

TDK-DOLGOZAT

Koós Krisztián

Szegedi Tudományegyetem
Természettudományi és Informatikai Kar
Képfeldolgozás és Számítógépes Grafika Tanszék

**Diszkrét tomográfiai algoritmusok optimális
paramétereinek meghatározása**

TDK dolgozat

Készítette:
Koós Krisztián
Programtervező informatikus MSc
I. évfolyam

Témavezető:
Dr. Nagy Antal
egyetemi adjunktus

Szeged
2012

Tartalomjegyzék

Bevezetés.....	4
1 Diszkrét tomográfia.....	5
1.1 Általánosságban a tomográfiáról.....	5
1.2 Diszkrét tomográfia.....	5
1.3 A probléma megoldása.....	6
2 Tomográfia formálisan.....	7
2.1 A probléma leírása.....	7
2.2 A SART algoritmus.....	8
3 Megvalósított algoritmusok.....	11
3.1 A DART algoritmus.....	11
3.1.1 Áttekintés.....	11
3.1.2 Formális definíció.....	13
3.2 Az ADART algoritmus.....	16
3.3 Implementációs megjegyzések.....	18
4 Eredmények.....	19
4.1 Fantomképek.....	19
4.2 Paraméterválasztás.....	20
4.3 Az algoritmusok hatékonysága.....	23
4.4 DART és ADART összehasonlítása.....	25
4.5 Zajjal terhelt vetületek.....	27
4.6 Különböző szűrők alkalmazása zajos környezetben is.....	29
Összefoglalás.....	32
Irodalomjegyzék.....	33
Köszönetnyilvánítás.....	34
Mellékletek.....	35

Bevezetés

A tomográfia feladata egy adott objektum rekonstruálása annak vetületeiből (projekcióiból). A diszkrét tomográfiában olyan megkötéseket lehet feltenni, mint például az objektum homogenitása, vagy legalábbis hogy néhány különböző anyagból tevődik össze.

A probléma megoldására számos módszer létezik. Ezek legtöbbje úgynevezett folytonos rekonstrukciós módszer, vagyis nem használnak fel a priori információt, miszerint a rekonstruálandó objektum csak néhány különböző elnyelődési együtthatóval rendelkező anyagból áll. Név szerint ilyen a klasszikus szűrt visszavetítés, a Total Variation Minimization és a dolgozatban bemutatásra kerülő SART (Simultaneous Algebraic Reconstruction Technique, Szimultán Algebrai Rekonstrukciós Technika) algoritmus [1].

A dolgozat fő témája az újnak számító DART (Discrete Algebraic Reconstruction Technique, Diszkrét Algebrai Rekonstrukciós Technika) és ADART (Adaptive DART, Adaptív DART) algoritmusok bemutatása és paramétereik elemzése. Ezek a rekonstrukciós problémát a priori információk felhasználásával oldják meg. Számos helyen alkalmazzák őket, és a tapasztalat alapján jó tulajdonságokkal rendelkeznek.

1 Diszkrét tomográfia

1.1 Általánosságban a tomográfiáról

A tomográfia feladata egy adott objektum vetületeiből visszaállítani az objektum keresztmetszeti képét. A probléma matematikailag jól definiált, ám a gyakorlatban az mégsem alkalmazható, mert rengeteg vetületre (esetenként több százra) van szükség. Ezek előállítását a legtöbb esetben a vetületképző eljárás anyagot károsító hatása miatt nincs lehetőség.

A vetületket többféleképpen állíthatjuk elő. Ilyen módszerek a röntgen-, elektron- vagy neutronsugárzás. Azonban közös jellemzőjük, hogy károsítják a vizsgált anyagot, így törekedni kell, hogy a sugárzást minél inkább csökkentjük. A vetületképzés további jellemzője, hogy az emissziós vagy transzmissziós sugárzással készül. Az emissziós röntgensugárzás esetében a vizsgált anyagba juttatnak sugárzó anyagot. Ez a sugárzást kívülről detektálható. Transzmissziós sugárzás esetén külső forrásból kell indítani a sugárzást amely az objektumon áthaladva bizonyos mennyiségben elnyelődik, és a túloldalon lemérhető annak intenzitása. A forrás és detektor távolságából, az anyagra jellemző elnyelődési együtthatóból, illetve a kezdeti és detektált intenzitásból kiszámítható az elnyelődés mértéke [4]. A dolgozatban csak a transzmissziós tomográfiáról lesz szó.

1.2 Diszkrét tomográfia

A diszkrét tomográfia azzal az esettel foglalkozik, mikor kevés vetület áll rendelkezésre az objektum visszaállításához és tudjuk róla, hogy csak néhány különböző fizikai anyagból tevődik össze, melyek más, ismert elnyelődési együtthatókkal rendelkeznek. Így a rekonstruálandó képen is csak néhány, előre ismert szürkeintenzitás jelenthet meg. Ha az objektum homogén, a rekonstruált képe két értékkel írható le, ebből az egyik az objektum környezetét jelenti (fekete), a másik pedig magát az objektumot (fehér). Ez a megkötés nagyban lecsökkenti a probléma megoldásához szükséges vetületszámot, mely a legtöbb esetben 2-10 között van. Az előbb említett homogén objektumok esetét gyakran bináris tomográfiának hívják, mely a diszkrét tomográfia egy egyedi esete, de

ezek az algoritmusok általában valamilyen speciális tulajdonságot feltételeznek az objektumról [4].

1.3 A probléma megoldása

A probléma megoldására számos módszer létezik már, melyek eltérnek minőségükben, futási idejükben, futtatásukhoz szükséges feltételekben. A klasszikus szűrt visszavetítést (Filtered Backprojection, FBP) felváltják a rekonstrukciós problémát egy nagy lineáris egyenletrendszerrel leíró és azt iteratíván megoldó algoritmusok. A futási idejük ugyan nagynak mondható, de megvan az az előnyük, hogy lényegesen kevesebb vetítésből is nagy pontosságú eredményt tudnak szolgáltatni [1].

Számos rekonstrukciós algoritmus jóval pontosabb eredményt tud szolgáltatni, ha rendelkezésére áll valamilyen a priori információ. Ez azt is jelenti, hogy sok esetben kevesebb vetület szükséges ugyanolyan minőségű eredményhez, továbbá kevesebb hatással van rá a zaj.

A prior tudás a tomográfia számos alkalmazási területén rendelkezésre áll. Az ipari tárgyak általában néhány anyagból állnak (pl. műanyag, alumínium) [1, 4]. Ez a tény jelen dolgozat kísérleteiben is fel van használva: a fantomképek, melyeken a kísérletek végezve lettek, szintén csak néhány (2-6) szürkeárnyalatból tevődnek össze. Esetenként az objektum geometriai tulajdonságai lehetnek ismertek, mint például a hv-konvexitás [4].

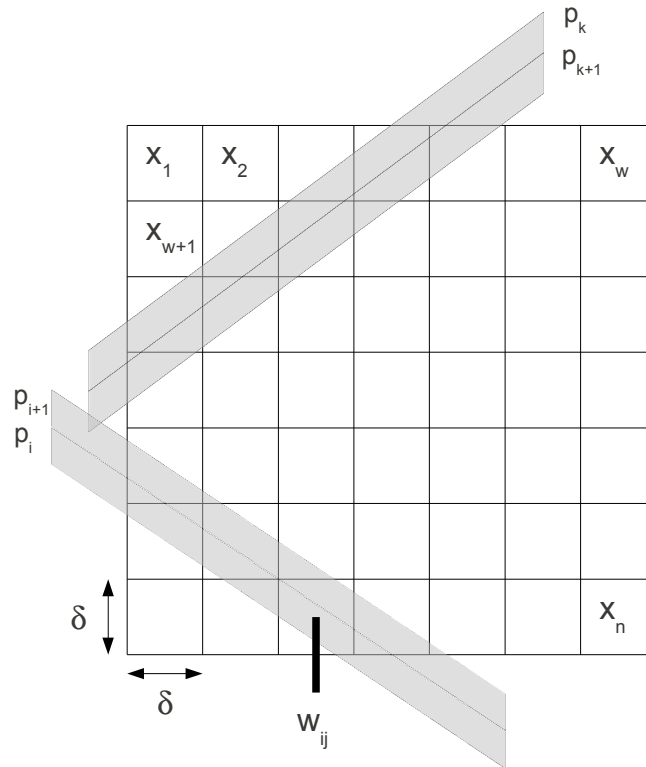
2 Tomográfia formálisan

2.1 A probléma leírása

A problémát egy lineáris egyenletrendszerrel írjuk le [1]. Így kétdimenziós képek rekonstrukcióját írjuk le azok egydimenziós vetületiből. A rekonstruált képet egy $n=w \times h$ méretű négyzetrács írja le, ahol w a kép szélességét, h pedig a magasságát jelöli. A vetületek értelmezhetőek úgy, mint a vetületi irányhoz tartozó detektorértékek vektorai, melyek forognak az objektum körül a vetület képzésekor. Jelölje d a detektorok, k pedig a vetületek számát. Minden vetületi irányon végighaladva összesen tehát $m = d \cdot k$ detektorértéket kapunk. Jelölje a minden vetületi irányon végighaladva kapott detektorértékek vektorát $p = (p_i) \in \mathbb{R}^m$. A vetületképzés leírható egy mátrixszorzással. Jelölje W a vetületi mátrixot (projekciós mátrix), melynek mérete $m \times n$, $x = (x_i) \in \mathbb{R}^n$ pedig az objektumot, csak négyzetrácsos ábrázolás helyett azt sorfolytonosan egy vektorra alakítva. A $W = (w_{ij})$ vetületi mátrix egyes értékei azt jelzik, hogy egy pixel értékei mennyire játszottak szerepet a detektált érték meghatározásakor. w_{ij} tehát a j -edik pixel hatását jelöli az i -edik mérésre. A vetületképzés leírása tehát formailag a $W \cdot x = p$ képlettel történik. Egy sematikus ábrája látható a 2.1 ábrán.

Az általános tomográfiai rekonstrukciós probléma ezek alapján a következőképpen írható le. Adott egy $W = (w_{ij})$, $\forall i, \forall j \ w_{ij} \in \mathbb{R}_{\geq 0}$ vetületi mátrix és egy $p \in \mathbb{R}^m$ detektorértékeket tartalmazó vektor. Keressük azt az $x \in \mathbb{R}^n$ vektort, amely kielégíti a $W \cdot x = p$ egyenletrendszert.

Abban az esetben, ha objektum anyagainak sugárelnyelődési együtthatója ismert (vagyis ismertek az intenzitásértékei digitális ábrázolásban), ez felhasználható a probléma leírásánál. Ekkor legyen szintén $W = (w_{ij})$, $\forall i, \forall j \ w_{ij} \in \mathbb{R}_{\geq 0}$ a vetületi mátrix, $p \in \mathbb{R}^m$ pedig a detektorokkal mért adatok. Legyen továbbá $l > 0$ a különböző szürkeintenzitások száma és $R = \{\rho_1, \dots, \rho_l\}$ a szürkeintenzitásokat leíró vektor.



2.1. ábra: A vetületképzés sematikus ábrája. A négyzetrácsban a négyzetek oldalhossza δ . Az x értékek a pixelek intenzitásait, a p vektor pedig a detektorokon mért értékeket tartalmazza. A W_{ij} érték a j -edik pixel i -edik mérésre való hatását jelenti.

A gyakorlatban rengetegszer előfordul az, hogy az előbbi egyenletrendszereknek nincs megoldásuk. Ez annak a következménye, hogy a vetületképzés gyakran zajos vagy hibásan működnek a detektorok. Ilyen esetben egy közelítő megoldást keresünk, melyre $\|W \cdot x - p\|$ minimális (valamilyen normával számolva). Dolgozatomban ismertetett algoritmusokat tesztelem ideális és zajos környezetben is.

2.2 A SART algoritmus

A DART és ADART algoritmusok, melyeket a 3. Megvalósított algoritmusok fejezet mutat be, úgynevezett folytonos rekonstrukciós és diszkretizációs lépések között iterálnak. A folytonos rekonstrukciós lépések során a rekonstruálandó képre mint egy valós értékeket felvevő tömbre tekintünk. A diszkretizációs lépések pedig felhasználják az a priori tudást, vagyis az ismert szürkeárnyalatokat. A folytonos lépéshez számos

algebrai rekonstrukciós algoritmus használható. Ilyenek az ART, SART, SIRT [4]. Az elvégzett kísérletekhez a SART algoritmus töltötte be ezt a szerepet, melyet bemutat jelen fejezet.

Az alábbiakban a SART algoritmus egy véletlenszerűsített változata olvasható. Az algoritmus minden egyes iterációja végighalad az összes vetületi irányon. Hogy ez milyen sorrendben történjen, számos módszer határozhatja meg. Jelen esetben ez lesz véletlenszerű, innen kapta az algoritmus ezt a jelzőjét.

A vetületi irány kiválasztásához vezessük be a következő jelöléseket [1]. Bontsuk szét a W mátrixot és p vektort d darab blokkra. (d jelöli a vetületszámot.) Ezek a blokkok k sorból állnak, hiszen egy irányból készített vetület során k darab detektor van. Jelölje ezeket rendre

$$W = \begin{pmatrix} W^1 \\ \vdots \\ W^d \end{pmatrix} \quad (2.1)$$

és

$$p = \begin{pmatrix} p^1 \\ \vdots \\ p^d \end{pmatrix}. \quad (2.2)$$

Minden $W^t = (w_{ij}^t)$ tehát egy irányhoz tartozó vetületi mátrix, p^t pedig a hozzá tartozó mérések.

Legyen

$$\beta_i^t = \sum_{j=1}^n w_{ij}^t, \quad i = 1, \dots, k, \quad t = 1, \dots, d \quad (2.3)$$

és jelölje

$$\gamma_j^t = \sum_{i=1}^k w_{ij}^t, \quad j = 1, \dots, n, \quad t = 1, \dots, d. \quad (2.4)$$

Az irányok sorrendiségét a következőképpen határozzuk meg. Legyen S_d az a halmaz, mely az $1, \dots, d$ számok minden lehetséges permutációját tartalmazza, és σ ennek a halmaznak egy véletlenszerűen kiválasztott eleme. Ez egy vektort jelent, mely tartalmazza az összes irány sorszámát valamilyen sorrendben. Legyen $r^{(s)} = p^{\sigma(s)} - W^{\sigma(s)} x^{(s-1)}$, amely a mért adatok és az aktuális rekonstrukció különbségét jelöli.

Az előbbiek felhasználásával az algoritmus iteratívan számítja ki egy becslést a rekonstrukcióra egy $x = x^{(0)}$ választott kezdőértékkel. Jelölje $x^{(s)}$ ($s = 1, 2, \dots$) az aktuális becslést, mely a következő képlettel számolható ki az $x^{(s-1)}$ előző becslésből:

$$x_j^{(s)} = x_j^{(s-1)} + \lambda \frac{1}{y_j^{\sigma(s)}} \sum_{i=1}^k \frac{w_{ij}^{\sigma(s)} r_i^{(s)}}{\beta_i^{\sigma(s)}}, \quad j = 1, \dots, n \quad (2.5)$$

ahol λ relaxációs faktor. A minden vetületi irányon végighaladó iterálást nevezzük egy SART iterációnak.

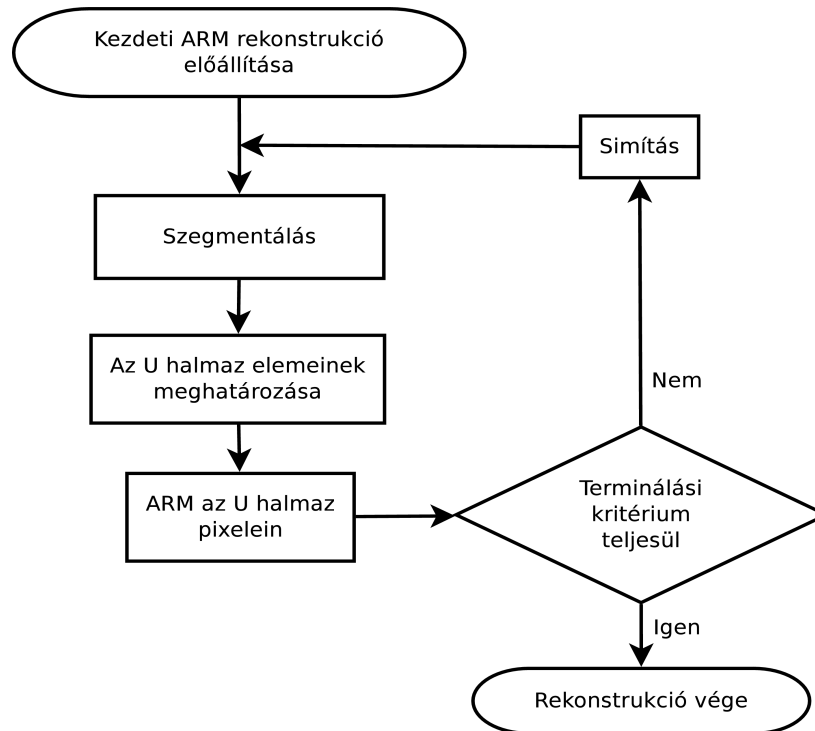
3 Megvalósított algoritmusok

Dolgozatom elkészítéséhez két algoritmust valósítottam meg az [1], [2] és [3] publikációk alapján. Ezek a DART és az ADART. Mindkettőre igaz, hogy egy másik rekonstrukciós algoritmust használnak szubrutinként. A DART esetén alkalmasak ART, SART, SIRT, stb., melyeket általánosan ARM (Algebraic Reconstruction Method, Algebrai Rekonstrukciós Módszer) néven említünk. Az ADART esetében ez az algoritmus maga a DART. Jelen fejezet ezeket az algoritmusokat írja le mind általánosságban, mind formálisan.

3.1 A DART algoritmus

3.1.1 Áttekintés

A DART algoritmus folyamatábrája a 3.1 ábrán látható. Az algoritmusnak szüksége van egy folytonos rekonstrukciós algoritmusra, melyek általános elnevezése az ARM. Első lépésként egy folytonos rekonstrukciót állít elő (a választott ARM által), ezt követik a diszkretizációs lépések. Az algoritmus azzal a feltevessel él, hogy a szürkeárnyalatok ismeretében az ARM által szolgáltatott pixel értékek a legtöbb esetben helyesek. Vagyis felmerül az ötlet, hogy találjunk olyan pixeleket, melyek értékei kérdésesek lehetnek és ne számoljunk azokkal, melyek valószínűleg már helyesek. Ez nagy mértékben csökkentheti a szükséges számításokat, vagyis a futási időt. Bizonyos feltételek teljesülése esetében (ilyen pl. az elegendő vetületszám) az vehető észre, hogy az első lépés után a háttérhez és az objektum belső területébe tartozó pixelek értékei közel megegyeznek az elvárttal. Így a kérdéses pixelek az objektum körvonalait jelentik. Az ismert szürkeárnyalatok felhasználásával küszöbölt képen keressük meg a pixelek B halmazát, mely az objektum körvonalához tartozó pixeleket tartalmazza. Azok a pixelek tartoznak a körvonalhoz, melyek 8-szomszédai közül több mint b darab eltérő intenzitású. A legtöbb esetben ez az érték 0, de fontos szerepe lesz ennek a paraméternek az ADART algoritmus esetében. Most tartozzanak tehát azok a pixelek a körvonalhoz, melyek érintkeznek legalább egy eltérő intenzitású pixellel.



3.1. ábra: A DART algoritmus folyamatábrája

A körvonal pixeljeinek ily módú meghatározása nem mindig elegendő. Abban az esetben, ha az objektum lyukat tartalmaz és annak körvonala nem lett észlelve a kezdeti ARM rekonstrukció során, később sem fog előkerülni. Ezt megelőzve határozzuk meg a pixeleknél azon U halmazát, melybe a B halmaz elemein felül véletlenszerűen kiválasztott elemek is tartoznak. Helyesebb ezt a halmazt szabad pixelek halmazának hívni. Projekciós hiba és zaj esetén is növeli a hatékonyságot ez a módszer. A valószínűség, mellyel egy pixel szabaddá válik, paraméter. Minél több pixel lesz szabad, vagyis minél nagyobb a valószínűségi érték, annál lassabb lesz az algoritmus, de nagy mértékű zaj esetén annál jobb a teljesítmény [1].

Ezt követően ismét néhány ARM iteráció következik, de most csak azon pixelekre végezzük a számítást, melyek az U halmazban vannak. Vagyis itt már alkalmazva van a feltételezés, miszerint a körvonal pixeljei nagyobb valószínűséggel helytelenek mint a többi. Fontos jellemző, hogy ez a (két) halmaz minden egyes iterációban változhat, követi a rekonstruált objektum körvonalát.

Az iteráció utolsó lépéseként a szabad pixelekre elvégzünk egy simítást. A használt konvolúciós maszk és annak mérete szintén lehet paraméter. Ennek elvégzésével befejeződött egy DART iteráció.

Ezt követően a megállási kritériumot ellenőrizzük. Amennyiben nem teljesül, az iteráció a küszöbölési lépéstől folytatódik. Megállási feltételnek alkalmazható maximális iterációs szám, de mivel kísérleti objektumonként ez a szám nagyban eltérhet, használata pazarló a futási időt tekintve. Általánosan a teljes projekciós hiba használatos annak meghatározására, hogy mikor termináljon az algoritmus, melyet a (3.1) képlet definiál.

$$E(x) = \| Wx - p \|_2 \quad (3.1)$$

Mivel ennek számítása költségesnek mondható, minden 3 iteráció után történik meghatározása. Amennyiben 3 iteráció után ez az érték nem csökken, az algoritmus terminál. Mivel a két módszer együtt is alkalmazható, kísérleteim során a maximális iterációs szám 500-ra lett állítva.

3.1.2 Formális definíció

Jelen fejezet formálisan mutatja be a DART algoritmust. Az algoritmusnak paraméterként szüksége van a p mért detektorértékekre és a szürkeárnyalatok

$R = \{ \rho_1, \dots, \rho_l \}$ halmazára. Természetesen ismernie kell az algoritmusnak a vetületek számát is. További konstansok, amelyek befolyásolják az algoritmus futását, az a pixelek f szabadsági valószínűsége (mely a fejezetben később részletesebben definiálva lesz) és az ARM esetleges további paraméterei (ilyen a SART esetében a λ relaxációs együttható).

Először egy kezdeti $x^{(0)}$ rekonstrukciót kell előállítani az ARM segítségével. Ezt követően az algoritmus egy iterációs eljárásba lép, melynek lépései az alábbiak [1]:

1.) Szegmentálás: A szegmentálási lépés során olyan képet kell előállítani, melynek minden intenzitása csak az $R = \{ \rho_1, \dots, \rho_l \}$ halmaz elemeinek valamelyike. Ez megvalósítható pont operációként, vagyis egy küszöbölt képpont intenzitása csakis az ugyanazon pozíciójú képponttól függ.

Legyen $x^{(t-1)}$ az aktuális rekonstrukció a t -edik iterációban. Ennek felhasználásával számoljuk ki a szegmentált $s^{(t)} \in \mathbb{R}$ rekonstrukciót. A használandó küszöbértékeket a $\tau_1, \dots, \tau_{l-1}$ értékek jelöljék, ahol

$$\tau_i = \frac{\rho_i + \rho_{i+1}}{2} . \quad (3.2)$$

Az $r: \mathbb{R} \rightarrow \mathbb{R}$ küszöbölési függvény legyen a következő:

$$r(v) = \begin{cases} \rho_1 & (v < \tau_1) \\ \rho_2 & (\tau_1 \leq v < \tau_2) \\ \vdots & \\ \rho_l & (\tau_{l-1} \leq v) \end{cases} . \quad (3.3)$$

Rövidített jelölésként vezessük be egy $x \in \mathbb{R}^n$ képre értelmezett küszöbölést a következőképpen:

$$r(x) = (r(x_1) \ r(x_2) \ \dots \ r(x_n))^T . \quad (3.4)$$

2.) Szabad pixelek választása: Először a körvonal pixeleket tartalmazó $B^{(t)} \subset \{1, \dots, n\}$ halmazt határozzuk meg az $s^{(t)}$ szegmentált rekonstrukcióból. Az i -edik pixel 8-szomszédait jelölje $N(i) \subset \{1, \dots, n\}$. Egy $s_i^{(t)}$ pixel akkor tartozik a $B^{(t)}$ halmazba, ha $s_i^{(t)} \neq s_j^{(t)}$ teljesül legalább egy $j \in N(i)$ esetén. A szabad pixelek $U^{(t)}$ halmazát két lépésből állíthatjuk elő: elsőként legyen $U^{(t)} = B^{(t)}$, majd f valószínűséggel válasszunk további pixeleket véletlenszerűen a halmazba. Ez lehetőséget ad arra, hogy olyan területek rekonstrukciója is változzon, melyek egyébként nem kerülnének a határpixelek halmazába.

Megjegyzendő, hogy az algoritmusnak kétféle változata létezik: véletlenszerűsített és nem véletlenszerűsített. A nem véletlenszerűsített során a szabad pixelek halmazába nem választunk véletlenszerűen további pixeleket. Ez megfelel a véletlenszerűsített egy speciális esetének, mikor $f = 0$.

3.) ARM a szabad pixeleken: A 2. Tomográfia formálisan című fejezetben

leírtak szerint a rekonstrukciós problémát a következő képlet írja le:

$$\begin{pmatrix} | & & | \\ w_1 & \dots & w_n \\ | & & | \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} = p, \quad (3.5)$$

ahol w_i jelöli a W vetületi mátrix i -edik oszlopvektorát. Definiáljuk egy x_i változó $v_i \in \mathbb{R}$ értéken való rögzítését. Ez átalakítja az előbbi egyenletrendszert oly módon, hogy az egyenletek száma ugyanannyi marad, de az ismeretlenek száma eggyel csökken, és a következőképpen néz ki:

$$\begin{pmatrix} | & & | & | & & | \\ w_1 & \dots & w_{i-1} & w_{i+1} & \dots & w_n \\ | & & | & | & & | \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_{i-1} \\ x_{i+1} \\ \dots \\ x_n \end{pmatrix} = p - v_i w_i. \quad (3.6)$$

Jelentse $F^{(t)} = \{1, \dots, n\} \setminus U^{(t)}$ a rögzített pixelek halmazát. Az algoritmus iterációi során minden $i \in F^{(t)}$ pixel rögzítve van a neki megfelelő $s_i^{(t)}$ értéken. Így az egyenletrendszer ismeretleneinek száma n -ről lecsökken $n - |F^{(t)}|$ értékre. Az így kapott $\tilde{W} \tilde{x} = \tilde{p}$ egyenletrendszeren bizonyos számú ARM iterációt hajtunk végre. Tegyük fel, hogy a legtöbb pixel a helyes értéken lett rögzítve. Ekkor az algoritmus jobb eredményt szolgáltat a szabad pixelekre nézve mintha nem lennének rögzített pixelek [1]. Ez annak a következménye, hogy a módszer a rekonstrukciós hibákat nem a kép teljes egészére vetíti vissza, hanem csak az U halmaz elemeire.

4.) Simítás: Az, hogy az ARM iterációkban csak a szabad pixelek változnak, azt eredményezi, hogy azon pixelek értékei változnak, melyek feltehetően még nem a helyes értéket vették fel. Viszont a módszer nem biztosítja azt, hogy ezen változtatás megfelelő az eredmény tökéletesítését illetően. A szabad pixelekre szétosztott hibákat egy simítással kiegészítve jobb eredményeket kaphatunk. A simító konvolúciós maszk sokféle lehet és nem csak 3x3-as méretű alkalmazható. A leggyakrabban használt

operátort az $x_i^{(t)} = 0.7x_i^{(t)} + 0.3b$ képlet írja le, ahol b az i -edik pixel 8-szomszédainak átlagos intenzitását jelöli [2, 3]. Végeztek továbbá kísérleteket 1 sugarú Gauss szűrővel [1] és medián szűrővel is [6]. Ezt a lépést tehát az U halmaz elemeire kell végrehajtani.

5.) Megállási feltétel vizsgálata: A DART egy iteratív, heurisztikus algoritmus. Nem garantálható, hogy minden esetben konvergál a helyes megoldáshoz. Általában elmondható, hogy az első néhány iterációban ugyan gyorsan javul a rekonstrukció, később ez lelassul. Így szükséges egy módszer, mellyel leállítjuk a futást. Célszerű egy maximális iterációs számot beállítani, de mivel ez képfüggő, szokás használni az $E: \mathbb{R}^n \rightarrow \mathbb{R}$ úgynevezett teljes projekciós hibát:

$$E(x) = \| Wx - p \|_2, \quad (3.7)$$

ahol más norma is alkalmazható.

3.2 Az ADART algoritmus

Az ADART a szabad pixelek választását definiálja felül a DART algoritmushoz képest. Az ADART algoritmus minden esetben 8 iterációból áll. Legyen ennek az iterációnak a ciklusváltozója a DART algoritmus áttekintésében bemutatott b változó. Ez a 0-tól 7-ig terjedő értékeket fogja felvenni. Minden iteráció során a DART algoritmust hajtjuk végre. A szabad pixeleket határozzuk meg úgy, hogy azok tartoznak ebbe a halmazba, melyek 8-szomszédai közül legalább b darab eltérő intenzitású. Ez azt jelenti tehát, hogy az első iterációban ($b = 0$) azon pixeleket tartoznak a halmazba, melyek 8-szomszédai között van tőle eltérő intenzitású, a másodikban ($b = 1$) több mint egy eltérő intenzitású van, a harmadikban ($b = 2$) pedig 2-nél több szomszédos pixel intenzitása tér el, és a sort lehetne tovább folytatni.

Az eljárás mögötti gondolat az, hogy a $B^{(t)}$ halmazba tartozó pixelekre végrehajtott ARM iteráció után nagy annak a valószínűsége, hogy a helyes értéket vették fel, de annak is, hogy a halmaz egy elemének legalább 1 szomszédos pixelének intenzitása eltérő. Az ADART iterációk során ez a feltétel tovább erősödik. Ez tovább csökkenti, viszonylag nagy mértékben, a rekonstrukció lineáris egyenletrendszerében

szereplő ismeretlenek számát.

Az ADART algoritmus a [3] publikációban lett közzétéve nem véletlenszerűsített DART algoritmust használva. Ki kell térnünk a véletlenszerűsített DART használhatóságára is. Mivel az adaptív algoritmus lényege, hogy tovább csökkentse a lineáris egyenletrendszerben szereplő ismeretlenek számát, de ennek definíciója az első iterációban teljesen megegyezik az alap DART algoritmusban használt definícióval, így a véletlenszerű pixelválasztás az adaptív algoritmus esetén csak az első iterációban nyer értelmet. Ez azt jelenti, hogy a DART algoritmus f szabadsági valószínűségi paramétere az első iterációt követően mindig 0 értéket kell, hogy felvegyen, vagyis az első iterációt követően mindig nem véletlenszerűsített DART algoritmust kell használni.

A 3.2. ábra az ADART algoritmus vázlatos leírását tartalmazza. Megjegyzendő, hogy a **for** ciklust elhagyva a DART algoritmust kapjuk (ahol $b = 0$ minden esetben).

```

 $t := 0$  ;
 $x^{(t)}$  kezdeti rekonstrukció számítása;
 $s^{(t)} = r(x^{(t)})$  szegmentált kép számítása;
for  $b := 0$  to 7
  do
     $t := t + 1$  ;
    A körvonal pixeleit (eltérő 8-szomszédjainak száma  $> b$ ) tartalmazó
       $B^{(t)}$  halmaz számítása  $s^{(t-1)}$  alapján;
    A szabad pixelek  $U^{(t)}$  halmazának számítása;
    Az  $y^{(t)}$  számítása, ahol  $y_i^{(t)} = x_i^{(t-1)}$  ha  $i \in U^{(t)}$ 
      és  $y_i^{(t)} = s_i^{(t-1)}$  egyébként;
    Az  $x^{(t)}$  ARM rekonstrukció számítása  $y^{(t)}$  kezdeti megoldással az
       $U^{(t)}$  halmaz elemein;
    Simítás az  $x^{(t)}$  rekonstrukció  $U^{(t)}$  halmazba tartozó pixelein;
    Szegmentálás:  $s_i^{(t)} = r(x_i^{(t)})$  ha  $i \in U^{(t)}$ ,  $s_i^{(t)} = s_i^{(t-1)}$  egyébként;
  while a megállási feltétel nem teljesül;
end
return  $s^{(t)}$  ;

```

3.2. ábra: Az ADART algoritmus fő lépései

3.3 Implementációs megjegyzések

Az algoritmusokat MATLAB nyelvvel valósítottam meg kihasználva annak speciális adattípusait a hatékonyság érdekében. A zaj hozzáadása C nyelven írt programmal történt. A C függvények MATLAB-ban való használatához egy úgynevezett mex függvényt kell elkészíteni. Ez a MATLAB Compiler szolgáltatásaként nyújtott *mex* paranccsal fordítható le a célarchitektúrára.

Az implementációban a kezdeti rekonstrukció 10 iterációt, a későbbi ARM futtatások pedig 3-at jelentenek.

4 Eredmények

Ebben a fejezetben bemutatásra kerülnek a kísérletek, azok eredményei, valamint a használt fantomképek. A kísérletek az algoritmusok legjobb paramétereinek kiválasztását, az algoritmus minőségének bemutatását és a két implementált algoritmus egymással való összehasonlítását foglalják magukba.

A rekonstrukció minőségének méréséhez a rekonstruálandó kép ismeretében a hibásan rekonstruált pixelek számát, valamint a csak bináris képre értelmezhető Relative Mean Error (RME) képlet lett használva, mely a következő [5]:

$$RME(x^*, y) = \frac{\sum_{i=1}^n |x_i^* - y_i|}{\sum_{i=1}^n (x_i^*)} . \quad (4.1)$$

A fenti képletben az x^* az eredeti képet (illetve az ideális rekonstrukciót) jelöli, míg az y egy tetszőleges rekonstrukciót. Az RME értelmezése a helytelenül visszaállított pixelek száma lenormalizálva az ideális képen az objektumot reprezentáló pixelek számával. Megjegyzendő, hogy mivel az érték függ az objektumot leíró pixelek számától, ez lehet nagyobb 1-nél.

4.1 Fantomképek

A kísérletekhez 10 kép lett felhasználva, melyek az 4.1 ábrán láthatóak. A képek mérete az egyes kísérletekben 128×128 vagy 256×256 pixel.

phantom2_2_linear: Egy nem konvex objektumot ábrázol, melynek körvonalai meglehetősen összetett.

phantom3_linear: Egy gyújtómotor hengerfejének keresztmetszeti képét ábrázolja.

6_circle: 6 különböző méretű és elhelyezkedésű kört ábrázol.

ring_6_circle: Az előbbi 6_circle képet tartalmazza egy gyűrűben.

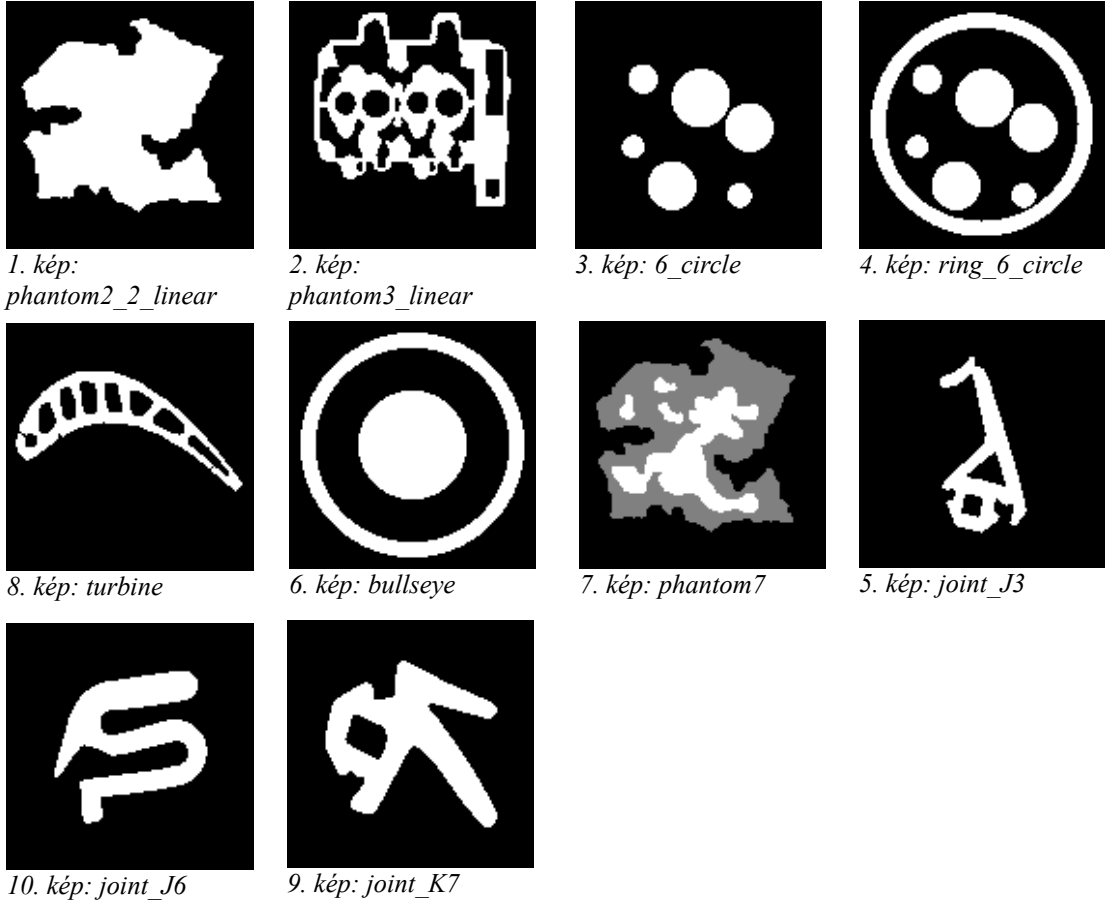
turbine: Egy turbinalapát keresztmetszeti képét ábrázolja.

bullseye: Egy a kép közepén elhelyezkedő kört tartalmaz egy gyűrű.

phantom7: A phantom2_2_linear képen látható objektumot tartalmazza azzal a különbséggel, hogy az objektumot két szürke intenzitás írja le, így a háttérrel három

különböző intenzitás van jelen a képen.

joint_J3, joint_J6, joint_K7: Absztrakt alakzatok.



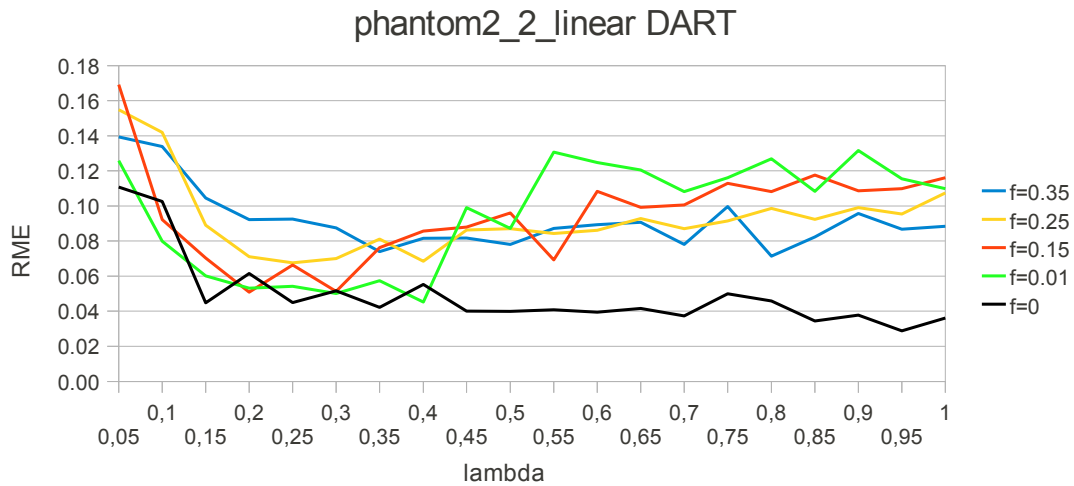
4.1. ábra: Felhasznált fantomképek

4.2 Paraméterválasztás

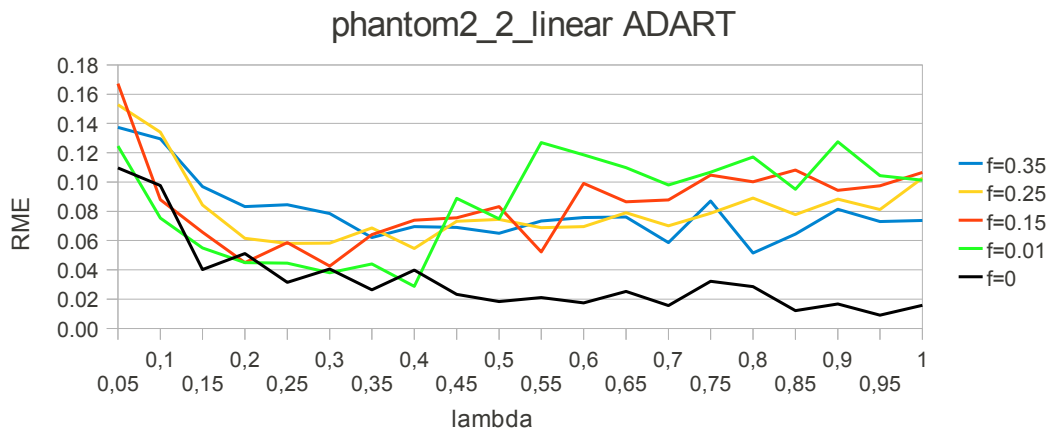
Megvizsgáltam néhány jelentős paramétert kiválasztva, hogy milyen értékekre adnak legjobb eredményt az implementált algoritmusok. Ezek a paraméterek a SART algoritmus λ paramétere és a DART algoritmus f szabadsági valószínűsége.

Ehhez a kísérlethez a *phantom2_2_linear* és *6_circle* kép lett felhasználva 128×128 pixel méretben, rendre 4 és 5 vetülettel. A két képen a $\lambda \in \{0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45, 0.5, 0.55, 0.6, 0.65, 0.7, 0.75, 0.8, 0.85, 0.9, 0.95, 1\}$ és $f \in \{0.35, 0.3, 0.25, 0.2, 0.15, 0.1, 0.05, 0.01, 0\}$ paraméterek összes

lehetséges kombinációjával lefuttattam 5 alkalommal az algoritmusokat. Egy alkalommal ez képenként 180 futást jelentett. Összesen pedig körülbelül 17 órát tett ki. Az egy képhez és azonos paraméterekkel kapott RME értékeket átlagoltam, majd vonaldiagramot készítettem belőlük algoritmusra és képre lebontva.



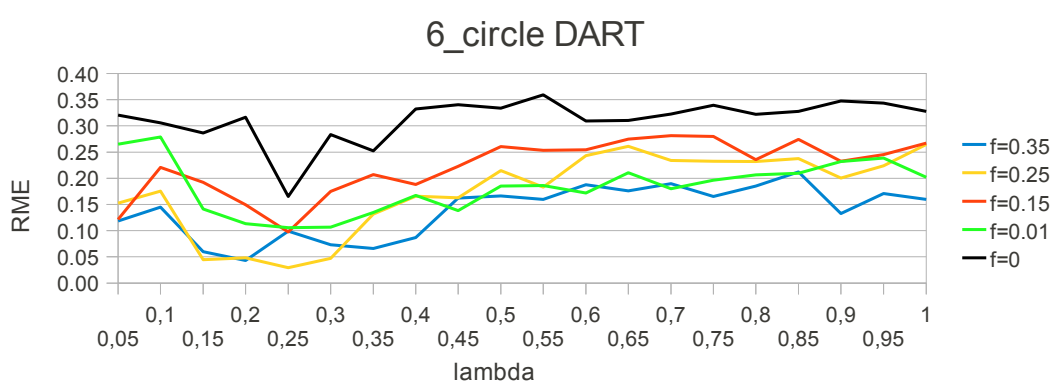
4.2. ábra: Paraméterválasztás 1



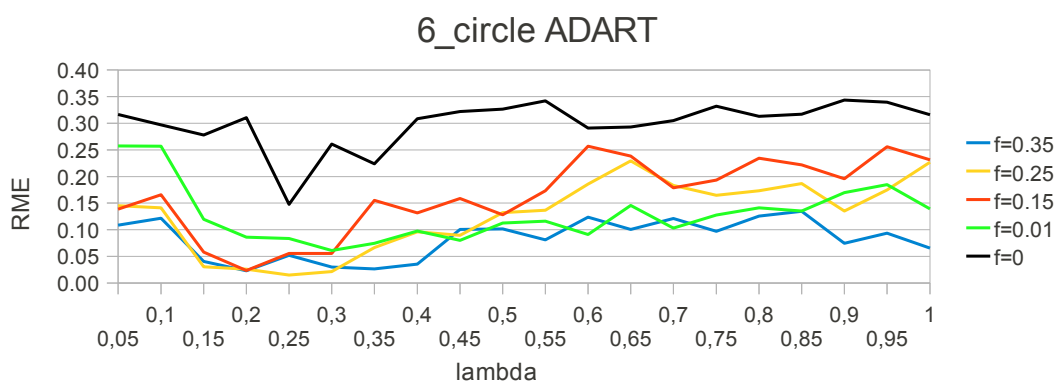
4.3. ábra: Paraméterválasztás 2

A 4.2 és 4.3 ábrán rendre a DART és az ADART algoritmussal kapott, a *phantom2_2_linear* képen futtatott átlagos eredmények láthatóak. Megfigyelhető, hogy a $\lambda \leq 0.1$ értékekre mindig a rossz eredmény született azonos f , de más λ

mellettihez képest. A legjobb eredményt az $f=0$ paraméter szolgáltatja szinte minden λ érték esetén. Ahogy azonban a következő kép vizsgálatából látszódní fog, ez annak köszönhető, hogy elegendő volt a vetületszám. Az $f=0.35$ paraméter mondhatni minden λ paraméterrel hasonló eredményt adott. A többi f értékre igaz, hogy 0.15 és 0.4 közötti λ értékekre adták a legjobb RME értéket, továbbá minél kisebb volt az f érték, az eredmény is annál jobb volt. Nagyobb λ értékekre pedig ez fordítva igaz, vagyis minél nagyobb az f , annál jobb az RME érték.



4.4. ábra: Paraméterválasztás 3



4.5. ábra: Paraméterválasztás 4

A 4.4 és 4.5 ábrán mivel a használt vetületszám kevésnek bizonyult, az előzőekkel ellentétben az $f=0$ paraméter minden esetben a legrosszabb eredményt szolgáltatja. Viszont most is igaz, hogy bármilyen szabadsági valószínűség érték mellett a 0.1 és 0.4

közötti λ értékek között találhatóak a legjobb RME eredmények, és e paraméter növekedésével romlanak.

A paraméterválasztás konklúziója tehát a következő. Amennyiben ismerjük a rekonstruálandó objektum tulajdonságait és tudni lehet, hogy elegendő vetület készült, az algoritmusok nem randomizált változatának használata javasolt nagy λ paraméter (0.85-1.0) alkalmazása mellett. Minden más esetben, vagyis ha a vetületszám nem elegendő, nem ismertek az eredeti kép szürkeintenzitásai, esetleg zajjal terheltek a vetületek, magasabb f értékkel 0.1 és 0.4 közötti λ paraméterrel fognak legjobban viselkedni az algoritmusok. Nem szabad azonban elfelejteni, hogy az alacsony f értékek mellett erős igazán a két algoritmus, ennek növelésével a futási idő is nagy mértékben növekszik, ezért igyekezni kell olyan környezet kialakítására, ahol ezt a paraméter értéket alacsonyan lehet tartani. K. J. Batenburg és társa DART algoritmussal kapcsolatos publikációjában [1] az $f=0.15$ használata javasolt.

4.3 Az algoritmusok hatékonysága

A következő kísérletek annak érdekében lettek elvégezve, hogy bemutassák, az algoritmusok képesek megoldani a rekonstrukciós problémát. A 4.1 fejezetben ismertetett 256×256 pixel méretű képek mindegyikéhez meghatároztunk egy vetületszámot, amelynél az algoritmusok már jó eredményt szolgáltatnak (vagyis a vetületszám növelésével járuló konvergenciája az eredménynek lelassul). Ezekkel a vetületekkel és a 4.2 fejezetben leírtak alapján a legjobbak közé tartozó $f=0.05$ és $\lambda=0.3$ paraméterekkel minden képre lefuttattuk az algoritmusokat.

DART					
fantom	RME	pix. err.	idő (s)	#iter	proj.
phantom2_2_linear.png	0.0333	881	107	270	5
phantom3_linear.png	0.0389	632	223	145	12
6_circle.png	0.0231	193	71	140	6
ring_6_circle.png	0.0387	716	92	125	7
turbine.png	0.0252	159	124	85	11
bullseye.png	0.0125	250	40	100	5
phantom7.png	0.0154	526	150	130	10
joint_J3.png	0.0355	207	69	105	7
joint_J6.png	0.0175	188	88	135	7
joint_K7.png	0.0149	198	87	115	7

4.1. táblázat: DART algoritmus futásai a kísérleti képeken

ADART					
fantom	RME	pix. err.	idő (s)	#iter	proj.
phantom2_2_linear.png	0.0318	842	120	305	5
phantom3_linear.png	0.0372	605	296	195	12
6_circle.png	0.0207	173	96	195	6
ring_6_circle.png	0.0345	638	155	220	7
turbine.png	0.0202	127	209	150	11
bullseye.png	0.0111	223	62	160	5
phantom7.png	0.0154	526	207	185	10
joint_J3.png	0.0367	214	90	140	7
joint_J6.png	0.0168	181	116	180	7
joint_K7.png	0.0144	191	128	165	7

4.2. táblázat: Az ADART algoritmus futásai a kísérleti képeken

Az említett paraméterekkel kapott eredményeket az 4.1 és 4.2 táblázat mutatja, rendre a DART és az ADART algoritmus futásait. Továbbá néhány rekonstruált kép látható a 4.6 ábrán. A táblázatokból leolvasható, hogy a legrosszabb eredmény esetén is 0.04 alatt marad az RME érték, ami a hibás pixelek számát tekintve kevesebb mint 1,5%-ot jelent. A futási idő a DART algoritmus esetében 1 és 5 perc közötti volt. Az ADART algoritmus esetében valamivel tovább tart a futás. Ez azért van, mert felhasználja a DART algoritmust, így legalább annak ideje szükséges a termináláshoz. Ezen kísérletek során az átlagos futási idő 105 és 148 másodperc volt. Csak egyetlen

alkalommal érte el a maximális 5 percet, ez pedig a phantom3_linear kép volt. Megfigyelhető, hogy a futási idő növekedése sosem érte el a DART algoritmus idejének kétszeresét. Például a turbine képen a DART algoritmus 124 másodperc után, míg az ADART algoritmus 209 másodperc után terminált, viszont az RME érték 0.0252-ről 0.0202-re csökkent, ami 20%-os javulást jelent.

További észrevétel, hogy a képen ábrázolt objektum bonyolultsága, vagyis a jó eredmény eléréséhez szükséges minimális vetületszám és a futási idő között szoros összefüggés van. Ez azt is jelenti, hogy több vetülettel több számítást kell végezni. Az 5-7 vetületet igénylő fantomképekre futott le leggyorsabban az algoritmus, a 10-11-12 vetületet igénylő 3 kép pedig ilyen sorrendben a leglassabban.



4.6. ábra: Az algoritmus hatékonyságát bemutató rekonstruált képek. Felül a DART, alul pedig az ADART algoritmus eredményei.

4.4 DART és ADART összehasonlítása

A fejezet a két implementált algoritmus összehasonlításáról szól. Ehhez a kísérlethez a 4.3 fejezetben leírt fantomképek 128×128 méretű változata, és a legjobbak közé tartozó, kiválasztott paraméterek lettek felhasználva, szintén zaj nélküli vetületekkel. A vetületszámok viszont olyanok, amikor az algoritmusok még nem szolgáltatnak közel tökéletes eredményt. A választások nehézségének okozója az a jelenség, amikor egy

vetületszámhoz tartozó eredmény még elfogadhatatlan, mert nem ismerhető fel róla az eredeti objektum, de az eggyel nagyobb vetületszám esetén már közel hibátlan a rekonstrukció. Ez különösen a kis vetületszámot igénylő fantomoknál van így. A rekonstrukciók képenként 5 különböző vetületszámmal lettek elkészítve. Ezek az előző fejezetben az adott képhez elegendőnek kimondott vetületszám és 4-gyel kisebb szám közötti minden egész számot foglalják magukba (ügyelve arra, hogy ez az érték ne csökkenjen 2 alá). Az ideális eredmények a kísérlethez a 10-25%-ban hibásak lennének, de az említett probléma miatt a képenként 5 futtatásból azok lettek kiválasztva, amelyek esetében leglátványosabb a javulás, bemutatva ezzel, hogy mennyire képes javítani a DART algoritmus eredményén annak adaptív változata.

	DART		ADART				
phantom	RME	pix. err.	RME	pix. err.	proj.	RME javulás	pix. err. javulás
phantom2_2_linear.png	0.0334	221	0.0256	169	4	0.0079	52
phantom3_linear.png	0.1410	594	0.1099	463	9	0.0311	131
6_circle.png	0.1886	403	0.1175	251	5	0.0711	152
ring_6_circle.png	0.0486	231	0.0225	107	6	0.0261	124
turbine.png	0.1097	184	0.0334	56	7	0.0763	128
bullseye.png	0.0396	203	0.0049	25	4	0.0347	178
phantom7.png	0.0453	388	0.0356	305	7	0.0097	83
joint_J3.png	0.1185	179	0.0867	131	5	0.0318	48
joint_J6.png	0.0422	117	0.0072	20	5	0.0350	97
joint_K7.png	0.0314	106	0.0163	55	5	0.0151	51
átlagosan:						0.0339	104
max:						0.0763	178

4.3. táblázat: DART és ADART eredmények összehasonlítása

A 4.3 táblázat mutatja a kísérleti képeken tapasztalt, két algoritmus közötti rekonstrukciós eredményjavulást. A bemutatott esetekben mindig javított az eredményen az ADART algoritmus. Viszont ez nem bizonyítható, hogy mindig így van, valamint szélsőséges esetekben (vagyis ha túl rossz, illetve nagyon jó a rekonstrukció) enyhén gyengébb eredményt szolgáltat. A táblázatból leolvasható továbbá, hogy az adaptív eljárás képes akár 0.07-et javítani az RME értéken, vagy a hibás pixelek számát tekintve a maximális 178 alapján egy 128×128 pixel méretű képen 1%-kal közelebb

kerülni az eredeti fantomképhez. Az ADART algoritmust publikációjában a javulás mértéke 512×512 méretű képeken 3% körüli, mely pixelszámban kb. 6000-t jelent. A legnagyobb javulások rekonstrukciói láthatóak a 4.7 ábrán, amikor a 6_circle kép vetületei voltak az inputok, valamint a 4.8 ábrán, ami a bullseye kép rekonstrukcióit ábrázolja.



4.7. ábra: Az RME érték alapján legnagyobb javulást mutató rekonstrukció. Balról a DART, jobbról pedig az ADART algoritmus eredménye látható.



4.8. ábra: A hibás pixelek száma alapján legnagyobb javulást mutató rekonstrukció. Balról a DART, jobbról pedig az ADART algoritmus eredménye látható.

4.5 Zajjal terhelt vetületek

Mivel a rekonstrukciós algoritmusok gyakorlati alkalmazása során mindig jelen van valamilyen zaj (például a detektorok nem tökéletes működése), jelen fejezetben olyan kísérleteket szerepelnek, melyek során a vetületek zajjal terheltek. Ez 2, 5 és 10%-os Gauss zaj hozzáadását jelenti.

A vizsgálatok a korábban bemutatott 128×128 pixel méretű képeken és paraméterekkel készültek. Mivel a zaj mindenképpen negatívan hat a rekonstrukcióra, a vetületszám meg lett emelve a zajmentes vizsgálatokhoz képest.

Zaj		RME					
		2%		5%		10%	
phantom	proj.	DART	ADART	DART	ADART	DART	ADART
phantom2_2_linear.png	6	0.0165	0.0164	0.0216	0.0211	0.0285	0.0263
phantom3_linear.png	15	0.0445	0.0396	0.0577	0.0519	0.0927	0.0790
6_circle.png	8	0.0179	0.0169	0.0262	0.0249	0.0474	0.0426
ring_6_circle.png	10	0.0193	0.0144	0.0368	0.0332	0.0496	0.0455
turbine.png	15	0.0367	0.0319	0.0565	0.0467	0.0870	0.0736
bullseye.png	8	0.0158	0.0141	0.0215	0.0198	0.0329	0.0297
phantom7.png	14	0.0256	0.0233	0.0485	0.0433	0.0820	0.0666
joint_J3.png	12	0.0322	0.0280	0.0427	0.0374	0.0749	0.0667
joint_J6.png	10	0.0172	0.0176	0.0283	0.0262	0.0428	0.0373
joint_K7.png	10	0.0195	0.0181	0.0294	0.0262	0.0436	0.0376

4.4. táblázat: Zajjal terhelt vetületek rekonstrukciójának RME értékei

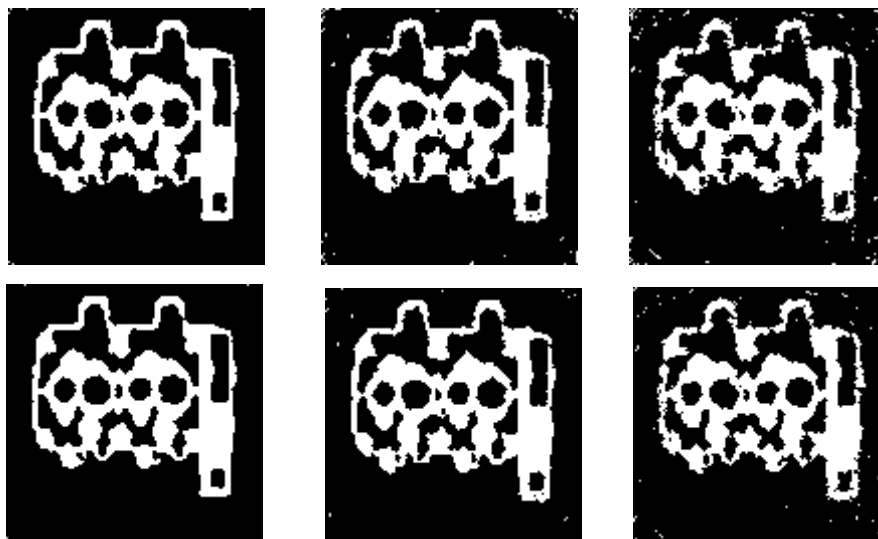
Zaj		Hibás pixelek száma					
		2%		5%		10%	
phantom	proj.	DART	ADART	DART	ADART	DART	ADART
phantom2_2_linear.png	6	436	435	571	559	756	697
phantom3_linear.png	15	723	644	937	843	1507	1284
6_circle.png	8	150	141	219	208	396	356
ring_6_circle.png	10	356	266	681	613	917	842
turbine.png	15	231	201	356	294	548	464
bullseye.png	8	316	283	430	396	659	595
phantom7.png	14	869	792	1644	1470	2766	2262
joint_J3.png	12	188	163	249	218	437	389
joint_J6.png	10	185	189	304	282	460	401
joint_K7.png	10	259	241	390	348	579	499

4.5. táblázat: Zajjal terhelt vetületek rekonstrukciójának hibás pixelszámai

A 4.4 és 4.5 táblázatban láthatóak a kísérlethez tartozó eredmények. Az algoritmusok szinte mindig képesek megbirkózni a feladattal és mind az RME értékek és hibás pixelek számainak alapján, mind látszólag elfogadható eredményt produkálni, konvergálni a helyes megoldás felé. Ez megfigyelhető a phantom3_linear kép eredményein, melyet a 4.9 ábra tartalmaz.

A rekonstrukciós hiba az objektum körvonalainál jelentkezik. A zaj úgy jelenik meg rajta, hogy az elmosódottnak, egyenetlennek vagy szakadozottnak tűnik. Az

objektum belsejében és az azon kívüli területen nem észlelhető akkora mértékben, viszont ezeken a részeken is jelennek meg lyukak vagy kis, elkülönülő területeken helytelenül az objektum részének tekintett intenzitások.



4.9. ábra: Zajos vetületek rekonstrukciói a phantom3 képen. A felső sor a DART rekonstrukció, míg az alsó az ADART algoritmus eredményei. Az oszlopok pedig balról jobbra rendre 2%, 5% és 10% zajjal terhelt vetületekből készültek.

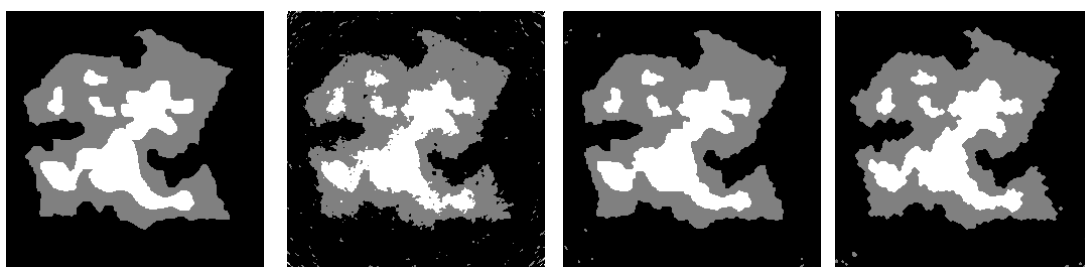
A legtöbb hibás pixel a legbonyolultabb képeknek mondható phantom3 és turbine képeken jelentkezett. Ezeknél volt szükség a legnagyobb vetületszámra. Ezt követi a több intenzitást tartalmazó phantom7 kép. Hibásan rekonstruált pixelek számát tekintve ehhez kapcsolódik a legrosszabb eredmény, mely még mindig csak kevesebb mint 5% hibát jelent. Az eredmények alapján elmondható, hogy az algoritmusok megfelelő vetületszám mellett képesek jó eredményt szolgáltatni.

4.6 Különböző szűrők alkalmazása zajos környezetben is

Az előző fejezetben az olvasható, hogy az algoritmusok képesek zajos környezetben is jól teljesíteni. Végeztek már kísérleteket különböző rekonstrukciós algoritmusokkal a simításnál használt szűrőket illetően. Balázs Péter és Hantos Norbert medián szűrőt alkalmazva hajtották végre vizsgálataikat [6]. Jelen dolgozatban kitérek különböző méretű és típusú szűrők hatékonyságára. Ez alatt az eredetileg javasolt szűrő, 1 sugarú

Gauss, valamint mediánszűrő értendő. Az eredetileg alkalmazottat csak 3×3 -as, míg a másik kettőt 5×5 -ös és 7×7 -es méretben is vizsgáltam. Ehhez a kísérlethez az algoritmusokat 256×256 méretű képeken hajtottam végre.

Az itt leírt következtetések az A melléklet alapján lettek levonva. Megoszlik a szűrők hatékonysága az egyes esetekben. Az eredetileg javasolt szűrő kis mértékű zaj esetén jobban teljesít a másik kettőnél. A zaj növekedésével azonban egyre többször nyújtja a legjobb eredményt a Gauss szűrő, 10%-os zajnál a DART algoritmus már nyolc esetben jobbnak minősül a tízből. Ez nem mondható el a mediánszűrőről az eredetivel szemben. Megerősíthető az utóbbi két filter azon ismert tulajdonsága, hogy jól eltünteti a zaj okozta jellegzetességeket [7]. Ugyanakkor simító hatásánál fogva ezek sem képesek tökéletes eredményt adni. Az 5×5 -ös és 7×7 -es méretűek sokszor hasonló értékeket szolgáltatnak, de esetenként lényegesen rosszabbat. A nagyobb méretűek használata akkor javasolt, ha a rekonstruált objektum körlapot, gyűrűt vagy hasonló szabályos alakzatot ábrázol, mint pl. a bullseye fantomkép.



4.10. ábra: A 10% zaj megjelenése a phantom7 tesztkép rekonstrukcióján. Balról az eredeti kép, majd rendre az eredeti szűrővel, a 3×3 -as Gauss szűrővel és 3×3 -as medián szűrővel futtatott DART algoritmus eredménye.

Az előbb említettek láthatóak a 4.10 ábrán a phantom7 inputkép példáján. Látható, hogy az eredeti szűrővel készült rekonstrukción nyúlványok és kiugró intenzitások formájában jelenik meg a zaj. A medián filter esetében a körvonal lesz egyenetlen. A Gauss szűrő esetén a finom részletek vesznek el, de mégis ez adja a legjobb eredményt.

Az ugyanakkora méretű Gauss és mediánszűrőket összehasonlítva elmondható róluk, hogy kis zaj esetén a mediánszűrés jobban teljesít. Figyelemre méltó, hogy az

eredeti vagy a legkisebb medián szűrést alkalmazva az ADART algoritmus egy kivételével minden esetben jobb eredményt adott az algoritmus alap változatához képest, a többi filter esetében pedig éppen fordítva, vagyis nagyon ritkán.

Összefoglalás

Dolgozatomban bemutatásra került a tomográfia, illetve a diszkrét tomográfia. Ismertette lett a SART folytonos rekonstrukciós algoritmus. Ezt követte a két megvalósított algoritmus, a DART és az ADART általános és formális leírása. A kísérletekből megállapítható, hogy az algoritmusok képesek megoldani a rekonstrukciós problémát. Működésüket zajos környezetben is teszteltem, gyakorlati használhatóságukat megmutatva ezzel. Megkerestem a vizsgált algoritmusok és a szubrutinként felhasznált SART algoritmus azon paramétereit, melyek mellett a legjobb eredményt nyújtják. A két algoritmus összehasonlításával azt a következtetést vontam le, hogy a DART algoritmusnak valóban továbbfejlesztettje az adaptív változata, vagyis jobb eredményt ad. Mivel az [1] és [2] publikációban eltérő simító maszk van használva (egyik a vizsgált pixel intenzitását 70%-ban, 8-szomszédainak átlagos értékét pedig 30%-ban veszi figyelembe, a továbbiakban eredetinek nevezve, a másik pedig 1 sugarú Gauss szűrő), megvizsgáltam ezek és a mediánszűrő zajos környezetben nyújtott minőségét. Az eredmény szerint kis zaj esetén az eredeti teljesít jobban, míg a zaj növekedésével a Gauss felülmúlja. 10% zajnál már az adja a legjobb értékeket. A mediánszűrőt kis zajnál érdemes preferálni a Gauss-szal szemben. Az ADART algoritmusnak nem kedvez a Gauss szűrő erős simító hatása, ezért ott minden esetben a kis méretű eredeti vagy medián használata javasolt.

Irodalomjegyzék

- [1] Kees Joost Batenburg, Jan Sijbers: *A practical reconstruction algorithm for discrete tomography*, IEEE Transactions on Image Processing 20 (2011) 2542-2553
- [2] Kees Joost Batenburg, Jan Sijbers: *DART: A fast heuristic algebraic reconstruction algorithm for discrete tomography*, IEEE International Conference on Image Processing 4 (2007) 133-136
- [3] F. Javier Maestre-Deusto, Giovanni Scavello, Joaquín Pizarro, Pedro L. Galindo: *ADART: An Adaptive Reconstruction Algorithm for Discrete tomography*, IEEE Transactions on Image Processing 20 (2011) 2146-2152
- [4] Balázs Péter: *Képrekonstrukció*, Egyetemi tananyag, Typotex Kiadó, (2011).
- [5] Varga László, Balázs Péter, Nagy Antal, *Direction-dependency of binary tomographic reconstruction algorithms*, Graphical Models 73 (2011) 365-375
- [6] Hantos Norbert, Balázs Péter: *Image Enhancement by Median Filters in Algebraic Reconstruction Methods: An Experimental Study*, Advances in Visual Computing, Lecture Notes in Computer Science 6455 (2010), 339-348
- [7] Czúni László, Tanács Attila: *Képi információ mérése*, Typotex Kiadó, (2011).

Köszönetnyilvánítás

Ezúton szeretném megköszönni témavezetőmnek, Dr. Nagy Antalnak a dolgozat elkészítéséhez nyújtott odaadó segítségét. Szintén köszönöm Varga László Gábornak az implementációval kapcsolatban nyújtott rendkívül hasznos tanácsait.

Mellékletek

A. melléklet:

		DART Pixel Error 2% zaj mellett						
		orig.	gauss				median	
	proj.	3x3	3x3	5x5	7x7	3x3	5x5	7x7
phantom								
phantom2_2_linear.png	6	436	800	1056	1040	503	552	1508
phantom3.png	15	723	1390	1775	1847	753	2180	3134
6_circle.png	8	150	258	332	330	213	215	509
ring_6_circle.png	10	356	687	765	752	631	757	997
turbine.png	15	231	336	405	428	271	460	837
bullseye.png	8	316	365	337	362	429	274	297
phantom7.png	14	869	933	1068	1022	951	1120	1633
joint_J3.png	12	188	278	368	374	195	344	589
joint_J6.png	10	185	267	345	313	266	238	500
joint_K7.png	10	259	348	395	445	296	283	639

		ADART Pixel Error 2% zaj mellett						
		orig.	gauss				median	
	proj.	3x3	3x3	5x5	7x7	3x3	5x5	7x7
phantom								
phantom2_2_linear.png	6	435	806	1077	1076	424	579	1769
phantom3.png	15	644	1559	1971	2042	519	2694	4735
6_circle.png	8	141	261	340	335	168	205	867
ring_6_circle.png	10	266	682	769	762	507	770	1370
turbine.png	15	201	368	473	495	191	593	2007
bullseye.png	8	283	359	337	358	370	266	253
phantom7.png	14	792	979	1163	1113	894	1333	2891
joint_J3.png	12	163	309	407	420	171	426	1085
joint_J6.png	10	189	281	362	331	191	235	704
joint_K7.png	10	241	348	413	450	234	276	971

		DART Pixel Error 5% zaj mellett						
		orig.	gauss				median	
	proj.	3x3	3x3	5x5	7x7	3x3	5x5	7x7
phantom								
phantom2_2_linear.png	6	571	908	1079	1037	725	735	1428
phantom3.png	15	937	1346	1852	1821	1063	2241	3011
6_circle.png	8	219	302	361	371	290	290	518
ring_6_circle.png	10	681	716	805	735	832	859	995
turbine.png	15	356	357	475	433	401	586	873
bullseye.png	8	430	458	384	387	632	435	330
phantom7.png	14	1644	1174	1250	1291	1306	1337	1696
joint_J3.png	12	249	310	351	386	300	390	587
joint_J6.png	10	304	326	324	328	417	299	459
joint_K7.png	10	390	426	450	434	416	325	630

		ADART Pixel Error 5% zaj mellett						
		orig.	gauss				median	
	proj.	3x3	3x3	5x5	7x7	3x3	5x5	7x7
phantom								
phantom2_2_linear.png	6	559	923	1105	1075	701	771	1760
phantom3.png	15	843	1465	2025	2026	899	2749	4632
6_circle.png	8	208	298	364	380	253	288	894
ring_6_circle.png	10	613	718	811	740	808	846	1355
turbine.png	15	294	365	526	499	336	708	1967
bullseye.png	8	396	449	378	376	627	410	253
phantom7.png	14	1470	1167	1310	1333	1245	1496	2942
joint_J3.png	12	218	333	399	422	257	485	1061
joint_J6.png	10	282	337	341	343	360	299	741
joint_K7.png	10	348	410	465	451	388	324	957

		DART Pixel Error 10% zaj mellett						
		orig.	gauss				median	
	proj.	3x3	3x3	5x5	7x7	3x3	5x5	7x7
phantom								
phantom2_2_linear.png	6	756	927	1073	1050	843	727	1554
phantom3.png	15	1507	1771	2074	1917	1506	2288	3030
6_circle.png	8	396	366	388	371	426	352	527
ring_6_circle.png	10	917	832	868	828	1205	923	1074
turbine.png	15	548	424	548	474	561	551	842
bullseye.png	8	659	548	572	529	786	572	466
phantom7.png	14	2766	1561	1451	1531	1960	1527	2024
joint_J3.png	12	437	370	415	388	396	443	576
joint_J6.png	10	460	375	355	380	451	374	496
joint_K7.png	10	579	485	478	471	498	424	637

		ADART Pixel Error 10% zaj mellett						
		orig.	gauss				median	
	proj.	3x3	3x3	5x5	7x7	3x3	5x5	7x7
phantom								
phantom2_2_linear.png	6	697	943	1077	1068	817	746	1830
phantom3.png	15	1284	1902	2286	2126	1444	2825	4685
6_circle.png	8	356	349	397	373	405	364	892
ring_6_circle.png	10	842	806	849	833	1123	970	1418
turbine.png	15	464	408	588	522	535	643	1908
bullseye.png	8	595	512	549	517	769	563	435
phantom7.png	14	2262	1421	1429	1503	1839	1737	3127
joint_J3.png	12	389	372	442	425	361	524	1104
joint_J6.png	10	401	385	367	393	417	393	749
joint_K7.png	10	499	467	477	477	478	412	994