de úgy, hogy egyformán fontosnak tartja azokat. Az aktív kontúr ezzel szemben egyes régiókat fontosabbnak nyilvánít, mint a többit és ezért az inicializálási ponthoz a legközelebbi kontúrt keresi meg.

Mindkettő esetében a modell egy energiafüggvény minimalizálásra vezet vissza a feladatott. Ez eredményezi ennek variációs jelegét. A Mumford-Shah modellre Chen-Veese [8] írt fel először egy gyakorlatba megoldható energiafüggvényt (funkcionálist), míg az aktív kontúrra Kass-Witkin-Terzopoulos [7]. E két modellből számos továbbfejlesztés alakult meg



az elmúlt évek során. Ezeknek egy alapos áttekintését megkaphatjuk Brensson 2005-ös doktori tézisében [9].

A hagyományos irodalomban a felírt energiafüggvények minimalizálása egy parciális differenciál egyenletre vezetődik vissza, amelyet véges-differenciál formákkal oldanak meg. Ennek problémái, hogy nehezen kezeli az esetleges létrejövő szakadási pontokat. Ennek elkerülésére 2009-ben Bernard [10] ajánlott egy B-Spline alapú megközelítést ahol e probléma a keret implicit tulajdonságai miatt már nem merül fel.

A dolgozat továbbá bemutatja e módszer felépítését, előnyeit és nem utolsósorban implementálási kihívásait. Ezt követően megfigyelhetjük a leprogramozott algoritmus tervét, kihívásait és annak konkrét forrás kódja is elkészül a C++ nyelvezetben.

Hogy a kép kezelését (beolvasás, kiírás, képpontok elérése és kép transzformációk) leegyszerűsítsük és elválasszuk a mi problémánktól, felhasználjuk az OpenCV könyvtárat. A dolgozat befejező részében különböző képen teszteljük az algoritmus helyes működését és vonnunk le következtetéseket ennek megfelelően.

2. ELMÉLETI MEGALAPOZÁS

A kép szegmentációban a szintbeállító modell alatt értjük azon deformálódó modelleket melyek a cél alakzatot úgy kapják meg, hogy egy határfelületet terjesztenek végig egy sima függvény nulla szintjén. Az ilyen függvényeket nevezzük szintbeállító típusúnak. Azt, hogy a határfelület terjeszkedjen, egy variációs megfogalmazáson keresztül érjük el. Azaz a szegmentáció problémáját kifejezzük, mint egy energiafüggvény minimalizálása mely magába foglalja a keresett forma tulajdonságait.

E lépést továbbá át lehet írni, mint egy időfüggő parciális differenciálegyenlet. Egy ilyenek a megoldása történhet véges-differenciál módszerrel [11], avagy ahogy ebben az iratban bemutatjuk, használunk, egy folytonos megközelítést. Itt a szintfüggvényt ki tudjuk fejezni, mint egy folytonos parametrikus függvény melyen a differenciálegyenlet már könnyen megoldható. A felhasznált folytonos függvények melyekből építkezni fogunk a B-Spline-ok.

2.1. Szintfüggvény

Legyen Ω nyílt határos része az \mathbb{R}^d - nek és $f: \Omega \rightarrow \mathbb{R}$ egy d dimenziójú kép. A szintbeállító megfogalmazásban a terjeszkedő $\Gamma \subset \mathbb{R}^d$ határfüggvény nem más, mint a nulla szint Lipschitz-folytonos függvénye ϕ - nak mely $d+1$ dimenziójú és teljesíti az alábbi megkötéseket (ahol Ω_{in} egy régió mely határolja a határfelületet ($\Gamma = \partial\Omega_{in}$) és Ω_{out} meg a teljes tér többi része, azaz $\Omega \setminus \Omega_{in}$):

$$\left\{ \begin{array}{ll} \phi(x) > 0, & \forall x \in \Omega_{in} \\ \phi(x) < 0, & \forall x \in \Omega_{out} \\ \phi(x) = 0, & \forall x \in \Gamma \end{array} \right. \quad (1.)$$

2.2. Energia kritérium

A határfelületet egy kezdeti helyzetből az energia kritérium függvény fogja azt a kívánt alakzatra vezérelni úgy, hogy közben ennek értéke a lehető legkisebb legyen. E függvénynek az általános alakját a következő alakba írhatjuk fel:



$$\begin{aligned}
J(\phi) = & v_{in} \int_{\Omega} g_{in}(x, \phi(x)) H(\phi(x)) dx_1 \dots dx_d \\
& + v_{out} \int_{\Omega} g_{out}(x, \phi(x)) (1 - H(\phi(x))) dx_1 \dots dx_d \\
& + v_c \int_{\Omega} g_c(x, \phi(x)) \delta(\phi(x)) \|\nabla \phi(x)\| dx_1 \dots dx_d \quad (2.)
\end{aligned}$$

Itt az első két elem a régió tagok mely a határfelületet határoló külső és belső régióhoz rendelt energia érték. Az utolsó meg a határfelülethez (Γ) van kapcsolva és a szakirodalomba, mint kontúr tag van említve. Ennek megfelelően g_{in} és g_{out} leírja az objektumot és a háttérrel, míg végül a g_c magát a keresett kontúrt. Továbbá a H a Heaviside (más néven egységugrás) és δ a Dirac egyváltozós függvényt jelöli. v_{in} , v_{out} és v_c pozitív paramétervektorok és egy-egy tag fontosságát lehet velük állítani.

2.3. B-Spline szintfüggvény modell

A Spline függvények nagyon jó interpolációs függvények az általános interpolációs keretben, ahol nem a konkrét interpolálni értékekkel, hanem interpolációs együtthatókon keresztül interpolálunk egy pontot. Ezen együtthatókat valamilyen, egyelőre nem érdekes, módszerrel kapjuk meg a bemeneti értékeinkből. E bemeneti értékek megkötésekkel egyenértékűek, hiszen elvárjuk, hogy a forma mellyel interpoláljuk, ezeken átmenjen.

A B-Spline a legegyszerűbb ilyen függvények (polinomok) és kifejtve *Bázis-Spline*-nak felel meg. Tehát a B-Spline alapokba egy n -ed fokú polinom. Az interpoláló függvény ilyen B-Spline – ok lineáris kombinációja. A lineáris kombináció együtthatói megegyeznek az becslő B-Spline függvény együtthatóival. Egy B-Spline csak a hozzá legközelebb eső $n+1$ csomópontra építkezik. Ugyanakkor egy pont konkrét interpolációs értékét a legközelebbi $n+1$ B-Spline lineáris kombinációja adja meg.

A B-Spline alkalmazási lehetőségeit és módszereit (főleg hatékony szűrők formájában) Michael Unser dolgozta ki [12] [13]. Fejezzük ki $\phi(x)$ – et mint B-Spline bázis függvények lineáris kombinációja [14]:

$$\phi(x) = \sum_{k \in \mathbb{Z}^d} c[k] \beta^n \left(\frac{x}{h} - k \right). \quad (3.)$$



Ahol $\beta^n(\cdot)$ az egységes és szimmetrikus d -dimenziójú n -ed fokú B-Spline. E függvény habár paraméterként egy vektort kap könnyen kezelhető, hiszen szétválasztható és ezért felírható, mint n darab egy dimenziós B-Spline szorzat [14]:

$$\beta^n(x) = \prod_{j=1}^d \beta^n(x_j) \quad (4.)$$

Ez gyakorlatba azt jelenti, hogy egy 2D-os B-Spline esetében például a ϕ kiszámítása nem más, mint alkalmazzuk az egy dimenziós számítási módszereket úgy a sorokra, mint az oszlopokra egymás után. A B-Spline függvények csomópontjai, ami köré felírjuk őket, legyenek egy háló eloszlásban a Ω felületen, és két pont közti a távolság legyen h . A B-Spline együtthatókat a $c[k]$ tárolja.

2.4. Energia kritérium minimalizálása

A hagyományos variációs keretben ezt az energia kritériumot a szintbeállító függvényre nézve Euler-Lagrange egyenletek [8] avagy Fréchet-Gateux [15] deriváltak felhasználásával végezzük el. Ezzel ellentétben mi a deriválást is a B-Spline keretben végezzük el, ahol kifejezzük a minimalizálást is, mint a B-Spline együtthatók optimalizálása úgy, hogy az energia kritérium a lehető legkisebb értéket vegye fel. Bernard [10] bebizonyította, hogy ez nem más, mint (2.) –es egyenlet deriválása a $c[k_0]$ együtthatókra nézve:

$$\frac{\partial J}{\partial c[k_0]} = \int_{\Omega} \omega(x) \beta^n\left(\frac{x}{h} - k\right) dx_1 \dots dx_d \quad (5.)$$

Ahol ω felírható a következő formában:

$$\begin{aligned} \omega(x) = & v_{in} \left(\frac{\partial g_{in}(x, \phi(x))}{\partial \phi(x)} H(\phi(x)) + g_{in}(x, \phi(x)) \delta(\phi(x)) \right) + \\ & v_{out} \left(\frac{\partial g_{out}(x, \phi(x))}{\partial \phi(x)} (1 - H(\phi(x))) - g_{out}(x, \phi(x)) \delta(\phi(x)) \right) + \\ & v_c \left(\frac{\partial g_c(x, \phi(x))}{\partial \phi(x)} \|\nabla \phi(x)\| - \text{div} \left(\frac{g_c(x, \phi(x)) \nabla \phi(x)}{\|\nabla \phi(x)\|} \right) \right) \delta(\phi(x)). \end{aligned} \quad (6.)$$

Mivel $\omega(x)$ megadja a szegmentált objektum fő jellemzőit a szakirodalomban *jellemző függvénynek* nevezzük. Ahogy ezt Bernard is leírja az energia kritérium minimalizálása a B-Spline együtthatókra nézve általánosan nem eredményez egy zárt alakú megoldást (azaz ez nem írható fel, mint véges és ismert függvények kombinációja).



Ezért a lokális minimum meghatározása érdekében az első fokú gradiens módszert alkalmazzuk mely a következő alakhoz vezet:

$$c^{(i+1)} = c^{(i)} - \lambda \nabla_c J(c^{(i)}) \quad (7.)$$

Itt λ a gradiens módszer lépése és ∇_c az energiafüggvény gradiense a B-Spline együtthatókra nézve, ahogy ezt az (5.) és (6.) egyenletek leírják. Az (5.)-ös egyenlet egy érdekes értelmezéshez vezet amennyiben az alábbi formában definiáljuk egy n-ed fokú h-val felskálázott B-Spline-ot:

$$\beta_h^n(x) = \beta^n\left(\frac{x}{h}\right) \quad (8.)$$

Így az energia gradiens felírható az alábbi alakban:

$$\nabla_c J = \frac{\partial J}{\partial c[k]} = \int_{\Omega} \omega(x) \beta_h^n(x - hk) dx_1 \dots dx_d \quad (9.)$$

Ez meg már igen csak hasonlít és meg is egyezik a jellemző függvény és a $\beta_h^n(x)$ konvolúciójával melyet h periódussal mintavételezünk.

2.5. Diszkrét forma

Képeinket a folytonos világból vesszük azonban tár kapacitásunk végleges ezért azt diszkrét formába alakítsuk valamilyen mintavételezési séma alapján. Kép szegmentációkor e forma áll rendelkezésünkre, ezért a fenti folytonos egyenleteket a gradiens kiszámolására diszkrét formára kell, hogy hozzuk. E feladat megegyezik a jellemző függvény és a B-Spline bázis diszkrét alakjuknak felírásával.

2.5.1. Diszkrét jellemző függvény

Ennek megvalósítása a (6.) – ós egyenlet diszkrét alakra való hozását jelenti és megegyezik a Heaviside és Dirac függvény megközelítésével. E problémával már több irat is foglalkozott részletesen. Mi most a Chan-Vese [8] javasolt formát fogjuk használni azáltal, hogy H-nak a Heaviside függvény bármely megközelítését és C^1 szabályosítását vesszük (H_ϵ). A Delta függvény (δ_ϵ) meg ennek a deriváltja (H'_ϵ) lesz, amikor $\epsilon \rightarrow 0$.

E két alakot behelyettesítjük a (2.) –es egyenletbe, hogy megkapjuk az energia kritérium szabályosított formáját. Hasonlóan cselekedve a (6.) –ós egyenlet esetében is



megkapjuk az ezzel azonosítót jellemző függvényt. Jelöljük az így kapott egyenletet $\omega_\epsilon[u]$ – val ahol $u \in \mathbb{Z}^d$.

2.5.2. Gradiens számítás

A gradiens diszkrét formájának felírásának érdekében a korábban meghatározott $\omega_\epsilon[u]$ jellemző függvény mellett a B-Spline-ot is átírjuk diszkrét alakra, ahogy azt Unser megfogalmazta [12]. Jelölje a szimmetrikus d dimenziós n -ed fokú diszkrét B-Spline –t a $b^n_h[u]$. Ezt megkapjuk, ha a mintavételezzük a folytonos $\beta^n(x)$ – et az egész pontokban. A folytonosnál érvényes szétválasztathatóság tulajdonság itt is érvényes marad. Ha egész értékű távolságot használunk a B-Spline csomópontoknak ($h \in \mathbb{N} \setminus \{0\}$) az n – ed fokú h – val skálázott diszkrét B-Spline felírható a következő alakban:

$$b^n_h[u] = \beta^n\left(\frac{u}{h}\right) \quad (10.)$$

Ezt a (9.) – es egyenletbe behelyezve kapjuk a keresett diszkrét alakot:

$$\langle \nabla_c J \rangle [k] = \left\langle \frac{\partial J}{\partial c[k]} \right\rangle = \sum_{u \in \mathbb{Z}^d} \omega_\epsilon[u] b^n_h[u - hk]. \quad (11.)$$

Tehát az energia gradiense megfelel a jellemző függvény konvolúciós szorzatának a B-Spline – al és alul mintavételezve egy h faktorrall:

$$\langle \nabla_c J \rangle [k] = (\omega_\epsilon * b^n_h)_{\downarrow h}[k] \quad (12.)$$

E egyenlet egy hatékony eszközt ad mellyel könnyedén és gyorsan kiszámolható a gradiens és ez által a szintbeállító függvény fejlődése a (7.) – es egyenlet segítségével.

2.6. Implementációs problémák

2.6.1. Korlátos szintbeállító függvény

A határfelület terjedése során a szintbeállító függvénybe létrejöhetnek nagyon meredek vagy lapos régiók. Mindkét esetben a gradiens ilyenkor félrevezetheti a szintfüggvényünket mivel úgy a nagyon nagy értékek, avagy a nullához közeli numerikus instabilitást vezetnek be módszerünkbe. A klasszikus szintbeállító keretben ennek elkerülésére időnként módosítják a kialakuló szintfüggvényt úgy, hogy az ismét egy távolság függvényt írjon le a nulla szinthez képest.



E megoldásnak azonban az egyértelmű algoritmus őszzámítási költség megnövekedése mellett egy topológia megkötést is eredményez. Ez úgy nyilvánul meg, hogy a kezdeti határfelülettől távol nem engedi meg, hogy újabb nulla szintek jöjjenek létre. Tehát csak azon kontúrokat kapjuk meg, amelyek már a kezdeti kontúr is tartalmazott valamilyen szinten.

E probléma egy elegáns megoldása a szint-beállító függvény korlátozását jelenti. Ezt a módszert Bernard már sikeresen használta a radiális bázis függvénnyel felírt szintbeállító függvény esetében [16]. Mivel a B-Spline alakja egy lineáris kifejezés ez is könnyen felírható, mint a B-Spline együtthatók normalizálása. Bernard [10] bebizonyította, hogy az így nyert korlátos szint-beállító függvény megakadályozza a meredek és lapos régiók kialakulását és ez által szükségtelené válik az időnkénti újra inicializálási lépés a klasszikus numerikus megoldási módszerekkel szemben.

Hogy gyakorlatban is könnyen alkalmazható legyen, a normalizáláshoz a végtelen normát használjuk. Ezt maximum normaként is ismeretes mivel általános formája:

$$\|c\|_{\infty} = \max(|c_1|, |c_2|, \dots, |c_n|). \quad (13.)$$

Egy implicit függvényt amennyiben beszorzunk egy nem nulla együtthatóval, az nem változtatja meg annak felületét. Mivel ϕ a B-Spline együtthatókon keresztül van ábrázolva ϕ szorzása ε – al megfelel a c/k együtthatók szorzásával ugyancsak ε – al. Bernard a B függelékébe [10] bebizonyította, hogy a B-Spline együtthatók végtelen normája ($\|c\|_{\infty}$) korlátossá teszi a szintbeállító függvényt:

$$|\phi(x)| < \|c\|_{\infty} \quad (14.)$$

Tehát a szintbeállító függvény újrnormalizálási lépése (a $[-1, 1]$ intervallumba) nem más, mint, hogy egy-egy első fokú gradiens módszer lépése után elvégezzük a következő műveletet:

$$c^{(i+1)} = \frac{c^{(i+1)}}{\|c^{(i+1)}\|_{\infty}} \quad (15.)$$

Szükséges megemlíteni, hogy ez egyúttal határokat szab a szintbeállító függvény gradiens normájára is. Ennek mértékére és annak bizonyítására az érdeklő olvasó Bernard cikkét kutathatja. [10]



2.6.2. Első fokú gradiens módszer

Az energia függvény minimalizálására egy első fokú gradiens módszert alkalmazunk melynél visszacsatolt lépésváltoztatást használunk. Minden iterációban alkalmazzuk először a (7.) – es, majd a (15.) – ös egyenletet, hogy a jelenlegi ismert $c^{(i+1)}$ – ből kiszámoljunk egy lehetséges $c^{(i+1)}$ – et és ezen együtthatók által eredményezett energia értéket. Ha a korábbi energia értékhez képest kisebb értéket kapunk, akkor a gradiens módszer e lépése sikeres volt és elfogadjuk a kiszámolt értékeket.

A továbbiakban ezzel számolunk, és a lépés méretét egy $\alpha_f \geq 1$ értékkel *beszorozzuk*. Ellenkező esetben eldobjuk a kapott értéket és a lépést egy $\alpha_f' \geq 1$ értékkel *osszuk*. Ezután megismételjük az iterációt addig, amíg egy maximális lépésszám határt el nem érünk vagy amíg $c^{(i)}$ egy kívánt küszöbérték alatti mértékben nem változik több egymás utáni iteráció során. E több lépésre azért van szükség, hogy megbizonyosodjunk, hogy a megoldásunk stabil és a felület már biztos nem fog alakulni tovább.



3. MEGVALÓSÍTÁS

3.1. Használt függvények és paraméterek

Az algoritmus implementálásához most definiálnunk kell a függvényeket, melyek leírják a keresett alakzat belső, külső és esetleg határfelületeit. Legyen ez a Chan-Vese féle funkcionális mely részenként konstans intenzitású régiókra oszt egy képet.

$$J_\epsilon(\phi, \mu_{in}, \mu_{out}) = \int_{\Omega} (f(x) - \mu_{in})^2 H_\epsilon(\phi(x)) dx_1 \dots dx_d + \\ \int_{\Omega} (f(x) - \mu_{out})^2 (1 - H_\epsilon(\phi(x))) dx_1 \dots dx_d + \\ v \int_{\Omega} \|\nabla \phi(x)\| \delta(\phi(x)) dx_1 \dots dx_d \quad (16.)$$

Ahol μ_{in} és μ_{out} az energia függvény két paramétere melyet a jellemző függvény paraméterei is lesznek. Továbbá az v egy vektorparaméter mely a régiók (az első két integrál) és kontúr energiája (az utolsó integrál) közti viszont fejezi ki. E kifejezést deriválva megkapjuk a jellemző függvényt:

$$\omega_\epsilon(x) = (f(x) - \mu_{in})^2 - (f(x) - \mu_{out})^2 \\ - v \operatorname{div} \left(\frac{\nabla \phi(x)}{\|\nabla \phi(x)\|} \right) \delta_\epsilon(\phi(x)) \quad (17.)$$

Általában az alakzatot leginkább a két régió tag határozza meg és a harmadik tag inkább egy megkötés, ellenszegül a határfelület terjedésének. Mivel a gyakorlatban sokkal gyakoribb, hogy a keresett alakzat és a tárgy egyértelműen meghatározza a határfelületet, elhagyjuk a harmadik tagot. Ezzel lényegesen felgyorsítjuk algoritmusunk futási idejét, hiszen a divergencia és a ∇ operátor költséges műveleteket jelent. Ugyanakkor Bernard kísérleti eredményei is alátámasztják e döntést [10]. A fenti egyenletekbe a Heaviside és a Dirac függvény, következő egyváltozós szabályosított alakjukat használjuk:

$$\begin{cases} H_\epsilon(x) = \frac{1}{2} + \frac{1}{\pi} \arctan\left(\frac{x}{\epsilon}\right) \\ \delta_\epsilon(x) = \frac{d}{dx} H_\epsilon(x) = \left(\frac{1}{\pi}\right) \frac{\frac{1}{\epsilon}}{1 + \left(\frac{x}{\epsilon}\right)^2} = \frac{1}{\pi\epsilon} \frac{1}{1 + \left(\frac{x}{\epsilon}\right)^2} \end{cases} \quad (18.)$$



E egyenletben ϵ egy valós pozitív konstans és a Dirac meg a Heaviside függvény a végtelen normalizált skálázását adja meg. Ennek elég nagyoknak kell lennie, hogy a határfelület terjeszkedése kihasson az egész szintbeállító függvényre és így az algoritmus végén megkapjuk annak globális minimumát [8]. Mivel a szintbeállító függvénye a maximális normalizálással a $[-1,1]$ intervallumba korlátozzuk, ennek értékét egyre állítjuk be.

Az μ_{in} és μ_{out} meghatározására az EM algoritmust használjuk. Nevét 1977-ben kapta mikor Dempster, Laird és Rubin a Royal Statistic folyóiratban először publikálták. Teljes megnevezése Expectation-Maximization és a maximum likelihood becslés iteratív számítására alkalmas főként olyan helyzetekben mikor hiányzó adataink vannak. Jelen esetben a μ_{in} és μ_{out} azok az értékek melyektől átlagosan a keresett tárgy, illetve a háttér intenzitása legközelebb található. Ezt előre viszont nincs, hogy honnan tudjuk, ezért a jellemzőfüggvény e két paraméterét is frissítjük minden iterációban.

Ennek lényege, hogy az energia és jellemző függvény értékét három dolog befolyásolja. A μ_{in} és μ_{out} paraméterek mellett, mint változó megjelenik a B-Spline együtthatók is. Ezért az optimalizálás egy-egy iterációja két lépésből áll. Először lekötjük a B-Spline együtthatókat és elvégezzük az energia minimalizálását a paraméterekre nézve. Ez felírható a következő alakban:

$$\begin{cases} \mu_{in} = \frac{\int_{\Omega} f(x) H_{\epsilon}(\phi(x)) dx_1 \dots dx_d}{\int_{\Omega} H_{\epsilon}(\phi(x)) dx_1 \dots dx_d} \\ \mu_{out} = \frac{\int_{\Omega} f(x) (1 - H_{\epsilon}(\phi(x))) dx_1 \dots dx_d}{\int_{\Omega} (1 - H_{\epsilon}(\phi(x))) dx_1 \dots dx_d} \end{cases} \quad (19.)$$

Ez diszkrét formában ekvivalens a jelenlegi tárgynak vet képpontok (μ_{in}) és a háttár képpontok átlag intenzitásával (μ_{out}) külön-külön. Az optimalizálás második lépését (az együtthatókra vonatkoztatva) meg a (7.) –es egyenletet felhasználva végezzük el. A gradiens lépését $\alpha_f = 1$ –el szorozzuk siker esetén és $\alpha_f' = 1.5$ –el osztjuk másképpen.

A kezdeti határfelületnek a kép közepébe a két átmérőjének negyede sugarú maszkot rajzolunk. A maszk egy bináris kép mely nulla és egyes értékekből áll. A kör belseje megfelel a keresett tárgynak és egyesekkel jelöljük a maszkban. A kezdeti szintbeállító függvény ennek lesz az előjeles távolság függvénye. E függvény felépítésére kiszámoljuk a maszk és az inverz maszk távolság függvényét. E kettőnek a különbsége meg a maszk mínusz még egy egység összege adja a keresett függvényt.



3.2. Diagramok

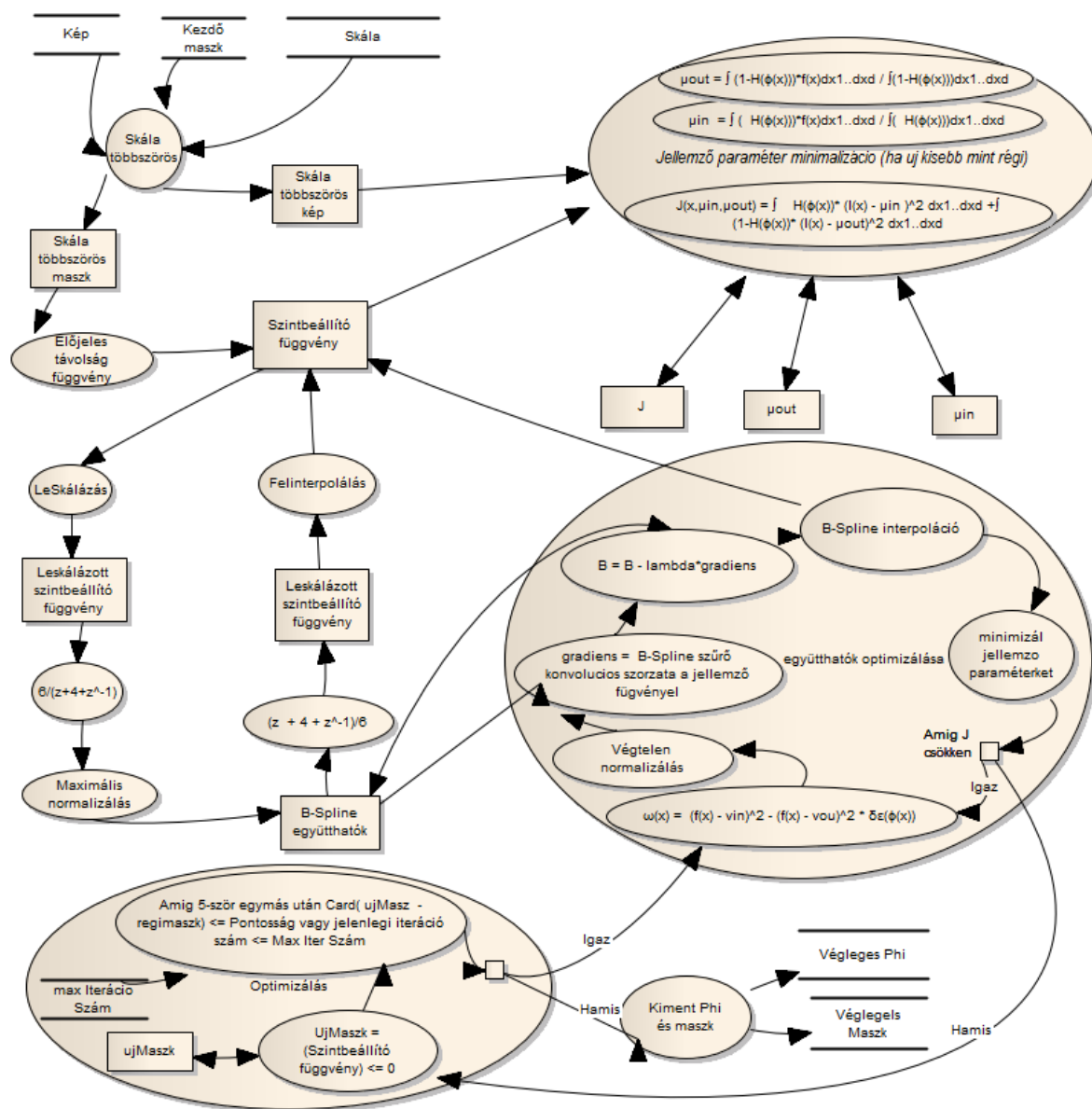
3.2.1. Osztálydiagram

Az algoritmust egy objektum orientált keretben valósítottam meg. Ezért az algoritmus be és kimeneti paraméterei egy osztály attribútum tagjai lettek. Az elvégezendő részműveletek az osztályba elrejtett (privát vagy védett) metódusokban található meg. Az osztály használatára csupán be kell állítani a bemeneti képet és a kezdeti maszkot (tehát milyen kezdeti határfelületből indul az algoritmus) és meghívjuk ennek a *Run ()* publikus eljárását.



3.2.2. Adatfolyam

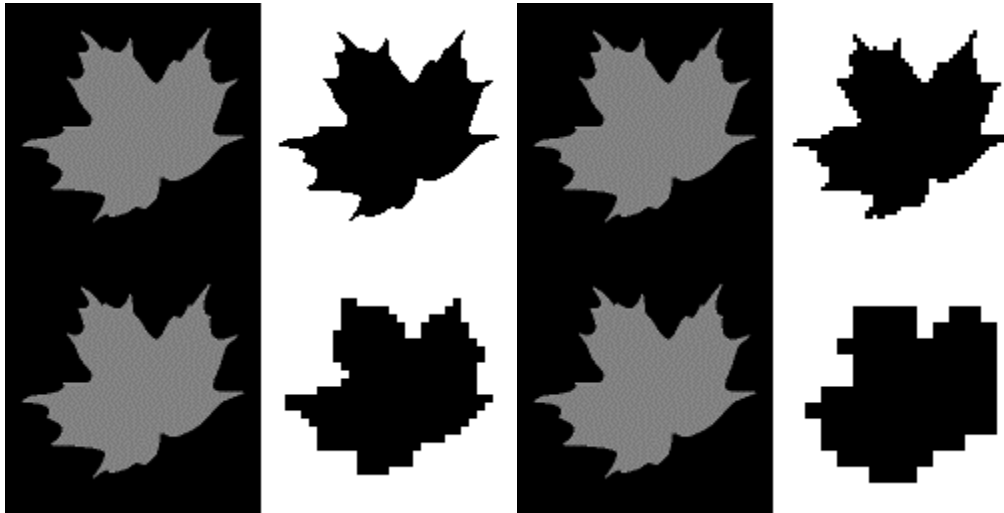
E ábrán megfigyelhetjük, hogy az adatok milyen átalakulásokon mennek keresztül, hogy megkapjuk a keresett tárgy alakzatát. Fontos megjegyezni, hogy ezen ábra nem tartalmaz minden apró részletet. Célja csupán, hogy egy jó képet adjon azon részfeladatokról melyet meg kell oldanunk.



4. SZEGMENTÁCIÓS EREDMÉNYEK

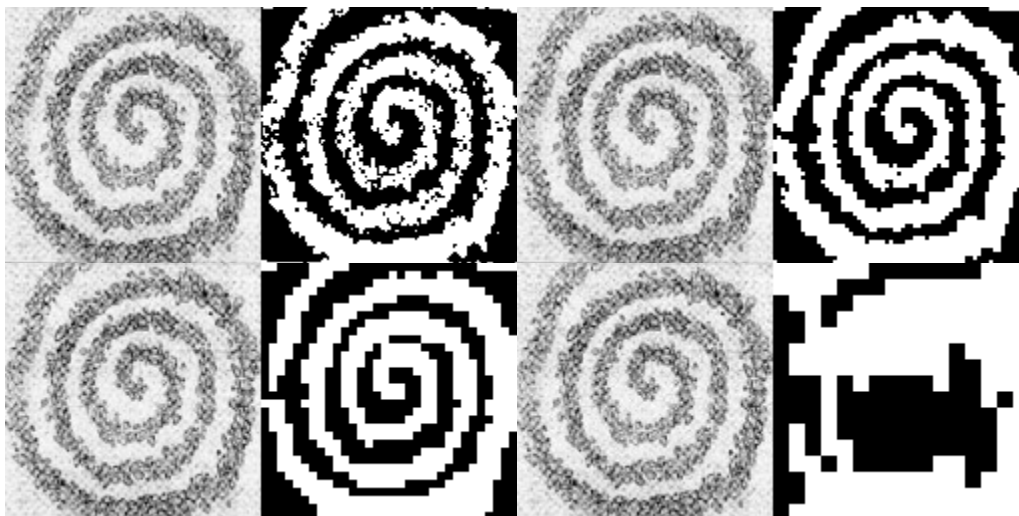
A tesztelés során megfigyeltük az algoritmus teljesítményét (futási időbe mérve), iterációs lépés számát és ugyanakkor megfigyeltük, hogy a diszkrét B-Spline – nak más-más mértéket véve (0, 2, 4 és 8-as lépésekkel), hogy befolyásolja a kialakuló szintfüggvényt.

4.1. Szimulációs képek



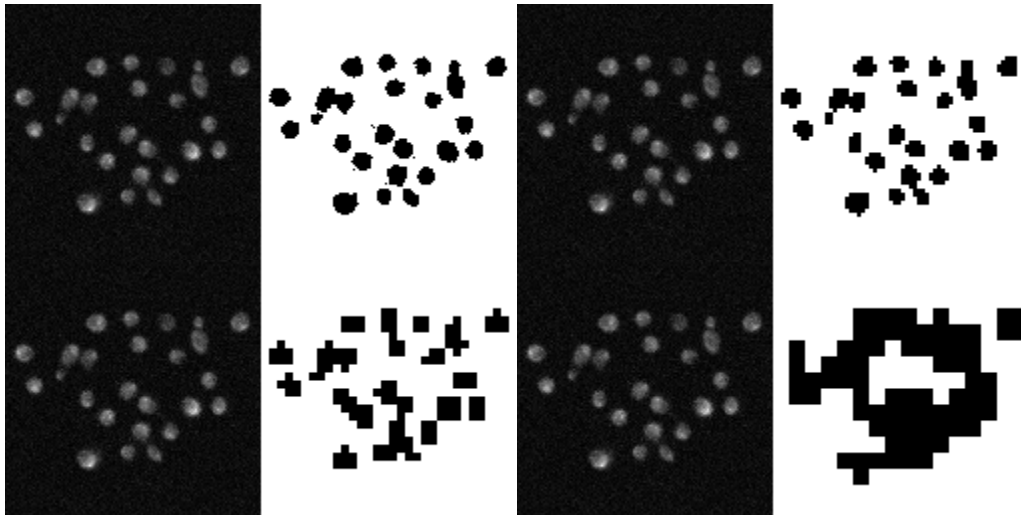
A fenti ábrán egy levél szimulálta formáját keressük. A bal felső sarokból rendre 0, 2, 4 és 8 lépésméretet használva a diszkrét B-Spline – nak. Észrevesszük, hogy a növekedő lépés szám egyre inkább eldobja az apró részleteket és durvább határfelületet eredményez.

4.2. Való világi képek



E képsorozat felépítésében követi a fentit. A növekvő lépés a zajokat egyre jobban kiszűri, de ha igen magas lépést használunk a B-Spline már nem rendelkezik elég kép információval és az azonosítás meghiúsul.

4.3. Orvosi képek



E ábrán egy mikroszkóp fluoreszkált felvételét láthatjuk. Az érdekelt sejtek a kezelés hatására a háttérhez képest világosabbak lesznek. E képen a sejtek megkeresése könnyen elvégezhető programunkkal, hiszen az topológiai rugalmas és képes a terjeszkedés során osztódni és a kezdeti határfelületen kívül újakat is létrehozni.

5. KÖVETKEZTETÉS

A fentiekben leírt módszert sikerült a C++ nyelvezetben objektum orientált ideológiákat követve megvalósítani és annak működését letesztelni különböző környezetű és minőségű kép esetében. Megfigyelhető volt, hogy az algoritmus gyorsan és könnyedén detektál alakzatokat, melynek nem létezik, esetleg nem tudjuk mértani formáját. Még akkor is igaz ez, ha ez nem egyetlen objektumot alkot, hanem több kis tárgy formájában található szétszóródva a képen.

A variációs modell melyre építünk társítva a B-Spline interpolációs függvényekkel lehetővé teszi, hogy a határfelületet globális szinten terjesztessük tovább az EM algoritmus iterációi során. Az energia függvény minimalizálását a szintbeállító felületet leíró B-Spline együtthatókra nézve a változó léptékű első fokú gradiens módszer segítségével oldottuk meg.

Az alkalmazott módszer előnye, hogy amikor a szintbeállító függvényünk közel van a megoldáshoz csupán pár iteráció szükséges, hogy a képhez rendelt energia függvény minimális legyen. Ez hasznos lehet egy forma követés feladatában, hiszen két kép kocka között többnyire kis különbség merül fel, amelyet az algoritmusunk hamar korrigálni tud. Ugyanakkor az alkalmazott módszer eredménye egy szintbeállító függvény lesz. Ez jól leírja, hogy egy-egy képpont mily mértékben tartózik az energia kritérium függvény által leírt tárgyhöz, háttérhez vagy amennyiben közel van a nulla szinthez, magát a tárgy határfelületéhez.

A zajos rendszerek esetében, vagy ha csökkenteni akarjuk a cél határfelület simaságát (akár zajkiszűrés miatt) jól alkalmazható a B-Spline diszkrét alakjának skálázása. A B-Spline csomópontok közti távolság növelésével a kapott határfelület egyre durvább lesz, ahogy ezt megfigyelhattük a tesztelés során.

A rendszer továbbfejlesztési lehetősége közé tartozik ennek implementációja és adaptációja egy képsorozat anyagra ahol egy bizonyos tárgyat próbálunk követni. Ilyen például egy echó kardiográf felvételen a bal szívkamra azonosítása és követése. Ennek sikere után fontos információ lenne az orvos számára, ha ennek a kerületét és területére vonatkozó információkat kiszámolnánk és esetleg egy grafikonon megjelenítenénk valós időben. Ennek segítségével könnyedén detektálható például az esetleges szívkamra zavarok jelenléte és annak mértéke egy beteg echókardiográf felvételén.



Továbbá az algoritmus sok-sok helyen használ úgynevezett batch műveleteket. Ezek könnyen párhuzamosíthatóak és lehetőséget adnának az algoritmus valós idejű alkalmazására (esetleg kis holtidő jelenlétében). Első lépésben ez történhet a központi processzor magra az OpenMP segítségével majd egy még látványosabb gyorsulásért a grafikai processzor magon felhasználva az Nvidia CUDA architektúráját, avagy az OpenCL keretet.

Didaktikai és kutatási célokból hasznos lépés lenne a szintbeállító felület dinamikus kirajzolása például az OpenGL grafikai csomaggal a határfelület terjeszkedése közben. Legvégül, de nem utolsó sorban megvalósíthatunk egy grafikai felületet ahol a felhasználó bármilyen általa betöltött képre letesztelhetné az algoritmus és annak teljesítményét.

Mint az előbbi paragrafusokban láthattuk e dolgozatban bemutatott algoritmus annak ellenére, hogy még gyerek cipőben jár nagyon ígéretes. Kevesebb, mint két éves és alkalmazási lehetőségei, területek ahol bevethető csak a fejlesztő fantáziáján múlik, hiszen egy topológiai rugalmas és nagyon párhuzamosítható módszert ad homogén régiók szegmentációira. A modern orvosi tudományban rengeteg ilyen helyzet merül fel, legyen szó echó kardiográf, röntgen, mikroszkóp, avagy 2D, esetleg 3D MRI kép és video felvételről.



IRODALOMJEGYZÉK

- [1] James C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*. MA, USA: Kluwer Academic Publishers Norwell, 1981.
- [2] John Canny, "A Computational Approach to Edge Detection," *IEEE Transactions On Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 6, pp. 679-698, November 1986.
- [3] D. H. Ballard, "Generalizing the Hough transform to detect arbitrary shapes," *Pattern Recognition*, vol. 13, no. 2, pp. 111-122, September 1981.
- [4] Milan Sonka, Vaclav Hlavac, and Roger Boyle, *Image processing, analysis, and machine vision*, 1st ed. London, UK: Chapman & Hall Computing, 1993.
- [5] D. Mumford and J. Shah, "Boundary Detection by Minimizing Functionals," *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, pp. 22-26, 1985.
- [6] D. Mumford and J. Shah, "Optimal Approximations of Piecewise Smooth Functions and Associated," *Communications on Pure and Applied Mathematics*, vol. 42, pp. 577-685, 1989.
- [7] Michael Kass, Andrew Witkin, and Demetri Terzopoulos, "Snakes: Active Contour Models," *International Journal of Computer Vision*, vol. 1, no. 4, pp. 321-31, 1988.
- [8] Tony F. Chan and Luminita A. Vese, "Active Contours Without Edges," *IEEE Transactions on Image Processing*, vol. 10, no. 2, pp. 266-277, February 2001.
- [9] Xavier Bresson, "Image segmentation with variational active contours," École polytechnique fédérale de Lausanne, Lausanne, Phd Thesis no. 3283, 2005.
- [10] Olivier Bernard, Denis Friboulet, Philippe Thévenaz, and Michael Unser, "Variational B-Spline Level-Set: A Linear Filtering Approach for Fast Deformable Model Evolution," *IEEE Transactions On Image Processing*, vol. 18, no. 6, pp. 1179-1191, June 2009.



- [11] Stanley Osher and Ron Fedkiw, *Level Set Methods and Dynamic Implicit Surfaces*. New York, USA: Springer-Verlag, 2002.
- [12] Michael Unser, Akram Aldroubi, and Murray Eden, "B-Spline Signal Processing: Part I - Theory," *IEEE Transactions On Sigal Processing*, vol. 41, no. 2, pp. 821-832, February 1993.
- [13] Michael Unser, Akram Aldroubi, and Murray Eden, "B-Spline Signal Processing: Part II - Efficient Design and Applications," *IEEE Transactions On Signal Processing*, vol. 41, no. 2, pp. 834-848, February 1993.
- [14] Michael Unser, "Splines: A perfect fit for signal and image processing," *IEEE Transactions on Signal Processing*, vol. 10, no. 2, pp. 22-38, November 1999.
- [15] Gilles Aubert, Michel Barlaud, Olivier Faugeras, and Stéphanie Jehan-Besson, "Image segmentation using active contours: calculus of variations or shape gradients?," *SIAM Appl. Math.*, vol. 63, no. 6, pp. 2128-2154, 2003.
- [16] Olivier Bernard, "Segmentation in echocardiographic imaging using parametric level set model," Institut National des Sciences Appliquees, Lyon, PhD Thesis in Computer Science 2006-ISAL-0096, 2006 December.

