

TDK DOLGOZAT

ML-EM képrekonstrukciós algoritmus transzmissziós tomográfiára

Molnár Balázs

Témavezető: Dr. Légrády Dávid
egyetemi docens
Nukleáris Technika Intézet
Budapesti Műszaki és Gazdaságtudományi Egyetem

BME
2014

Absztrakt

A TDK-dolgozatban egy transzmissziós tomográfias képalkotásra alkalmas algoritmust ismertetek. Elméleti háttérét a Lange és Carson által transzmissziós tomográfiára javasolt iteratív ML-EM algoritmus adja [1]. Az iterációs sémának részletes elemzését és a megvalósítási lehetőségeket [2]-ben már ismertettem, a jelenlegi dolgozat a grafikus kártyán történt implementációra, a módszer matematikai fantomok rekonstrukciójával végrehajtott konvergencia-vizsgálatára, valamint a valós mérési adatokon való tesztelésre fókuszál.

A fejlesztés során elértem az algoritmus nagyfokú gyorsítását, kihasználva, hogy hagyományos CPU helyett Nvidia GTX 690 típusú grafikus kártyán programozva párhuzamos feldolgozás érhető el. Verifikáltam az előrevetítés eredményei a Matlab Radon-transzformáltjával összevetve. Kétdimenziós eloszlások rekonstrukcióját elemezve megállapításokat tettem a konvergenciára, a rekonstrukció szabad paramétereinek függvényében is. A képminőségre vonatkozólag bevezettem a kvantitatív elemzésre is gyakran használt L2 normát, és vizsgáltam az elvégzett iterációk számának, futási időnek tekintetében. A valós mérési eredmények feldolgozásának problémáit, és lehetséges megoldásukat is ismertetem a 6. fejezetben, valamint bemutatom a végrehajtott rekonstrukciót is, összehasonlítva a szűrt visszavetítés eredményeivel.

Önállósági nyilatkozat

Alulírott Molnár Balázs, a Budapesti Műszaki és Gazdaságtudományi Egyetem fizikus MSc szakos hallgatója kijelentem, hogy ezt a TDK dolgozatot meg nem engedett segédeszközök nélkül, önállóan, a témavezető irányításával készítettem, és csak a megadott forrásokat használtam fel. Minden olyan részt, melyet szó szerint, vagy azonos értelemben, de átfogalmazva más forrásból vettem, a forrás megadásával jelöltem.

.....
Molnár Balázs

Tartalomjegyzék

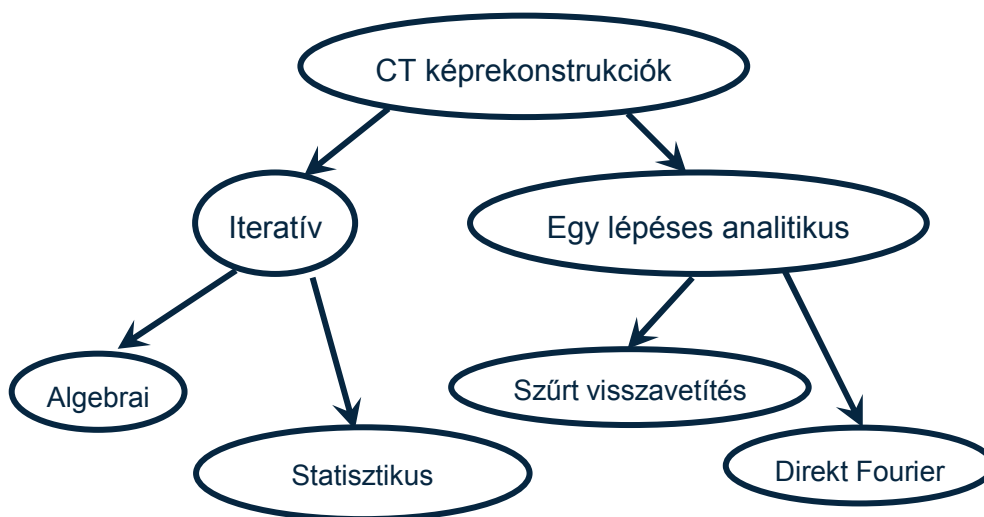
| | |
|--|-----------|
| 1. Bevezetés | 4 |
| 1.1 Általános bevezetés | 4 |
| 1.2 Transzmissziós ML algoritmus Poisson approximációval | 6 |
| 1.3 A megvalósított iterációs séma előállítás | 8 |
| 2. Előző munkák..... | 11 |
| 3. Az algoritmus implementációja | 12 |
| 3.1 Grafikus kártya | 12 |
| 3.2 A szimuláció geometriája, alkalmazott közelítések | 15 |
| 3.3 A program felépítése | 16 |
| 4. Az előrevetítés és visszavetítés függvényei..... | 17 |
| 5. Tesztelés matematikai fantomokon | 18 |
| 5.1 Az előrevetítés verifikációja..... | 18 |
| 5.2 Az L2 norma..... | 22 |
| 5.3 Tesztesetek | 23 |
| 5.3.1 Shepp-Logan fantom rekonstrukciója, L2 norma iterációs szám függése..... | 23 |
| 5.3.2 Összehasonlítás a CPU-n futó algoritmussal | 25 |
| 5.3.3 Konvergencia | 26 |
| 5.3.4 Paraméter optimalizáció, futásidők | 28 |
| 5.3.4 Homogén abszorpciójú fantom rekonstrukciója, középvonal mentén vett metszete | 33 |
| 6. Tesztek valós mérési eredményekkel | 35 |
| 6.1 Kísérleti elrendezés | 36 |
| 6.2 A valós mérési adatok feldolgozásának kihívásai..... | 36 |
| 6.3 Geometriai kalibráció | 38 |
| 6.4 Elvárások | 39 |
| 6.5 Tesztesetek | 40 |
| 7. Konklúzió, kitekintés | 42 |
| 8. Irodalomjegyzék..... | 43 |

1. Bevezetés

1.1 Általános bevezetés

Az orvosi képalkotás egyik alapvető technikája a transzmissziós tomográfia, aminek legismertebb, és leggyakoribb alkalmazása a komputált tomográfia (CT¹). Egy CT berendezés által elkészített felvételek (továbbiakban: projekciók) ideális esetben a röntgensugárással szemben mutatott teljes gyengítési együttható eloszlás Radon-transzformáltjai, amiből valamilyen inverziós módszerrel kaphatjuk meg az eredeti eloszlást. Ezt nevezzük képrekonstrukciónak. Az emberi szövetek gyengítési együtthatójának különbözősége kontrasztot jelent a rekonstruált képen, ez segítséget nyújthat a helyes diagnózis felállításához.

A rekonstrukciós módszereket alapvetően két csoportba sorolhatjuk: iteratív és analitikus. Az iteratív rekonstrukciókon belül léteznek algebrai és statisztikus módszerek (itt található az ML-EM is), az analitikuson belül pedig többek közt a szűrt visszavetítést találjuk. Áttekintést erről az 1. ábra, valamint [3] ad.



1. ábra
Képrekonstrukciós módszerek [3]

Az általánosan használt szűrt visszavetítés egy jól működő, bevált módszere a kép rekonstrukciójának, azonban számos hátránya van más, például iteratív, módszerekkel szemben. Alapvető feltételezése ugyanis a módszernek, hogy a detektor által begyűjtött

adatok a gyengítési együttható eloszlás radon-transzformáltjai, ami a valóságban tökéletlen leképezés miatt nem teljesül maradéktalanul [4]. A rekonstruált képen ez műtermékek formájában jelentkezik, szélsőséges esetekben, például fém jelenléte esetén, a szűrt visszavetítés nem is alkalmazható (legalábbis korrekcióra szorul) a spektrumfelkeményedés és szórás okozta fém-műtermékek [5] megjelenése miatt. Egy statisztikus, iteratív képrekonstrukciós eljárás kevésbé érzékeny a detektor által mért adatok statisztikus zajára, hiányos adatra, nagy gyengítési együttható inhomogenitás jelenlétére, így egyes esetekben alkalmazása előnyösebb lehet. A legnagyobb lehetőség abban rejlik, hogy az inverzió megvalósításakor a teljes fizikai valóság beépíthető a modellbe, a rekonstrukció során figyelembe vehetjük például a röntgen sugárzás spektrumát, a szórást, nem-szokványos geometriákat.

A képszintézis elvi lehetősége a röntgen sugárzás anyaggal való kölcsönhatásán alapszik, ezért fontos megemlíteni ezeket a folyamatokat. Az orvosi röntgen-diagnosztika esetében (sugárzás energiája, anyagi tulajdonságok) a legjelentősebb kölcsönhatások a rugalmas- és rugalmatlan szórás, valamint az abszorpció. A párkeltés jelensége nem játszik szerepet a CT vizsgálatok esetében, hiszen az alkalmazott röntgen sugárzás maximális energiája általában 100-200 keV. Első közelítésben érdemes úgy kezelni a problémát, hogy csak abszorpciót szimulálunk, ezt mutatja be a dolgozatban szereplő algoritmus is. Az ML-EM módszer ugyanis megköveteli a kölcsönhatás(ok) fizikai modellezését is. Ezt a modellt Monte-Carlo részecsketranszport-szimulációval építettem fel, aminek alkalmazása ma már igen elterjedt az orvos-fizikai problémák körében is. Az ún. visszavetítés algoritmus elemzése során gyakran fogok hivatkozni a Beer-Lambert törvényre, ugyanis a visszavetítő függvény lényegében egy azt felhasználó analitikus módszer. A Beer-Lambert törvényt az (1) alakban írhatjuk fel, ahol μ jelöli a gyengítési együtthatót ($\mu = \mu(E, Z)$ energia és rendszámfüggő), dl az infinitezimális elmozdulást, Γ az integrálandó tartományt (görbét), N a részecskék számát, N_0 pedig a kezdeti részecskék számát.

$$N = N_0 \cdot \exp\left(-\int_{\Gamma} \mu(\bar{r}) \overline{dr}\right) \quad (1)$$

A Beer-Lambert törvény homogén esetben, egyenes mentén vett sugárgyengülés esetén (2) alakú.

$$N = N_0 \cdot \exp(-\mu l) \quad (2)$$

1.2 Transzmissziós ML algoritmus Poisson approximációval

A dolgozatomban bemutatott rekonstrukció a Maximum Likelihood (továbbiakban ML) alapötletét használja fel. Tekintsük egy transzmissziós kísérletnél Y -t a mért beütések számának valószínűségi változó vektoraként (továbbiakban v). Jelölje μ a rekonstruálandó skaláreloszlást (gyengítési együttható), és vegyük figyelembe μ -t úgy, mint egy paramétert, ami mellett Y -t mérünk. Egy adott μ mellett mért Y v súlyfüggvénye legyen $g(y|\mu)$, aminek a maximumát keressük μ szerint, vagyis azt a skaláreloszlást, ami mellett y a legnagyobb valószínűséggel felelne meg a valódi mérésünk eredményének. Ez általában nem könnyű, ugyanis az Y adatok nem tartalmaznak elegendő információt a függvény maximalizálásához, hanem egy teljes adathalmaz vetületét adják csak. Ezt a teljes adathalmazt, amiből a rekonstrukciót maradéktalanul végre lehetne hajtani, jelölje X v , és legyen ennek súlyfüggvénye $f(x|\mu)$. Ha a mérendő térfogatot felosztjuk voxelekre¹, akkor a legkézenfekvőbb választás, ha X tartalmazza az egyes voxelekbe belépő részecskék számát. Ekkor ugyanis egy projekció esetén a j . voxel μ_j abszorpciós értékét egyértelműen meg lehet határozni az X_j illetve az X_{j+1} , vagyis a belépő és kilépő fotonok számának, segítségével. Egy LoR² által érintett voxelek száma legyen $m-1$, így igaz lesz az $Y = X_m$ egyenlőség (az utolsó voxel elhagyó részecskék száma X_m). Tekintsük most a

$$H(\mu | \mu_n) = E(\ln f(X | \mu) | Y, \mu_n) - \ln[g(Y | \mu)] \quad (3)$$

függvényt. Megmutatható a Jensen egyenlőtlenség alkalmazásával, hogy H maximuma μ szerint éppen μ_n -nél van, vagyis $\max_{\mu} \{H(\mu | \mu_n)\} = H(\mu_n | \mu_n)$. A bizonyítás részleteit mellőzve belátható (felírva a $H(\mu_n | \mu_n) - H(\mu | \mu_n)$ mennyiséget, és a logaritmus függvényre alkalmazva a Jensen egyenlőtlenséget), hogy ha úgy választjuk meg μ_n -t, hogy $f(x | \mu_n) > f(x | \mu)$, akkor H -nak valóban $\mu = \mu_n$ -nél van maximuma. Állítsunk elő tehát egy $\{\mu_n\}$ sorozatot, és minden lépésben maximalizáljuk $E(\ln f(X | \mu_{n+1}) | Y, \mu_n)$ értékét μ_{n+1} szerint. Ekkor (3)-at tekintve kimozdultunk $H(\mu_n | \mu_n)$ -ből $H(\mu_{n+1} | \mu_n)$ -be, így az előzőek alapján H kisebb lett. Ez maga után vonja azt, hogy $\ln[g(Y | \mu_{n+1})] > \ln[g(Y | \mu_n)]$, vagyis megvalósítottuk a Maximum Likelihood módszert. Az EM módszer összességében

¹ Háromdimenziós pixel

² Line of Response - válaszegyenes

visszavezeti a likelihood becslést várható érték maximalizálásra, ahol a teljes adathalmaz Y melletti várható értékét kell optimalizálni a gyengítési együttható eloszlás szerint.

Egyetlen dolog hiányzik még a fenti módszer konvergenciáját tekintve: még nem mondtuk meg, hogy a log-likelihood függvénynek milyen szélsőértéke van. Természetesen azt szeretnénk, ha maximuma lenne, vagyis $\ln[g(Y|\mu)]$ μ -ben konkáv lenne. Ez pontosan akkor teljesül, ha második deriváltmátrixa negatív definit. A következőkben beláthatjuk, mikor is teljesül ez a szimulációkban.

Jelölje most Y_i az i . projekciónál a detektort elérő fotonok számának vv-ját. Modellezzük Y_i -t Poisson eloszlású vv-ként³, vagyis eloszlásfüggvénye legyen

$$g(y_i|\mu) = \frac{\lambda_i^{y_i}(\mu) \cdot e^{-\lambda_i(\mu)}}{y_i!} \quad (4)$$

egy adott μ paraméter mellett. Az Y_i valószínűségi változó λ_i várható értéke felírható a Beer-Lambert törvény alapján:

$$\lambda_i(\mu) = b_i \exp\left(-\sum_{j \in L_i} l_{ij} \mu_j\right) \quad (5)$$

ahol b_i az i . projekciónál a forrást elhagyó részecskék száma, l_{ij} pedig az i . projekcióban a j . voxelben a foton által befutott út hossza. Az összes projekcióra felírható Y együttes eloszlásfüggvény a $g(y_i|\mu)$ függvények szorzata lesz, hiszen az Y_i -k egymástól független vv-k. Ebből felírhatjuk a log-likelihood függvényt a következő alakban:

$$\begin{aligned} \ln[g(y|\mu)] &= \ln\left[\prod_i g(y_i|\mu)\right] = \sum_i \ln[g(y_i|\mu)] = \\ &= \sum_i \left\{ -b_i \exp\left(-\sum_{j \in L_i} l_{ij} \mu_j\right) - y_i \sum_{j \in L_i} l_{ij} \mu_j + y_i \ln b_i - \ln y_i! \right\} \end{aligned} \quad (6)$$

A konvexitás vizsgálatához venni kell a fenti függvény kétszeres parciális deriváltjait:

$$H_{kl} = \frac{\partial^2}{\partial \mu_k \partial \mu_l} \ln[g(y, \mu)] = -\sum_i a_{ik} b_i \exp\left(-\sum_{j \in L_i} l_{ij} \mu_j\right) a_{il} \quad (7)$$

ahol a_{ik} együtthatók az A mátrix elemei. A egy olyan mátrix, aminek sorai az egyes projekciókat indexelik, oszlopai pedig a voxeleket. A a_{ik} eleme azon úthossz, amit az i . projekcióban a k . voxelben befut a részecske, vagyis

³ Arról, hogy ez miért megfelelő feltételezés az esetünkben, és milyen feltételekkel, még lesz szó a későbbiekben

$$a_{ik} = \begin{cases} l_{ik} \leftrightarrow k \in L_i \\ 0 \leftrightarrow k \notin L_i \end{cases} \quad (8)$$

Ahhoz, hogy $\ln(g)$ konkáv legyen, H -nak negatív definitnek kell lennie, vagyis minden v vektorra: $vHv < 0$ kell, hogy teljesüljön. Ez pontosan akkor áll fenn, ha az A mátrix teljes rangú $n \times m$ -es mátrix, ahol $n \geq m$. Tehát legalább annyi projekciót kell végeznünk, ahány voxelből áll a rekonstruálandó skaláreloszlás által elfoglalt térfogat.

A következőkben tegyük fel, hogy a konkávság feltétele teljesül (vagyis az ML-EM konvergens), és nézzük meg, hogyan lehet az iterációt előállítani.

1.3 A megvalósított iterációs séma előállítása

Az előzőekben már felírtuk a log-likelihood függvényt (6) alakban. Ugyanígy a teljes adathalmazra (X) is megtehetjük a log-likelihood függvény felírását. Észre kell vennünk ehhez, hogy a $j+1$. voxelbe jutó részecskék binomiális eloszlást követnek (feltéve, hogy a j . voxelbe beérkező részecskék száma fix, nem véletlen), vagyis $X_{j+1} | X_j \sim Binom(n, p)$, ahol a próbák száma $n = X_j$, valamint a siker valószínűsége $p = \exp(-\mu_j l_j)$. Tételezzük fel azt is, hogy az X_j , csakúgy, mint a detektort ért beütések száma, Poisson eloszlást követ λ_j paraméterrel ($X(j) \sim Poi(\lambda_j)$). Az egyszerűség kedvéért bevezettük a $\lambda_r = b \exp\left(-\sum_{k=1}^{r-1} \mu_k l_k\right)$ jelölést. Így a teljes adathalmaz log-likelihood függvénye:

$$\begin{aligned} \ln[f(x | \mu)] = & \sum_i -b_i + X_1 \ln b_i - \ln X_1! + \sum_{j=1}^{m-1} \ln \binom{X_j}{X_{j+1}} + \\ & + X_{j+1} \ln(\exp(-l_{ij} \mu_j)) + (X_j - X_{j+1}) \ln(1 - \exp(-l_{ij} \mu_j)) \end{aligned} \quad (9)$$

Láthatjuk, hogy ha ennek a mennyiségnek a várható értékét akarjuk maximalizálni X_m feltétel mellett, akkor még ki kell találnunk az $E(X_j | X_m)$ várható értéket.

Most fontos megemlíteni, hogy mikor is jó az a feltételezés, hogy mind X_j , mind $Y = X_m$ (a detektálható, vagyis az utolsó voxel elhagyó részecskék száma) Poisson eloszlást követ. Mint láttuk a voxel elhagyó részecskék a bemenő részecskék számától függő binomiálisak, az exponenciális gyengülés szerinti valószínűséggel. Ez több voxelen való áthaladásra is igaz,

a valószínűség változik meg: $p = \exp\left(-\sum_k \mu_k l_k\right)$ A binomiális eloszlás Poisson-

approximációját alkalmazhatjuk erre az esetre, ha a binomiális eloszlás n és p paraméterére: $n \rightarrow \infty$ ugyanakkor $p \rightarrow 0$, de úgy, hogy $np \rightarrow \lambda < \infty$. Ezt úgy teljesítjük, (visszanézve a fenti n és p paraméterekre) hogy sok fotont indítunk a szimulációban, valamint kellően abszorbens és/vagy nagyméretű közeget használunk (ezt a valóságban nem túl nehéz teljesíteni).

Ezek után kanyarodjunk vissza az eredeti célunk felé, ami szerint az $E(X_j | X_m)$ feltételes várható értékről szeretnénk információt kapni. X_m binomiális eloszlású, feltéve X_j -t,

a korábbi jelölésekkel $X_m | X_j \sim Binom\left(X_j, \frac{\lambda_m}{\lambda_j}\right)$. A Bayes-tétel segítségével a fordított

feltételt is kiszámíthatjuk, vagyis a kérdéses $P(X_j | X_m) = \frac{P(X_m | X_j) \cdot P(X_m)}{P(X_j)}$ kiszámítható,

hiszen mindegyikről tudjuk már, hogy milyen eloszlású (az egyenlet jobb oldala ismert súlyfüggvényekből áll). A részletes levezetés [1]-ben olvasható. Arra a következtetésre jutunk, hogy $X_j - X_m | X_m \sim Poi(\lambda_j - \lambda_m)$, amiből egyből következik, hogy $E(X_j - X_m | X_m) = \lambda_j - \lambda_m = E(X_j) - E(X_m)$, s ekképp:

$$E(X_j | X_m) = X_m + E(X_j) - E(X_m) \quad (10)$$

ami pontosan a maximalizálandó várható érték számolásához kell.

Most már elkezdhetjük a tényleges maximalizációt, vagyis deriváljuk le az $E(\ln f(X | \mu) | X_m, \mu)$ -t μ szerint, tegyük egyenlővé nullával. $\ln(f)$ adott a (9) formulában, a feltételes várható értékeket pedig éppen az előbb írtuk fel (10). A deriválandó függvény tehát:

$$\sum_{j \in L_i} \left\{ E(X_{j+1} | X_m) \ln(\exp(-l_{ij}\mu_j)) + (E(X_j | X_m) - E(X_{j+1} | X_m)) \ln(1 - \exp(-l_{ij}\mu_j)) \right\} + R \quad (11)$$

ahol R tartalmazza az összes olyan paramétert, ami nem függ μ -tól. A szélsőérték keresést elvégezzük:

$$\begin{aligned} 0 &= -\sum_i E(X_{j+1} | X_m) l_{ij} + \sum_i (E(X_j | X_m) - E(X_{j+1} | X_m)) \frac{l_{ij} \exp(-l_{ij}\mu_j)}{1 - \exp(-l_{ij}\mu_j)} \\ &= \sum_i -E(X_{j+1} | X_m) l_{ij} + \sum_i (E(X_j | X_m) - E(X_{j+1} | X_m)) \frac{l_{ij}}{\exp(l_{ij}\mu_j) - 1} \end{aligned}$$

Ez a transcendens egyenlet numerikusan oldható meg, ha $s = l_{ij}\mu_j \ll 1$:

$$\frac{1}{e^s - 1} = \frac{1}{s} - \frac{1}{2} + \frac{s}{12} + O(s^3) \quad (12)$$

(12) elsőrendű közelítését alkalmazva az iterációs képlet a (13) alakot ölti.

$$\mu_j^{n+1} = \frac{\sum_i (E(X_j | X_m) - E(X_{j+1} | X_m))}{\frac{1}{2} \sum_i (E(X_j | X_m) + E(X_{j+1} | X_m))} \cdot l_{ij} \quad (13)$$

Behelyettesítve a feltételes várható értékeket (10)-ből, észrevehetjük, hogy a számlálóban csak a j . és a $j+1$. voxelre eső fotonok várható értéke marad, a nevezőben viszont megjelenik a detektort ért beütések száma, valamint ennek várható értéke is.

$$\mu_j^{n+1} = \frac{\sum_i (E(X_j) - E(X_{j+1}))}{\frac{1}{2} \sum_i (E(X_j) + E(X_{j+1}) + 2X_m - 2E(X_m))} \cdot l_{ij} \quad (14)$$

A könnyebb érthetőség kedvéért alkalmazzunk más jelöléseket, amik már alkalmazkodnak az implementációhoz. (14)-ben mindkét összegzést az i -re hajtjuk végre, ami nem más, mint a LoR-okat számozó futóindex. Jelöljük most ezt úgy, hogy összegzünk minden olyan LoR-ra, ami átmegy a j . voxelen, aminek gyengítési együtthatóját épp rekonstruálni akarjuk (jelölés: $lor \in L_j$). Ezentúl a (14)-beli várható értékeket szimulálni fogjuk, a szimulációnkban ezek tényleges fotonszámok lesznek, vagyis egy részecsketranszport-szimulációval megbecsüljük, hogy hány db részecske esett azon voxelre. Ezeket a becsléseket a továbbiakban $X_{lor}[j]$ alakban reprezentálom, valamint az utolsó m . voxelt elhagyó részecskeszám esetén visszatérek az Y_{lor} jelölésre. Az i . LoR j . voxeljében befutott úthosszt pedig $L_{lor}[j]$ -vel fogom jelölni. Tekintsük tehát (15)-öt, ami megegyezik az implementált iteráció sémával.

$$\mu[j] = \frac{\sum_{lor \in L_j} (X_{lor}[j] - X_{lor}[j+1])}{\sum_{lor \in L_j} \left(Y_{lor} - \hat{Y}_{lor} + \frac{X_{lor}[j] + X_{lor}[j+1]}{2} \right)} \cdot L_{lor}[j] \quad (15)$$

(15) azt fogalmazza meg a matematika nyelvén, hogy a soron következő iterációban a j . voxel gyengítési együtthatóját úgy számolhatjuk, hogy a rajta átmenő összes válaszgyenest (LoR-t) tekintve összegezzük a benne elnyelődött fotonok számát (ez maga a számláló), majd elosztjuk egy olyan, szintén LoR-okra vonatkozó, összeggel, amiben a befutott úthosszakkal súlyozzuk a voxelbe bemenő, és abból kimenő részecskék számának átlagát, valamint a mért és a szimulált detektor-beütésszámokat. Hiszen a (14)-ben X_m éppen a detektort ért beütésszám vv-ja, valóságban ennek helyére X_m egy realizációja kerül, vagyis a mérési

eredményünk: Y_{lor} (a lor által kijelölt detektorpixel beütésszáma). $E(X_m)$ pedig mérési eredmények várható értéke, amit itt, (15)-ben, \hat{Y}_{lor} -ral jelöltem.

2. Előző munkák

A TDK-dolgozatban bemutatott eredményeket érdemes összehasonlítani hasonló, statisztikus iteratív képrekonstrukciókkal, valamint a korábban elért eredményekkel. Annak céljából, hogy ezen dolgozat olvasásakor ne vesszen el a lényeges információ a részletek közt, nem fogom részletesen megtenni ezt az összehasonlítást, azonban helyenként hivatkozni fogok az előző munkáimra, ebben a fejezetben pedig röviden összefoglalom ezeket.

A Bsc szakdolgozatomban [6] egy hasonló, iteratív képrekonstrukciós eljárást teszteltem, ami szintén az ML becslésből indult ki, azonban egy másik közelítő módszerrel, a teljes adathalmaz bevezetése nélkül, csupán a mérési eredmények, és a befutott úthosszak becsléséből számította ki az iteráció következő elemét. A rekonstrukciós módszerre az irodalomban gyakran ML-TR néven hivatkoznak. [1] szerint az EM algoritmus, és a segítségével előállított formula sokkal pozitívabb kilátásokkal kecsegtet, főleg a konvergencia, és a futási idő szempontjából.

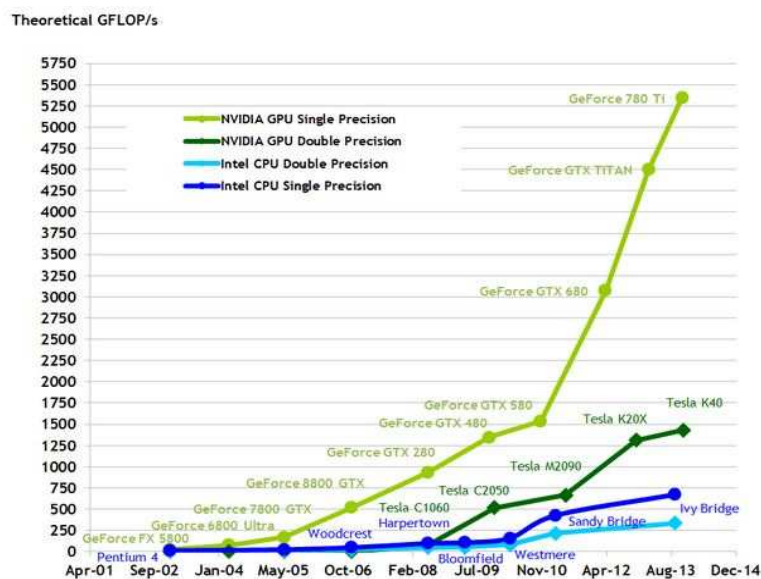
Gyakorlatilag közvetlen előzménye ennek a dolgozatnak az előző évi TDK-munkám [2], amiben háromféle implementációját vázoltam fel az ML-EM algoritmusnak, de ezeket csak CPU-n teszteltem. Azt a következtetést vontam le a vizsgálatuk során, hogy GPU-ra a [2]-ben „v3”-mal jelölt implementáció a legalkalmasabb, ezért ennek fejlesztését végeztem el. Az 5.3.2 fejezetben be is mutatom az ezáltal elért eredményeket. A TDK-munkám új eredményei közé tartozik a teljes GPU-s implementáció, egy merőben más struktúrájú szoftver létrehozása. Ennek segítségével számos lehetőség tárul elénk, ugyanis a rekonstrukció sokkal gyorsabb, hatékonyabb lett, így tesztelését végre lehet hajtani más rekonstrukciós algoritmusokkal összevetve. Másrészt hatalmasat lépett előre ezzel a fejlesztés abban a tekintetben is, hogy valós mérési adatok rekonstrukcióját is létre lehet hozni segítségével, innentől pedig nyitva állnak a rekonstrukciós módszer kihasználásának lehetőségei. Mivel az ML-EM algoritmus GPU-n történő megvalósítása nem triviális programozási feladat, ezért a következőkben összefoglalom a grafikus kártyák szerepét az implementációban, ami talán elősegíti ezen előrelépés mikéntjének megértését.

3. Az algoritmus implementációja

3.1 Grafikus kártya

Általánosan egy iteratív képrekonstrukciós megoldást választva a jobb képminőség elérésének reményében lemondunk a kis számításigényről, vagyis hatalmas mennyiségű műveletet végzünk annak érdekében, hogy az elért eredményeink precízebbek legyenek. Szerencsére, a párhuzamos feldolgozás adta lehetőségeket kihasználva csökkenteni tudjuk rekonstrukció (azonos körülmények között nagyságrendekkel hosszabb) futási idejét. Ahogyan azt [2]-ben is láthatjuk, az 1.3 fejezetben ismertetett (statisztikus, iteratív) ML-EM rekonstrukció CPU-n való futása igen lassú tud lenni. Célszerű kihasználni a grafikus kártyákban rejlő lehetőségeket az algoritmus gyorsítására. Mivel a NTI-ben lehetőségem van egy Nvidia GTX 690-es grafikus kártyát programozni, ezért az algoritmus implementációjának programozási nyelvét CUDA-nak választottam. Ez az Nvidia saját programozási nyelve, alapvetően C/C++ alapú, kiegészítő parancsokkal, GPU specifikus memória fajtákkal, változókkal [7].

A grafikus kártyák kihasználása a párhuzamosan végrehajtható műveletek elvégzésére az utóbbi évtizedekben igen gyakorivá vált, mivel a jelenleg releváns programozási feladatok számításigényét már nem elégíti ki a hagyományos processzorok fejlődése (Moore-törvény). A grafikus kártyák és a processzorok fejlődését jól szemlélteti a 2. ábra.



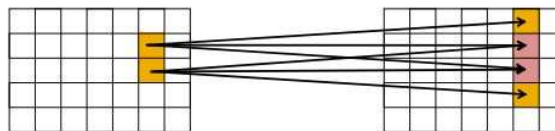
2. ábra

Az Nvidia kártyák számításkapacitása az Intel processzorokkal szemben [7]

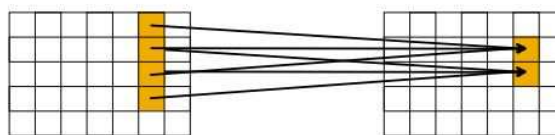
Fontos kiemelni, hogy bár a számítási kapacitás (GFLOP/s) általában több nagyságrenddel nagyobb a GPU-k esetében, mint a CPU-knál, ez természetesen hátrányokkal is együtt jár, amik általában az algoritmus struktúrájából fakadnak.

Az orvosi képalkotásban gyakran előfordul, hogy a feldolgozás/szimuláció sematikus szóró jellegű (ld 3. ábra), vagyis a kiinduló adatból az elvégzendő számítás több memóriahelyre tud (rész)eredményt biztosítani. Ez azt jelenti, hogy a párhuzamosan futó operációk között versenyhelyzet (race condition) alakulhat ki, aminek nem megfelelő kezelése információvesztést okoz. A grafikus kártyákon használható atomi műveletek erre megoldást nyújtanak, amik bár lassabbak, mint a hagyományos műveletek, de a párhuzamosan futó szálak között így nem alakul ki versengés.

Szórás (Scatter)



Gyűjtés (Gather)

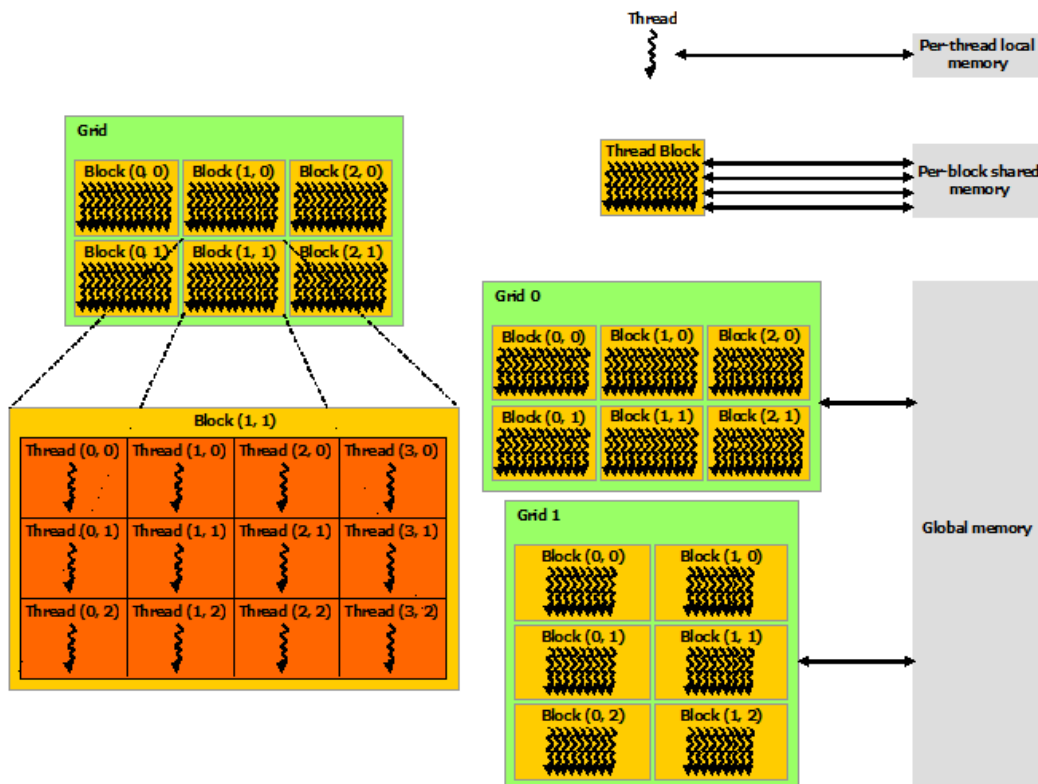


3. ábra
Párhuzamos sémák

A megoldandó probléma komplexitása szintén növeli a GPU-n való futás idejét, ezért alapvető követelmény a hosszú ciklusok elkerülése, valamint sok elágazás (if-else) használata a programozás során. A képrekonstrukciónál, és hasonló alkalmazásoknál, érdemes definiálni az aritmetikus intenzitás fogalmát (arithmetic intensity), ami annak mérőszáma, hogy milyen arányban vannak a forráskódban az aritmetikai műveletek a memória-elérési parancsokhoz képest. Minden GPU-s programozási feladatnál célszerű ésszben tartani, hogy a memória-hívások igen költségesek, főleg ezért is érvényes az, hogy az inherensen párhuzamos jellegű feladatok futása sokkal gyorsabb, mint a komplexebb problémáké. A GPU-ról, valamint annak orvosi képalkotásban való felhasználásáról nagyszerű cikket olvashatunk, és további hasznos információhoz juthatunk [8] megtekintésével.

A 4. ábra mutatja be, hogy a grafikus kártyához tartozó memória kezelése miként történik a CUDA architektúrán. Az egyes részfeladatokhoz (amiket párhuzamosan kívánunk

elvégezni) a CUDA memóriaszálakat (továbbiakban thread-ek) rendel, amiket alapvetően két nagyobb struktúrába ágyaz. A thread-ek blokkokban (továbbiakban block) helyezkednek el, amik pedig egy rácsra (továbbiakban grid) vannak felhelyezve. A végrehajtás során megadhatjuk a block-ok által tartalmazott thread-ek számát, valamint azt, hogy egy grid hány block-ot tartalmazzon. Minden thread-nek van egy azonosítója (egy GPU specifikus változó), aminek használatával kioszthatjuk a párhuzamosan elvégzendő számításokat. A thread-ek rendelkeznek egy lokális memóriával, ami cache-elt, vagyis lényegesen gyorsabb elérésű, mint a globális memória. A block-ok is rendelkeznek megosztott (shared) memóriával, ami azonban lassabb, így a szálak közti kommunikáció kifejezetten hátrányos a futási időre nézve. Tehát ha olyan implementációt tudunk előállítani, amiben az egyes memóriaszálak egymástól függetlenül dolgozhatnak, és a memóriát megfelelően strukturáljuk (threads/block, blocks/grid), akkor elérhetjük a párhuzamosítás maximális kihasználtságát. Ez a határ nyilván grafikus kártya-specifikus, ezért érdemes ismerni a kártya tulajdonságait, mielőtt elindítjuk a programunkat.



4. ábra
Memóriakezelés CUDA-ban [7]

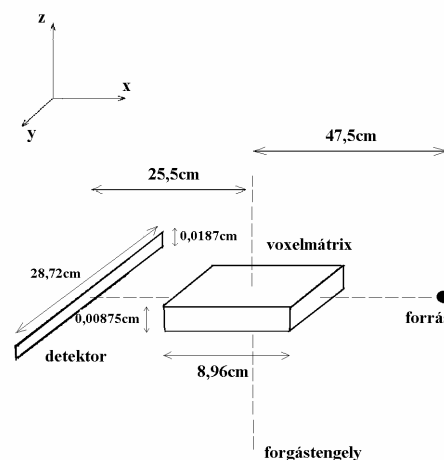
A TDK-dolgozatban bemutatott képrekonstrukciót alapvetően két nagyobb programozási feladatra bonthatjuk: előrevetítés és visszavetítés. E kettőhöz egy-egy ún.

kernel-függvény tartozik, ami szintén egy GPU specifikus, lényegében azt jelenti, hogy meghívásakor a GPU számításkapacitásának felhasználásával egyidőben végrehajtódik N db művelet (amit maga a kernel-függvény definiál), ahol N-et meghatározza a grafikus kártya paraméterei (például az ún. warp size), valamint a memória kiosztása (block-ok, grid-ek).

3.2 A szimuláció geometriája, alkalmazott közelítések

Az implementáció részletezése előtt szükséges a könnyebb érthetőség kedvéért tisztázni a geometriát, ami a szimuláció virtuális világát jellemzi. Mind az előre-, mind a visszavetítésben az 5. ábrán látható elrendezést veszem figyelembe. A fejlesztés jelenlegi stádiumában az algoritmus egyszerűsített formában működik a következő szempontokból:

- 1) Kétdimenziós, de kúpsugaras (cone-beam) geometriát valósít meg: a szimulációban a forrásból indított fotonok térbeli eloszlása egyenletes, de csak a voxelmátrix irányába enged részecskéket, aminek kiterjedése függőleges irányban egy voxelnyi, csakúgy, mint ahogy a detektor is egy pixel magas
- 2) A voxelmátrixon kívül a rekonstrukció vákuumot feltételez, itt nincs abszorpció
- 3) A rekonstruálandó térfogat vízzel van feltöltve, gyengítési együtthatójának anyagfüggése ennek megfelelő
- 4) Monokromatikus nyalábbal dolgozik, a matematikai fantomokon végzett teszteknel 100keV-os fotonokat szimulál (5. fejezet), a valós mérési adatok rekonstrukciójánál a megfelelő effektív energiát (ld. 6. fejezet)
- 5) Nem szimulál szórást, a fotonnal csak abszorpció történhet



5. ábra
A tesztelés geometriája

3.3 A program felépítése

Az alábbiakban az algoritmus felépítését fogom ismertetni általánosságban, csak főbb elemeit említve meg. A matematikai fantomok rekonstrukciójára fejlesztett forráskód és a valós mérési eredmények feldolgozására dedikált program lépései néhol eltérnek egymástól, ezeket ismertetni fogom.

Elsőként megtörténik az eredeti (később rekonstruálandó) kétdimenziós kép bevitele. Az eredeti képen az algoritmus mérési szimulációt végez monokróm nyalábbal, vagyis végrehajtja az előrevetítést, aminek eredménye a (15)-ben Y -nal jelölt beütésszám. A program a fejlesztés jelen fázisában nem szimulál szórást, a fotonnal csak abszorpció történhet. Az előrevetítést Monte-Carlo részecsketranszport módszerrel valósítja meg, a szabad úthosszt a Woodcock-módszerrel mintavételezve. A valós mért adatok rekonstrukciója esetén a mérésből származó szinogram beolvasása, és a detektor által biztosított beütésszámok részecskeszámokká alakítása történik meg először (bővebben ld. 6.5 fejezet).

Az algoritmus ezután elindítja az iterációs ciklust, amely során végrehajtódik a rekonstrukció. Az iterációkban még egy ciklus van beágyazva, ami a projekciókat lépteti. A fejlesztés során azt tapasztaltam, hogy az előrevetítést bár egyszerre is el lehet végezni az összes projekcióra, mégis érdekesebb projekciónként egy előrevetítést és utána közvetlenül visszavetítést elvégezni, hiszen ekkor nem szükséges tárolni csak az aktuális projekcióra vonatkozó eredményeket.

A visszavetítés-függvény végzi el az X , L változók becslését, vagyis a voxelekbe bejutott, és azokból kimenő részecskék számát, illetve a voxelekben befutott úthosszakat számítja ki. A visszavetítés struktúrája 3.1 fejezetben elmondottakhoz viszonyítva némileg ambivalens, mivel nem voxel-vezérelt (ez egy „gather” típusú, vagyis gyűjtő séma lenne), hanem LoR-vezérelt (szórás séma). Ez azt jelenti, hogy a visszavetítés során több LoR-hoz tartozó (párhuzamosan futó) memóriaszál is hozzájárul ez eredményekhez, és esetleg ugyanazon voxel memóriájához kíván írni. Ezért a visszavetítésben atomi műveleteket alkalmazok. Intuíció alapján a voxel-vezérelt (voxel-driven) visszavetítés gyorsabbnak tűnhet, de [3] szerint a LoR vezérelt (ray-driven) algoritmus jobb tulajdonságokkal bír, ezért is alkalmaztam inkább az utóbbi megoldást, ami könnyebben is implementálható.

4. Az előrevetítés és visszavetítés függvényei

Az előrevetítés elvégzéséhez Monte-Carlo részecsketranszportot alkalmaztam, a szabad úthosszat Woodcock-módszerrel mintavételezve. A hozzá tartozó kernel függvény először kiszámítja, hogy melyik detektorpixelbe ütközik majd a foton, és hogy hol kerül be a rendszermátrixba, majd végiglépteti a részecskét a rendszeren, nyilvántartva a pozícióját, és a súlyát, ami jelenleg még bináris változó, 0, ha elnyelődött, 1, ha nem. A GPU-memória kiosztása értelemszerűen történt: egy thread (memóriaszál) felelős egyetlen fotoncsomag szimulációjáért, a memóriaszálak között információ-áramlás nincs, így a párhuzamosítás nagy hatáskkal végezhető. Az előrevetítő kernel-függvény meghívódik minden egyes projekcióban. Olyan lehetőséget is vizsgáltam, ami ebben különbözik, vagyis az előrevetítést egyszerre végzi az összes projekcióra, azonban ekkor a memória-korlátok erősen befolyásolták a rekonstrukció minőségét. Gondoljunk bele abba, hogy már kétdimenziós esetben is mekkora memóriát igényel a rekonstrukció. A detektor felbontása, ahogy a 6. fejezetben is látni fogjuk, egyik irányban 1536 pixel. Ha legalább a detektorpixelek felének beütésszámát néhány százalékos relatív hibával akarjuk becsülni, akkor a Poisson-statisztika szerint átlagosan legalább néhány ezer foton kell, hogy egy pixelt érjen. Tegyük fel például, hogy $N=2500$ db részecske éri el a detektor egyes pixeleit átlagosan (ennek eléréséhez a forrásból már eleve többet kell indítanunk!), ekkor a relatív hiba $1/\sqrt{N} = 2\%$. Így egy projekcióban legalább $1536/2 \cdot 2500 \approx 2 \cdot 10^6$ db részecskét kell szimulálnunk ahhoz, hogy a látómezőnk felét kihasználjuk. Ez 360 projekció felvétele esetén már több százmillió db, ami memóriában (4 byte egy integer változó) máris több Gb-ot jelent. Ennyi memória általában nem áll rendelkezésre, illetve háromdimenziós esetben a voxelmátrix is Gb-os nagyságrendű helyet foglal, tehát itt ezt a megoldást el kell vetni. Ezért alkalmaztam a projekciónkénti előre-majd visszavetítési struktúrát, ekképp ugyanis az előző projekció előrevetítésének eredményére már nincs szükség.

Miután a mérési szimuláció eredményei a kezünkben vannak, megpróbálunk következtetni a kérdéses μ eloszlásra, vagyis visszavetítést végzünk. A megvalósítási lehetőségek struktúráját [2]-ben ismertettem, most a GPU-s implementációra legalkalmasabb módszert foglalom össze. A (15) formula szerint a visszavetítésben becsülni az egyes voxelekben befutott úthosszakat (L), valamint a voxelekbe jutott részecskék számát (X). Egyszerű, gyors, és hatékony becslést kapunk a sugárkövetés módszerét alkalmazva, vagyis ha az L -et a voxelmátrixon való végiglépegetéssel becsüljük, X -et pedig eközben analitikusan

a Beer-Lambert törvény alapján számoljuk. Választanunk kell egy lépésközt, valamint meg kell adjuk a LoR irányát, és ekkor a LoR és a voxelmátrix két dőféspontja (bemenet és kimenet) között ezzel a lépésközzel végigléptethetjük a virtuális részecskéinket. Eközben minden voxelre regisztráljuk a bennük elkövetett lépések számát, ebből számoljuk az úthosszakat. A visszavetítést LoR-vezérelt struktúrával valósítottam meg, vagyis egy memóriaszál egy válaszegyenes mentén végezte el a kérdéses mennyiségek számítását. Érdekes problémákat vet fel (ld. 6.1 fejezet), hogy a visszavetítésben ismerni kell a forrásból indult részecskék számát, vagy legalábbis meg kell becsülni egy fotonfluxussal arányos mennyiséget. Ugyanis a sugárkövetés (ray-tracing) során akkor tudjuk megbecsülni a voxelbe jutó részecskék számát, ha ismerjük, hogy mennyi indult azon LoR irányába. A futásidővel kapcsolatban a visszavetítés másképp reagál a voxelmátrix méretének változására, mint az előrevetítés. Míg az előrevetítés számításigénye lineárisan, vagy három dimenzióban maximum négyzetesen függ a voxelmátrix (lineáris) méretétől (hány voxelből áll egy oldala), a visszavetítés időigénye ennél magasabb hatvány szerinti függést mutat. Hiszen ugyanúgy, mint az előrevetítésnél három dimenzióban, a visszavetítő LoR-ok száma a rendszer lineáris méretével négyzetesen növekszik, de ezen visszavetítéseken belül az érintett voxelek száma is köbösen növekszik, így minden memóriaszálnak a műveletek sokszorososa jut a rendszer felbontásának növekedésével.

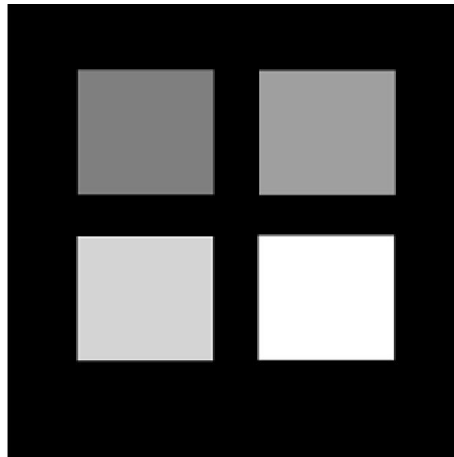
5. Tesztelés matematikai fantomokon

A rekonstrukció minőségének megállapítására egy hasznos módszer a matematikai fantomokon való tesztelés. Ennek lehetőségét az adja, ha egy digitálisan előállított fantom esetén tudjuk a mérést szimulálni. Mivel az algoritmus eleve felhasználja az iteráció során a mérési eredmények szimulációját, ez nem jelent külön problémát, egyszerűen el kell végezni egy előrevetítést az eredeti fantomra, majd ezt a szinogramot kell felhasználni a rekonstrukció során.

5.1 Az előrevetítés verifikációja

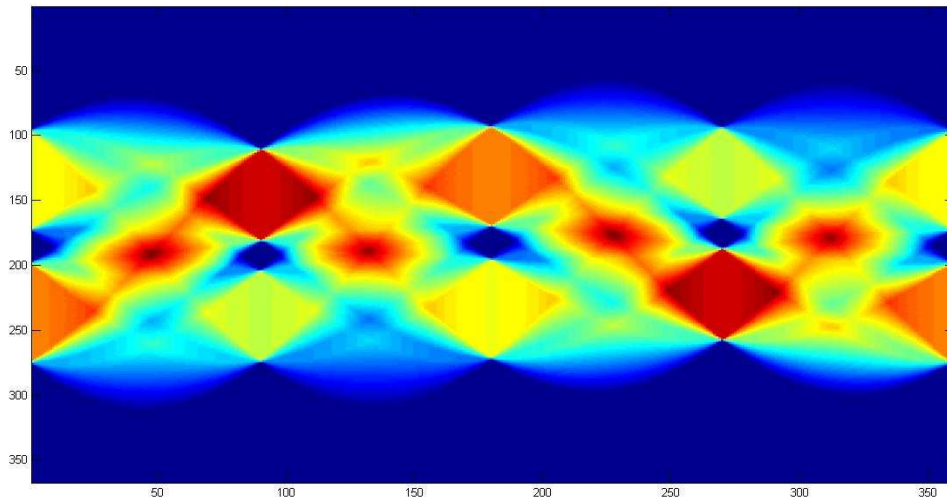
Alapvető elvárás a matematikai fantomokon elvégzett tesztek során, hogy a kiindulási alap, ami az iteráció „hajtóerejét” adja lehetőleg minél kevesebb zajjal legyen terhelt. Vagyis

a mérés szimulációjából származó szinogram lehetőleg közel tökéletesen egyezzen egy ideális mérés eredményével, vagyis épp a skaláreloszlásunk Radon-transzformáltja legyen. Ennek ellenőrzésére felhasználtam a Matlab *radon* nevű függvényét, amivel megkaptam a mesterségesen előállított fantom Radon-transzformáltját. Ezt vettem össze a saját szimulációm eredményével. Ez a gyakorlatban azt jelentette, hogy az előrevetítést elvégeztem az eredeti fantomon, összeszámoltam az egyes detektorpixelekre eső fotonok számát (a (15) képletben az \hat{Y} -al jelölt mennyiség), majd az ezen pixel irányába indított részecskék számával leosztva vettem a hányados logaritmusát. Ez természetesen nem más, mint a gyengítési együttható eloszlás vonalintegráljának közelítése, a forrást a detektorpixellel összekötő LoR mentén (Beer-Lambert törvény). Ekképpen az előrevetítés egy becslést ad a kép szinogramjára, amit össze lehet hasonlítani a Matlab által készített szinogramokkal⁴. Ezt megtettem különböző fantomokra (6. és 10. ábra), az eredmények a 7.-9., 11.-13. ábrákon láthatók. Azt tapasztaljuk, hogy kellően sok szimulált részecske esetén ($\sim 10^9$), a szinogramok szinte tökéletesen megegyeznek. Külön feltüntettem a 9. és 13. ábrákon a Matlab Radon-transzformáltja, és a saját előrevetítés által szimulált szinogramok különbségét. Ezen észrevehetjük, hogy az előrevetítés nem tökéletes, tartalmaz egy offset-hibát, ami a szinogram-különbségeken vízszintes csikokként figyelhetők meg.

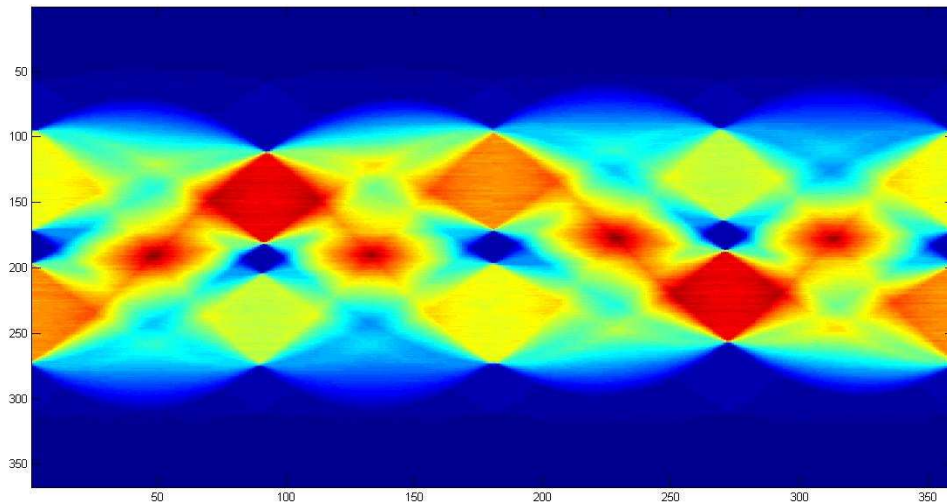


6. ábra
Az előrevetítés verifikációjára előállított matematikai fantom

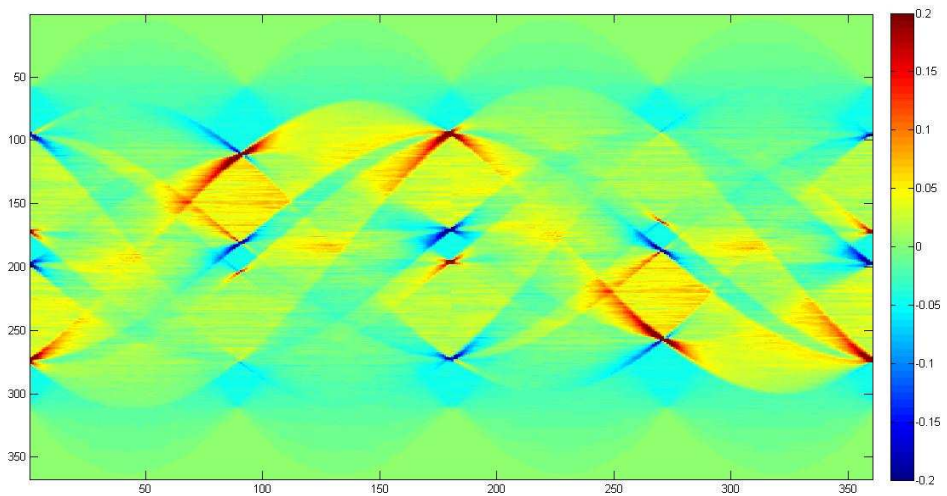
⁴ A fan-beam geometria miatt a szimuláció szinogramja eltér a Matlab paralell-beam geometriára értelmezett Radon-transzformáltjától, így a `fan2para` parancsot kell alkalmazni a szimulált szinogramra



7. ábra
A Matlab Radon-transzformáltja a 6. ábráról



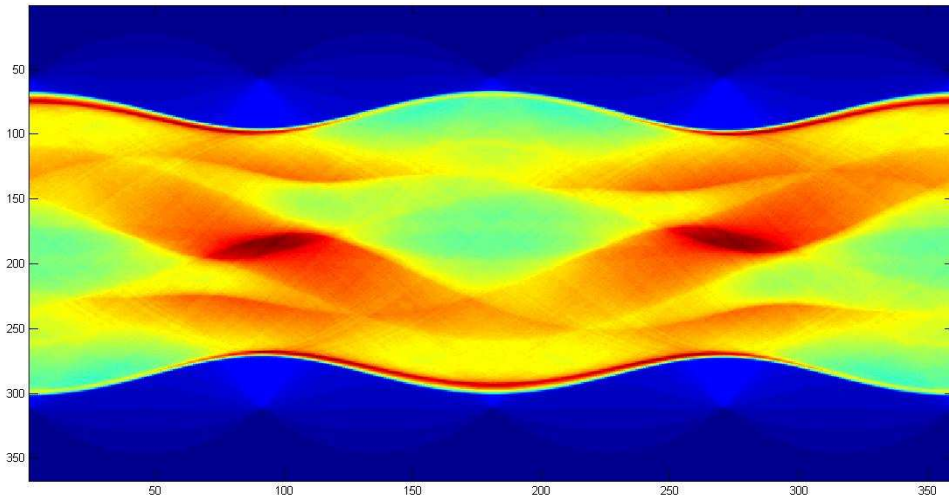
8. ábra
Saját előrevetítésem eredménye



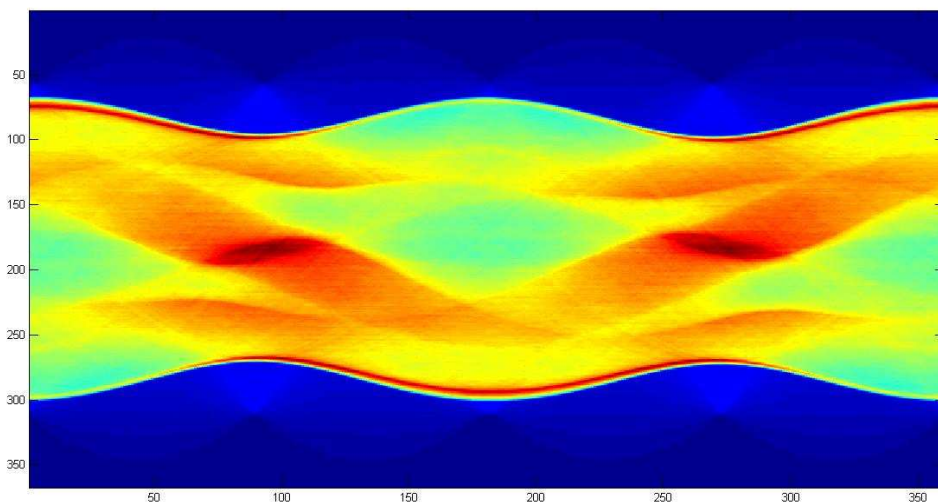
9. ábra
A szinogramok különbsége



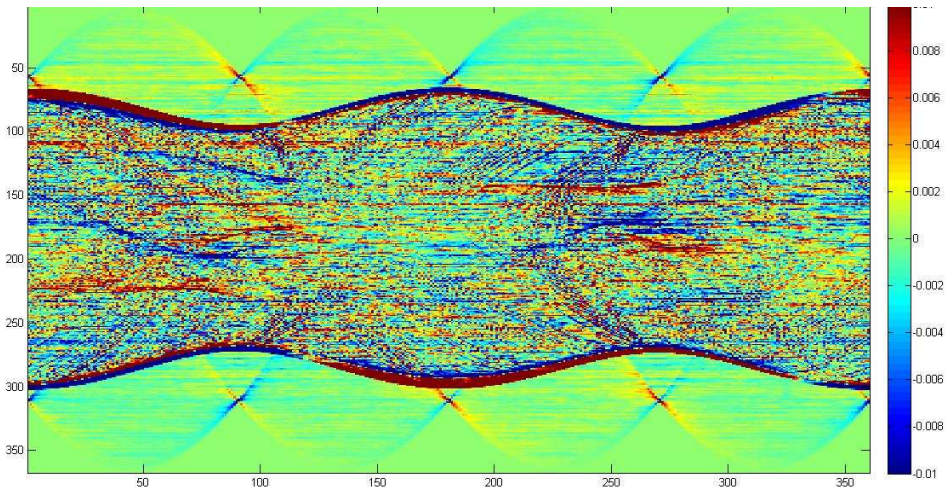
10. ábra
Shepp-Logan fantom



11. ábra
A Matlab Radon-transzformáltja a Shepp-Logan fantomról



12. ábra
A Shepp-Logan fantommal készített saját előrevetítem eredménye



13. ábra
A szinogramok különbsége

Láthatjuk, hogy az előrevetítő függvény által generált szinogramon erősebben megjelenik a „háttér” Radon-transzformáltja, vagyis magának a képnek a négyzet alakja. Ez azért érvényesül jobban a szimulációmban, mert az iteráció kezdő lépését egy homogén abszorpciójú fantomról indítottam.

Nagyobb eltéréseket látunk még a méretkülönbségek miatt a tényleges fantom (nem háttér) széleinél.

5.2 Az L2 norma

Alapvetően az a célunk, hogy eldöntsük, mennyire hasonlít a rekonstruált kép az eredeti fantom gyengítési együttható eloszlására. Ennek vizsgálatára több módszer is létezik, az egyik legegyszerűbb ilyen az L2 norma kiszámítása. Az L2 normát a két kép, vagy képrészlet között az alábbi formában definiáljuk:

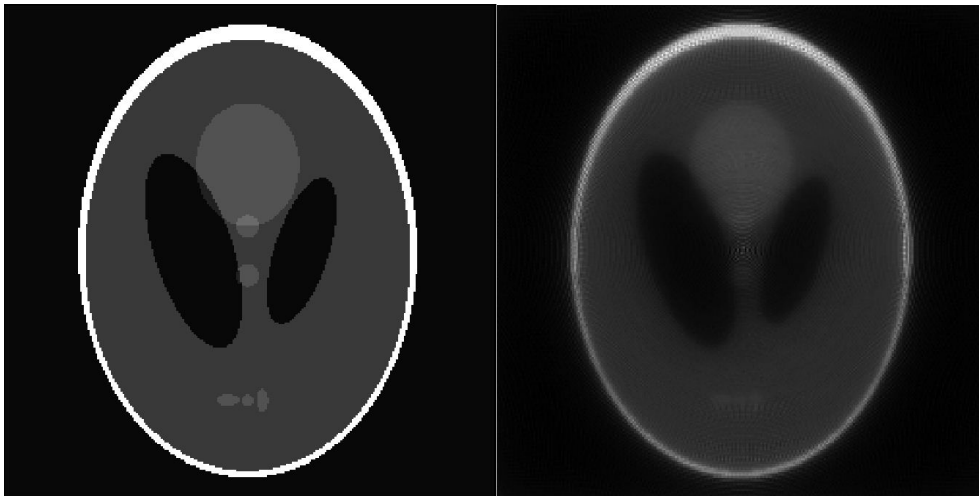
$$L2 = \frac{\sum_{i=1}^N (M_i^{\text{eredeti}} - M_i^{\text{rekon}})^2}{\sum_{i=1}^N (M_i^{\text{eredeti}})^2} \quad (16)$$

ahol M_i a kép(részlet) i . pixeljéhez tartozó érték (jelen esetben lineáris gyengítési együttható). Minél jobban hasonlít az eredeti eloszlásra a rekonstruált kép, annál jobb a képminőség, és annál kisebb az L2 norma. Gyakran érdemes ábrázolni az L2 norma értékeit az elvégzett iterációk függvényében, annak érdekében, hogy a módszer konvergenciájáról kapjunk információt. Ezen függvény grafikonjára L2-görbéként fogok hivatkozni a következőkben.

5.3 Tesztesetek

5.3.1 Shepp-Logan fantom rekonstrukciója, L2 norma iterációszám függése

Elsőként az általánosan rekonstrukciók tesztelésére használt Shepp-Logan fantomon mutatom be a képalkotás minőségét. A 14. ábrán látjuk ezt a fantomot, és a rekonstrukció eredményét. A mérés-szimulációt, vagyis a rekonstrukció alapját a 12. ábrán látható szinogram jellemzi, amit $3,6 \cdot 10^9$ részecske szimulációjával készítettem. A virtuális detektor 256 pixelből állt, ami azt jelenti, hogy egy pixel irányába nagyságrendileg 10^4 db foton indult (360 projekciót vettem fel). A beütésszámok relatív hibája ebben az esetben $\sim 1\%$ -nak becsülhető, vagyis kellően pontos mérést szimuláltam ahhoz, hogy a rekonstrukció minősége ne legyen korlátozva ez által. Az indított részecskék számának rekonstrukcióra gyakorolt hatását az 5.3.3 fejezetben tekinthetjük meg.

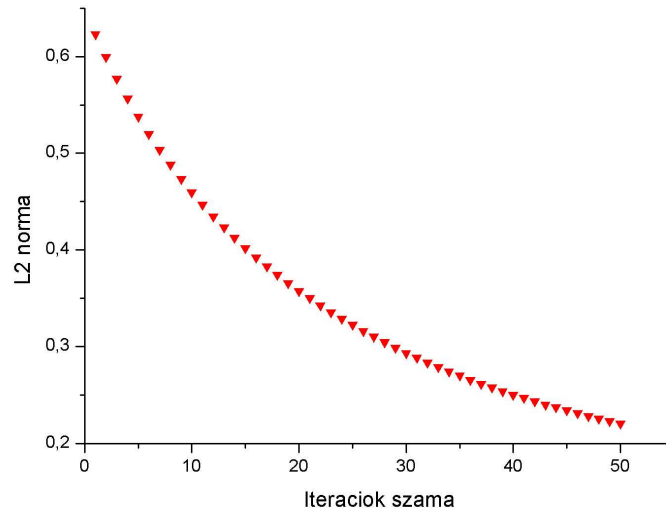


14. ábra
Shepp-Logan fantom: eredeti (balra), rekonstruált (jobbra) 50 iterációt végezve
Felbontás: 256x256

A 15. ábra mutatja az L2 norma (ld. 5.2 fejezet) értékét az egyes iterációk után. Az L2-görbe konvex, vagyis egyre kisebb mértékben csökkenő, az utolsó néhány iterációnak kis hatása van a képminőségre vonatkozólag⁵. Mivel a rekonstrukció hosszú időt vesz igénybe, ezért empirikus módon állapítottam meg egy megállási pontot, amikor is nem éri meg több időt

⁵ Az L2 görbe „gyenge” mérési eredmények esetén (kis jel/zaj arány) egy minimális érték elérése után növekszik, mivel a mérés által hordozott zaj felerősödik az iteráció során, a mérésből több információ nem nyerhető ki.

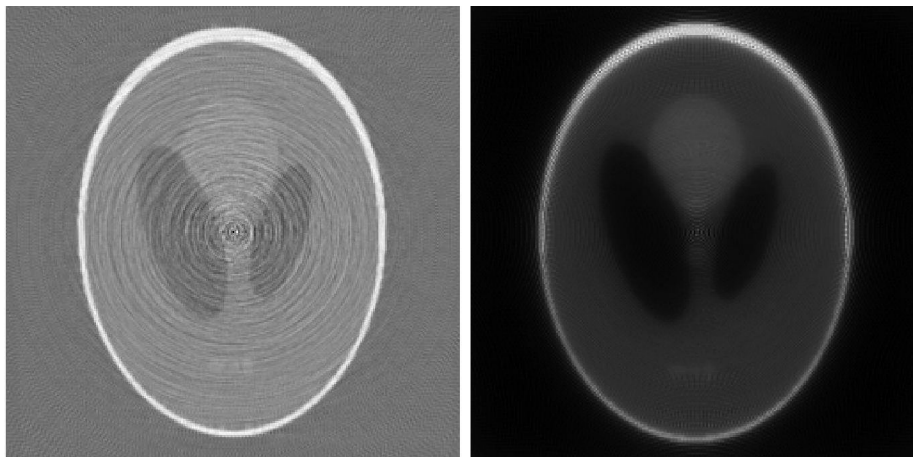
fordítani az iterációra, mivel nem kapunk lényegesen jobb minőségű fantomot. További elemzést a maximális iterációs szám beállításáról az 5.3.4 fejezetben adok.



15. ábra

Az egyes iterációk utáni L2 norma alakulása a Shepp-Logan fantom rekonstrukciója során

Érdekes összehasonlítást tenni a Matlab *iradon* függvényével, ami szűrt visszavetítést alkalmazva rekonstruálja a fantomot. A saját előrevetítem által generált szinogramot (12. ábra) adtam meg a rekonstrukciónak. Az összehasonlítást láthatjuk a 16. ábrán. Ahogy azt a szinogramok elemzése során is említettem, az általam generált szinogramon vízszintes csíkosság tapasztalható, ami a rekonstrukcióban koncentrikus körök formájában jelenik meg. Ez a saját rekonstrukciónon kevésbé jelenik meg (hiszen az előrevetítés konzekvensen hibás), viszont a Matlab rekonstrukciója esetében erősen látszik. Ennek kijavítása a közeljövő feladata marad.



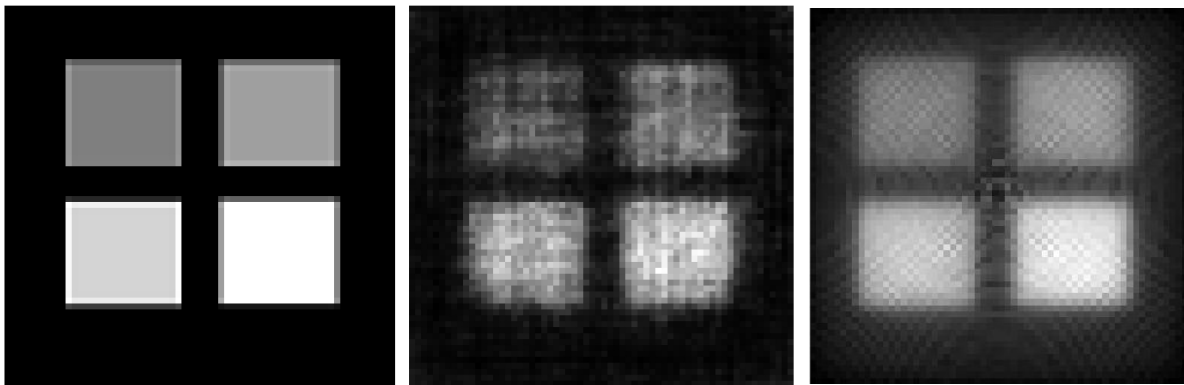
16. ábra

Saját rekonstrukció (jobbra) összehasonlítása a Matlab rekonstrukciójával (balra)

5.3.2 Összehasonlítás a CPU-n futó algoritmussal

A TDK-dolgozat szempontjából érdekes összehasonlítást lehet készíteni a korábbi elért eredményekkel. [2]-ben a bemutatott algoritmus(ok) nem használtak grafikus kártyát a rekonstrukcióra, futtatásukat CPU-n végeztem. Emiatt limitált lehetőségekkel rendelkeztem a részecskeszám, illetve a voxelmátrix mérete (felbontása) szempontjából. A rekonstrukció futási ideje inkább a napok skáláján mozgott. Ehhez képest a GPU-s implementáció segítségével sikerült elérni egy lényeges javulást mind a kép minőségében, mind a rekonstrukcióra szánt időben. Ez éppen azért történhetett meg, mert több részecske szimulációjára van lehetőség ugyanannyi idő alatt (nagyságrendi változás), valamint a visszavetítés is párhuzamosan történik, vagyis nemcsak a részecsketranszport-szimuláció, hanem a teljes rekonstrukció futási ideje is a töredékére csökken.

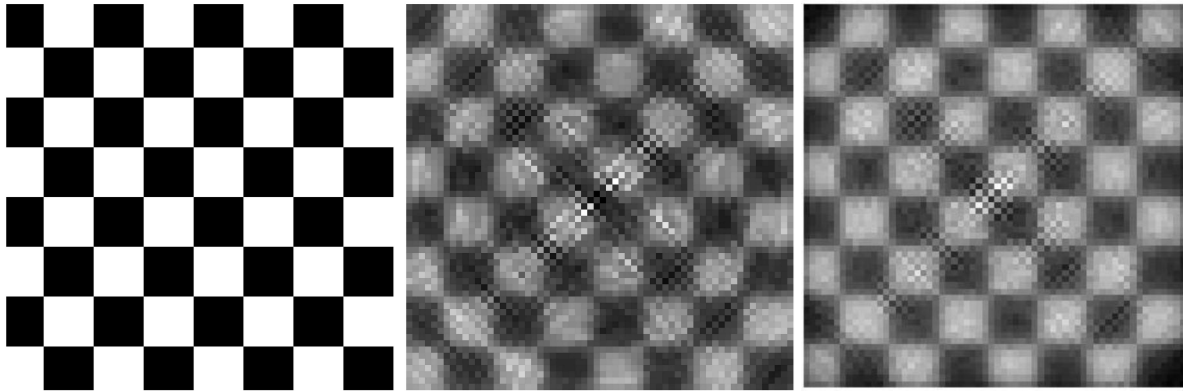
Az összehasonlíthatóság érdekében most 64x64-es felbontású fantomok rekonstrukcióját mutatom be, mivel ezek rekonstrukciója volt lehetséges CPU-n kivárható időn belül. A GPU-n működő rekonstrukció futási ideje nagyságrendileg percekben mérhető, részleteiben még az 5.5 fejezetben olvashatunk a futási időkről.



17. ábra

Eredeti fantom (balra), CPU-n rekonstruált (középen) – 2013.10, GPU-n rekonstruált (jobbra) – 2014.10.

A 17. ábrán (balra) látható különböző abszorpciójú négyzeteket rekonstruáltam, középen láthatjuk a tavalyi TDK-munkámban [2] szereplő eredményt, jobb oldalt pedig a GPU-n futó algoritmusét. Azonnal észrevehető a minőségbeli javulás, ami elsősorban az előrevetítésben indított részecskék számának növelésének köszönhető, pontosabb mérés-szimulációt, illetve a rekonstrukcióban is jobb jel/zaj arányt jelent. Ugyanez megfigyelhető a 18. ábrán látható „sakktabla” fantomon. A képek minőségi elemzésére nem fordítottam időt az eleve gyenge felbontás miatt.



18. ábra

Eredeti fantom (balra), CPU-n rekonstruált (középen) – 2014.02, GPU-n rekonstruált (jobbra) – 2014.10.

Az indított részecskék számát két nagyságrenddel megnövelhetjük, ha a grafikus kártyán implementáljuk az algoritmust, a bemutatott esetekben ~10 millió részecske (CPU) helyett milliárdos nagyságrendben szimuláltam fotonokat.

5.3.3 Konvergencia

Az 1.2 fejezet rámutatott arra a pusztán matematikai tényre, hogy a maximalizálandó likelihood-függvénynek akkor van minimuma, ha több LoR-t jelölünk ki az iterációban, mint ahány voxelből áll a rendszer. A visszavetítésben ehhez mérten kell megválasztanunk a válaszegyenesek számát. Ez szükséges, de nem elégséges feltétele a konvergenciának, hiszen a (15) iterációban valójában várható értékek szerepelnek, amiket csak (valamilyen hibával) becsülni tudunk. A módszer konvergenciájáról további elméleti megfontolásokat olvashatunk még [1]-ben, ahol megmutatták, hogy az EM algoritmus konvergens, ha a említett várható értékeket nagy pontossággal tudjuk becsülni (minden határt túllépő pontossággal), illetve a (12) sorfejtés érvényes. A gyakorlatban ez nem múlik máson, mint a szimuláció statisztikáján. Tekintve (15)-öt észrevehetjük, hogy az iteráció hajtóereje az előrevetítés eredménye, vagyis az, hogy mennyire tér el az aktuálisan iterált gyengítési együttható eloszlás mellett mérhető beütésszám a valós mérés beütésszámaitól. Ennek belátására tekintsük (15) iteráció fixpontját:

$$\mu[j] = \frac{\sum_{lor \in L_j} (X_{lor}[j] - X_{lor}[j+1])}{\sum_{lor \in L_j} \left(\frac{X_{lor}[j] + X_{lor}[j+1]}{2} \right) \cdot L_{lor}[j]}$$

A fixpontban ugyanis $Y_{lor} = \hat{Y}_{lor}$. A számlálóban a j . voxelben elnyelődött részecskék számának összege marad (ΔN), a nevezőben viszont a befutott úthosszakkal (l) súlyozott összegét kapjuk a j . voxelbe bejutott és abból kimenő fotonszámok átlagának (N). Ezekkel az egyszerűsített jelölésekkel a kérdéses fixpontot (17) alakba írhatjuk át.

$$\mu = \frac{\overline{\Delta N}}{\overline{Nl}} \quad (17)$$

A (17)-ben $\bar{\quad}$ -sal jelölt átlagot a LoR-okra kell elvégezni, vagyis ha feltesszük, hogy N értéke közel azonos a LoR-ok közt, vagyis minden válaszegyenes mentén hozzávetőlegesen ugyanannyi részecske halad, akkor N -et kihozhatjuk a várható értékből, és máris a (18) összefüggéshez jutunk.

$$\mu \bar{l} = \frac{\overline{\Delta N}}{N} = 1 - \frac{N - \overline{\Delta N}}{N} \quad \Rightarrow \quad 1 - \mu \bar{l} = \frac{N - \overline{\Delta N}}{N} \quad (18)$$

(18) pedig nem más, mint a Beer-Lambert törvény közelítése $\mu \bar{l} \ll 1$ esetben.

Mindezzel arra igyekszem rávilágítani, hogy az algoritmus konvergenciája három dolog által van biztosítva:

- 1) Több LoR-t jelölünk ki, mint ahány voxelre bontjuk a rekonstruálandó térfogatot
- 2) A voxelméret (d) elég kicsi, a gyengítési együtthatóval vett szorzatra $\mu d \ll 1$ áll fenn
- 3) Az előrevetítés és a mérés/mérés-szimuláció statisztikája kielégítően pontos (hiszen az $Y_{lor} - \hat{Y}_{lor}$ különbség hajtja az iterációt a fixpontja felé!)

Ez utóbbi statisztika, mint tudjuk, a Poisson eloszlásra jellemző tulajdonságokkal rendelkezik, vagyis például relatív hibája $1/\sqrt{N}$ alakban írható fel, ha N a szimulált részecskék száma. Ahhoz, hogy néhány százalékos relatív hibát érjünk el, legalább ezer fotonnak kell a detektorra érkeznie, 1%-os hibához 10000-nek. Ez azt jelenti, hogy az előrevetítésben érdemes arra figyelni, hogy legalább 10^3 , de inkább 10^4 nagyságrendben indítsunk fotonokat minden egyes detektorpixel felé.

A konvergenciát a gyakorlatban az L2-görbe lefutásából állapítottam meg. A 15. ábrán már láthattunk egy ilyen görbét, amin azt tapasztaljuk, hogy egyre csökkenő mértékben csökkennek az L2 norma értékei az egyes iterációk után. Másfajta viselkedést tapasztalunk, ha rossz statisztikával szimulálunk, vagyis nem indítunk elég részecskét az előrevetítésben. Ekkor már az első iterációnál sem javul az L2 norma, vagyis a legjobb kép a kiinduló eloszlás lesz. Minden egyéb esetben, amikor a szimulált részecskék száma elég ahhoz, hogy a görbe

legalább néhány iteráció erejéig csökkenő legyen, akkor az továbbra is monoton csökkenést fog mutatni, aszimptotikusan viselkedik. Az emissziós tomográfiára implementált ML-EM algoritmussal szemben [9] ez különös viselkedése a módszernek, hiszen azt várjuk, hogy egy idő után az iterációban felerősödnek az elégtelen statisztika miatt a zajok, és az L2 görbe egy minimumhely elérése után monoton növekedni kezd. Mivel azonban a visszavetítés épp a Beer-Lambert törvény alapján számolja az X értékeket, ezért logikus, hogy a fixpontból a módszer jelentősen nem mozdulhat el. Az L2 görbe e szélsőséges viselkedése tehát a szimulált fotonok számának függvénye, a tapasztalat azt mutatja, hogy a konvergencia eléréséhez (15. ábrához hasonló L2 görbe) minimum ~500-1000 részecskének kell elérnie az egyes detektorpixeleket az előrevetítésben.

Az L2-görbe aszimptotikus viselkedése miatt felvetődik a megállási pont problémája. Mivel nincs egyértelmű minimumhely, ezért ki kell jelölni azt az iterációs számot, aminél már nem érdemes több iterációt elvégezni, a kép minősége szignifikánsan nem javul. Ezt úgy állíthatjuk be, hogy az iterációt megállítjuk azon a ponton, ahol már 0,5%-nál kevesebbet csökkent az L2-norma az előző iterációhoz viszonyítva. Ezt a 0,5%-ot empirikusan választottam meg, a későbbi vizsgálatok érdekes témája lehet az L2-görbék karakterizálása, ami alapján a megállási pont logikus módon megválasztható. Néhány gyors teszt azt sejteti, hogy az L2 norma értéke nagyjából reciprokos összefüggésben van az iterációs számmal.

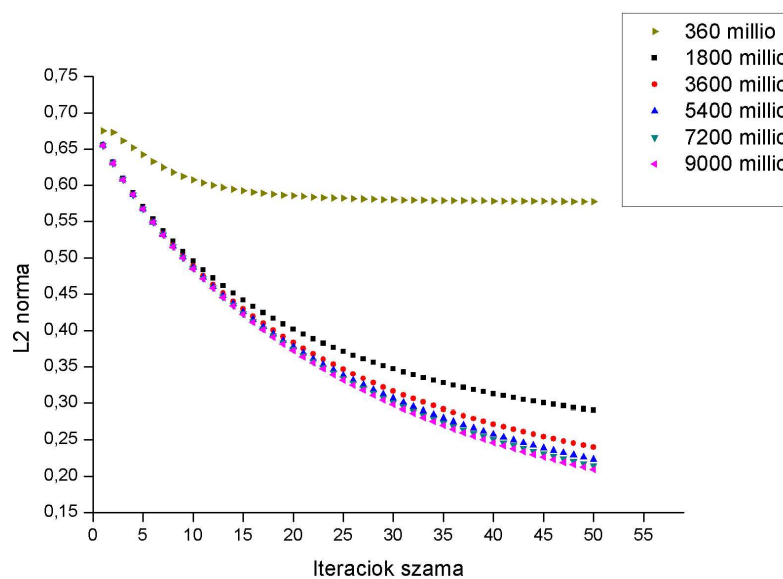
5.3.4 Paraméter optimalizáció, futásidők

Az iteratív ML-EM algoritmus egyik nagy kihívása a többdimenziós paramétertérben való optimalizálás. Ahhoz, hogy ezt megtegyük, ismernünk kell a rekonstrukció minőségét befolyásoló szabad paramétereket, és ezek változtatásának hatását. Az elérendő cél az, hogy a lehető legkevesebb idő alatt minél jobb képminőséget kapjunk, illetve a mérésből származó információt minél jobban kihasználjuk.

A vizsgálatokat [9]-hez hasonlóan végeztem el, ahol egy emissziós tomográfiára (PET) alkalmas ML-EM algoritmus elemzését találjuk. A képminőségre két paraméter hatását vizsgáltam: mérés/mérés-szimuláció jel/zaj aránya, illetve az iterációban lévő előrevetítés jel/zaj aránya. A képminőséget az L2 normával jellemeztem.

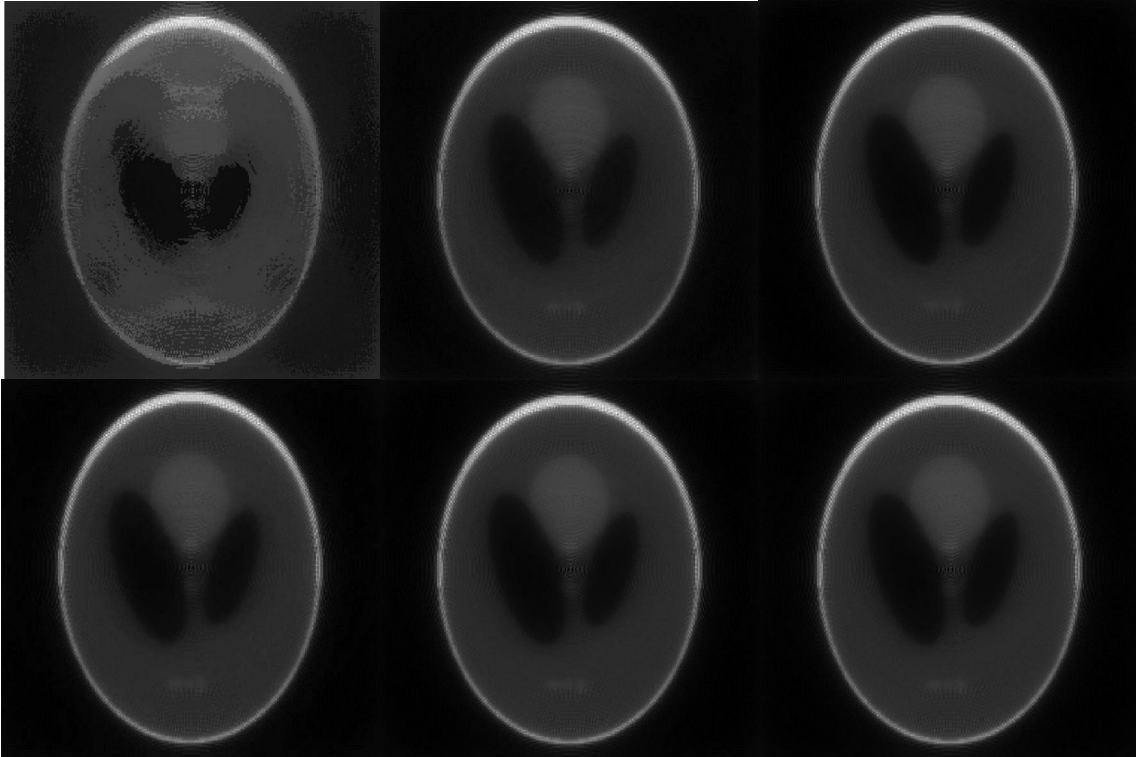
Az 5.3.1 fejezetben bemutattam, milyen rekonstrukcióra képes az algoritmus a Shepp-Logan fantom esetében. Most annak céljából fogom ezt bemutatni, hogy megismerjük, mennyi részecskét érdemes szimulálni az iteráció során. Ehhez létrehoztam egy eddig

bemutatottaknál is jobb minőségű szinogramot a fantomról, a mérés-szimulációban 18 milliárd részecskét indítottam a forrásból, azonban a rekonstrukció során az előrevetítésben kevesebb fotont szimuláltam. Ennek a vizsgálatnak célja az, hogy optimumot találjunk képminőség és a rekonstrukcióra fordított idő függvényében, pontosabban arra vagyunk kíváncsiak, hogy az iterációban hány részecske szimulációja elégséges ahhoz, hogy a mérés-szimuláció információ-tartalmának közel 100%-át kinyerjük annak során. A részecskeszámok itt most az egy iterációban szimulált összes részecskét jelentik, vagyis ha le akarjuk fordítani ezt a pixelek átlagos beütésszámaira, akkor el kell osztanunk a projekciók számával (360), valamint a pixelek számával (250).



19. ábra
Az L2 norma alakulása különböző indított részecskeszámnál

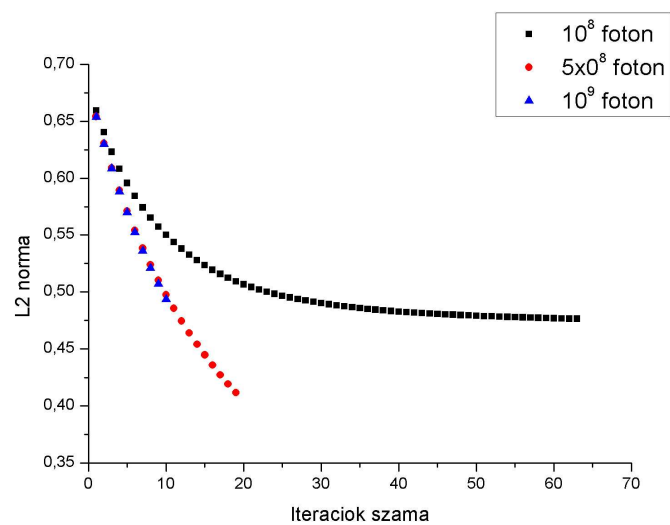
A 19. ábra mutatja meg, hogy miként alakul az L2-görbe különböző mennyiségű foton szimulációja esetén. A rekonstrukció minőségét vizuálisan is megtekinthetjük a 20. ábrán, ahol az 50. iteráció elvégzése után jelenítettem meg a rekonstruált képeket. Összességében következtetésként azt vonhatjuk le, hogy a rekonstruált kép minősége jelentősen nem javul $3-4 \cdot 10^9$ foton szimulációjához képest, ha tovább növeljük a részecskeszámot. Az előző fejezetben a konvergenciára vonatkozó megjegyzések tekintetében ez azt jelenti, hogy a detektorpixeleket ért fotonok száma legalább $1-2 \cdot 10^4$ legyen (ez $\sim 1\%$ -os jel/zaj aránynak felel meg).



20. ábra

Balról-jobbra, fentről-le sorrendben: 360, 1800, 3600, 5400, 7200, 9000 millió részecske szimulációjával rekonstruált fantom az 50. iteráció után

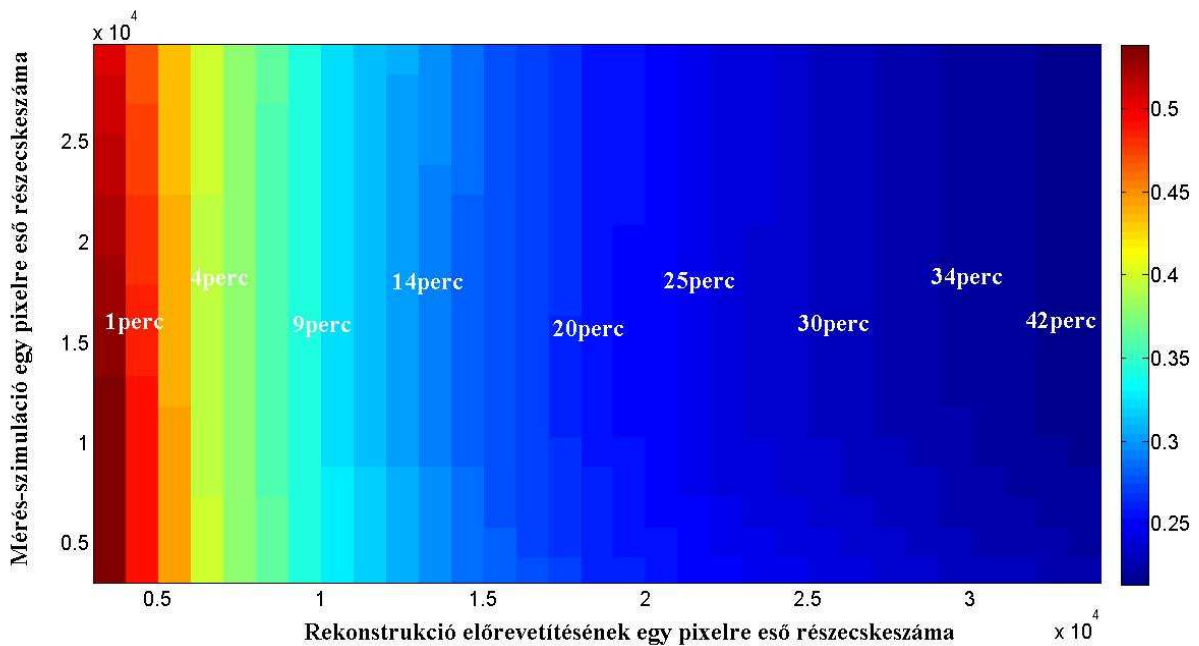
Ha a rekonstrukcióra korlátozott idő áll rendelkezésre, akkor kérdéses lehet, hogy hány részecskét szimuláljunk: inkább többet, és kevesebb iterációt végezzünk el, vagy kevesebbet, viszont többet iteráljunk. Ennek problémáját mutatja be a 21. ábra. Az elérhető legjobb L2 érték egy optimális beállítás mellett érhető el, ennek becslésére végeztem el az alábbi vizsgálatot.



21. ábra

Elérhető képminőség korlátozott rekonstrukciós idő alatt (5 perc)

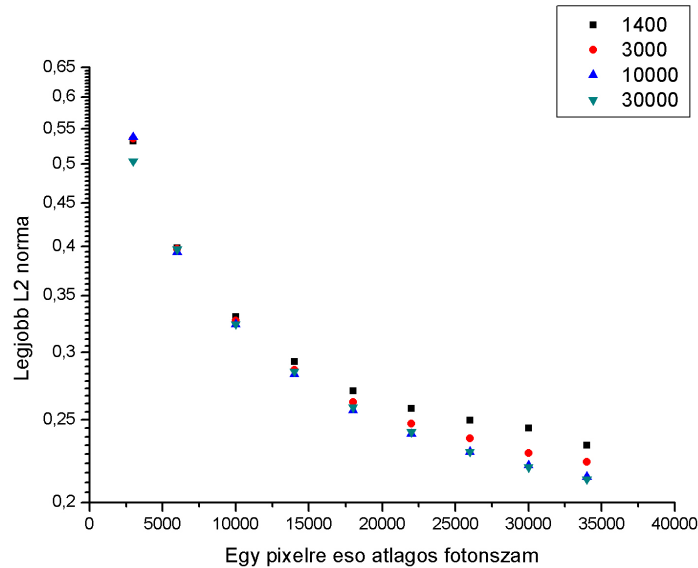
Az előzőekhez képest a paraméterteret kibővítettem a mérés-szimulációban indított részecskék számával. Az előrevetítés relatív hibájának szemléltetése kedvéért a fotonszámok alatt most az átlagosan egy pixelre eső részecskeszámot értem. Ilyen skálájú paraméterterben (mérés-szimulációban, illetve az iteráció előrevetítésében indított fotonok száma) szinkódolással az elérhető legjobb L2 normát tüntettem fel a 22. ábrán. Meglepő lehet, hogy a mérés-szimuláció statisztikájától kevésbé függ a rekonstrukció minősége, ez azonban azt jelenti, hogy gyenge (kis jel/zaj arányú) mérési adatokból is tudunk jó minőségű rekonstrukciót végrehajtani. A 23. ábra szemléletesebb olyan szempontból, hogy könnyebben észrevehetjük, hogy ha rossz statisztikájú mérési adatok állnak rendelkezésre, az csak akkor mutatkozik meg egyáltalán a képminőségben, ha sok részecskét szimulálunk a rekonstrukció során is, de hatása ekkor is mindössze néhány százalékos. Hasonló következtetésre jutottak ML-EM PET-képrekonstrukcióban is [9].



22. ábra
Elérhető legjobb L2 norma, valamint az eléréséhez szükséges rekonstrukció ideje

A 22. ábrán feltüntettem a rekonstrukció futási idejét is, ez csakis a rekonstrukcióban indított fotonszámtól függ, így az ábrát úgy kell értelmezni, hogy az igénybevett idő a függőleges tengely irányában nem változik. Ezzel azt kívánom szemléltetni, hogy érdemes kompromisszumot kötni a futási idő, és a kép minősége között, hiszen egy bizonyos statisztikánál előfordulhat, hogy nem érdemes jobbat előállítani, hiszen többszörösére nőhet a futási idő, ugyanakkor az L2 norma csupán pár százalékos csökken. Az ábrára tekintve

levonhatjuk azt a következtetést, hogy ~20000foton/detektorpixel szimulációjával elérhetünk egy elég jó képminőséget belátható időn belül (ez jelen esetben 20 perc volt).



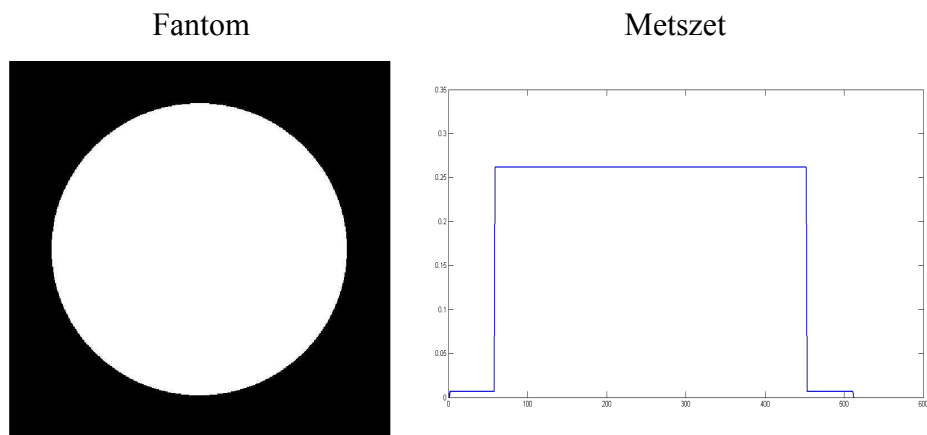
23. ábra

Az elérhető legjobb L2 norma az iterációnkénti előrevetítés fotonszámának függvényében, különböző mérés-szimulációs statisztikáknál

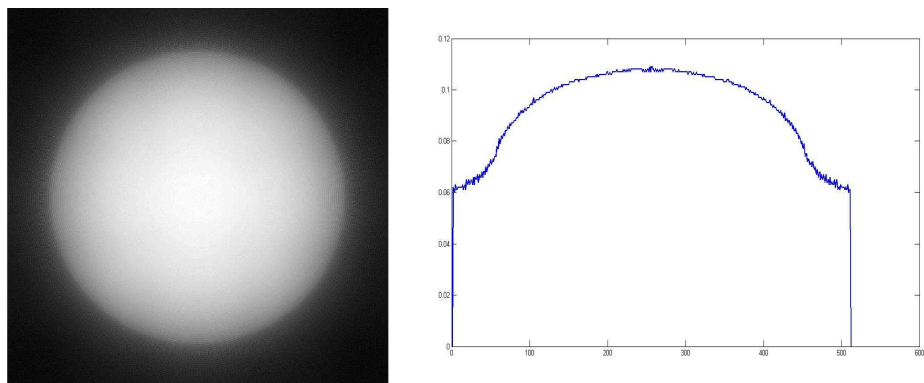
Azt mondhatjuk kitekintésképp a háromdimenziós rekonstrukció felé, hogy a futási idők nem túl biztatóak. Gyors számolást végrehajtva az eddigiek alapján ezt könnyen láthatjuk. Az előzőkben megtudtuk, hogy egy kielégítő minőségű képszintézishez az előrevetítésben 10^4 nagyságrendben kell fotonokat szimulálnunk egy detektorpixelre vonatkoztatva. Egy 1536×768 -as felbontású detektornál a 10^9 -es nagyságrendbe esik így az egy projekcióban szimulálandó részecskék száma. Ez két nagyságrenddel több, mint kétdimenziós esetben, ami 20-30 órára jósolja a futási időt. Szerencsére nem ennyire rossz a helyzet, hiszen az előrevetítés a fejlesztés jelen fázisában még optimalizálatlan, valamint a rendelkezésre álló grafikus kártya számítási kapacitása sincs tökéletesen kihasználva. Az algoritmus további gyorsítása mindenképp a jövőbeli feladatok egyike lehet.

5.3.4 Homogén abszorpciójú fantom rekonstrukciója, középvonal mentén vett metszete

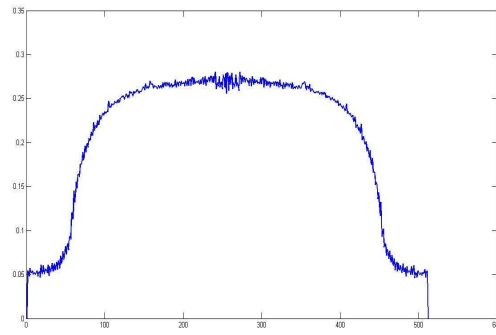
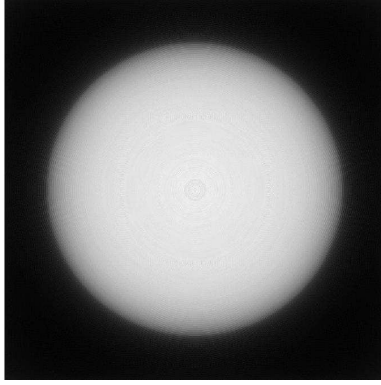
A korábbi vizsgálatok során már előkerült annak kérdése, hogy mikor érdemes az iterációt befejezni, melyik az a maximális iterációs szám, mettől nem érdemes több számítást végezni, az eredményként kapott képünk nem lesz jobb minőségű (esetleg rosszabb is lehet). A mérési eredmények által hordozott zaj egy idő után a rekonstrukcióban felerősödik, az L2 norma értéke romlik az iterációs szám növelésével. Ennek vizsgálatára egy újabb matematikai fantomot állítottam elő, amit a 24. ábrán tekinthetünk meg. Elvégeztem rajta 50 iterációt, és a közbenső eredményeket is ábrázoltam, valamint a kép középvonalán vett metszetét is feltüntettem. Ezeket tekinthetjük meg a 25.-28. ábrákon.



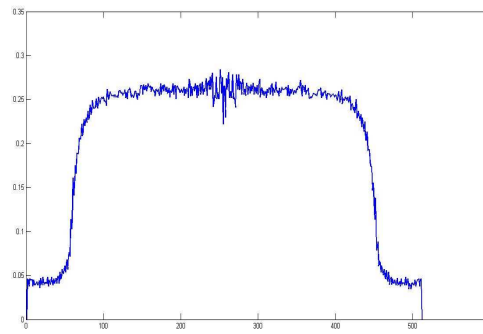
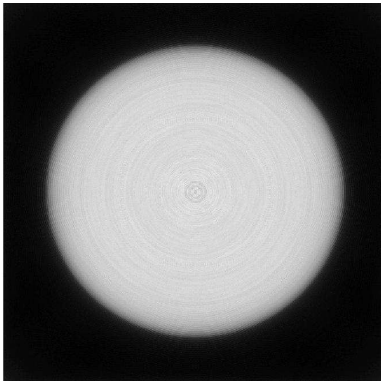
24. ábra Eredeti fantom, és keresztmetszete



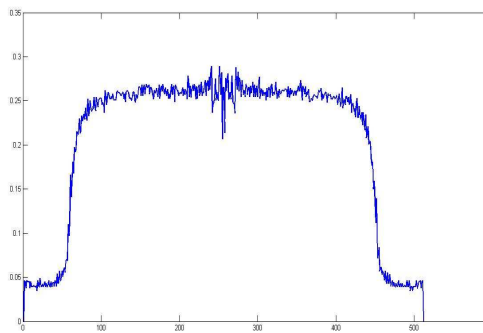
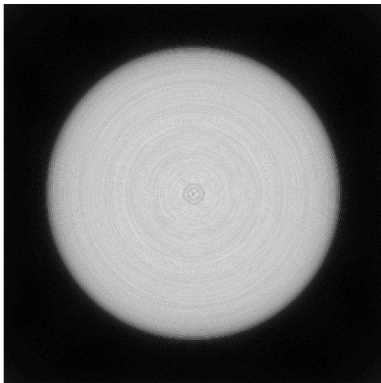
25. ábra 1. iteráció



26. ábra 10. iteráció



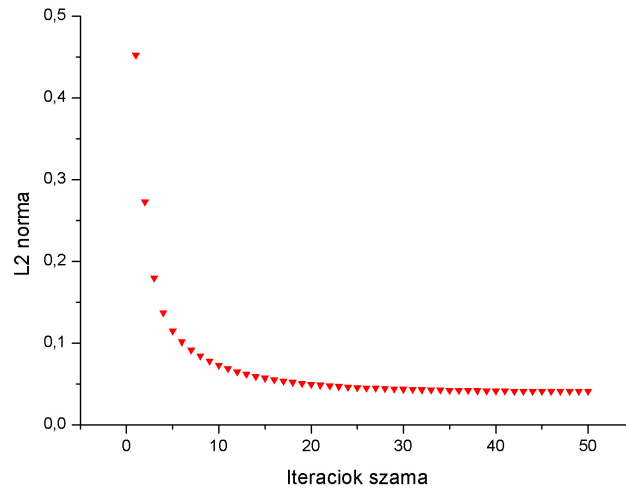
27. ábra 30. iteráció



28. ábra 50. iteráció

A képekre tekintve azt tapasztaljuk, hogy az iteráció során a fantom kontúrjai egyre élesebbek lettek, de ezzel párhuzamosan felerősödtek a rekonstrukció műtermékei. Ez utóbbi alatt a fantom közepén feltűnő koncentrikus körökre gondolok, ami legjobban a 28. ábrán látszik. Ennek eredete valószínűleg programozási hiba, aminek kijavítása mindenképpen célszerű lehet a jövőben. A koncentrikus körök megjelenése valószínűleg kapcsolatban lehet a szinogram csíkosságával, aminek problémáját az 5.1 fejezetben ismertettem. A 29. ábrán a rekonstrukcióhoz tartozó L2-görbét is megtekinthetjük, aminek segítségével

következtethetünk arra, hogy ezen fantom esetében a 20. iteráció után nem érhetünk el javulást (a metszetekre tekintve a kontúrok jobbak lesznek, viszont a műtermékek erősebben kirajzolódnak).



29. ábra
L2 norma alakulása

6. Tesztek valós mérési eredményekkel

A képrekonstrukciós szoftver fejlesztésében új kihívásokkal szembesülünk, ha valós mérési adatokon teszteljük azt. Ezzel egy lépést tehetünk annak irányába, hogy az algoritmus felhasználása bekerülhessen az orvosi gyakorlatba. Lehetőségem volt a rekonstrukciót kipróbálni az NTI Oktatóreaktorában működő CT készülékkel. A mérési adatok kinyerése, a kalibrációs eljárások elvégzése nem volt része a TDK-munkámnak, ezeket Kleizer Gábornak és Tölgyesi Botondnak köszönhetem.

6.1 Kísérleti elrendezés

A kísérleti elrendezés egy hagyományos kúpsugaras (cone-beam) CT geometriát valósít meg, azzal a különbséggel, hogy a mintát lehet forgatni egy tengely körül, nem pedig a forrás-detektor rendszert. A mérési eredményeket szolgáltató CT berendezés az alábbi egységekből állt:

- Source Ray Inc. SB-80-1k mikrofókuszos röntgenszó
- Dexela 1207-as számú nagyfelbontású röntgendetektor
- Phi Instruments lineáris és rotációs mozgatómodulok

A röntgendetektor rekonstrukciós szempontból lényeges tulajdonságait az 1. táblázatban külön kiemeltem, mivel ezek minden rekonstrukciónál álló paraméternek számítotak.

| | |
|------------|----------------|
| Pixelméret | 74,8 μ m |
| Felbontás | 1536x864 pixel |

1. Táblázat
A Dexela 1207 röntgendetektor tulajdonságai

A berendezés további specifikációiról bővebben [10]-ben olvashatunk.

6.2 A valós mérési adatokkal végzett rekonstrukció kihívásai

A dolgozatban eddig a pontig olyan rekonstrukciókat mutattam be, amik bejövő adatként a mérés-szimuláció eredményeit kapták meg. Ezt a rekonstrukcióban is használt előrevetítéssel állítottam elő, ezért egyszerű volt a „mérés” és a szimuláció összehangolása. A valós mérés során előállított projekciók esetében azonban felmerülnek olyan problémák, amik eddig nem kerülhettek elő.

- (i) Először is tudnunk kell, hogy miként egyeztessük össze a detektorból kiolvasható adatokat az iterációs sémával. Tekintsük meg újra az 1.3 fejezetben szereplő (15) iterációt:

$$\mu[j] = \frac{\sum_{lor \in L_j} (X_{lor}[j] - X_{lor}[j+1])}{\sum_{lor \in L_j} \left(Y_{lor} - \hat{Y}_{lor} + \frac{X_{lor}[j] + X_{lor}[j+1]}{2} \right) \cdot L_{lor}[j]}$$

A bevezetőben tisztáztuk, hogy az Y és \hat{Y} mennyiségek rendre a valós mérési eredményeket, és a rekonstrukcióban aktuális iterált gyengítési együttható eloszlás mellett szimulált mérés eredményeit jelentik. Pontosabban ezek részecskeszámokat jelentenek, ahogy az X -szel jelölt mennyiségek is, a voxelekbe bejutott részecskék számára vonatkoznak. Továbbá a szimuláció során, a visszavetítésnél ismernünk kell, hogy hány részecske indult el a forrásból az (egyébként önkényesen kijelölt) LoR mentén. A részecskeszámokat azonban nem tudjuk megmérni, így másképp kell következtetni ezekre a mennyiségekre.

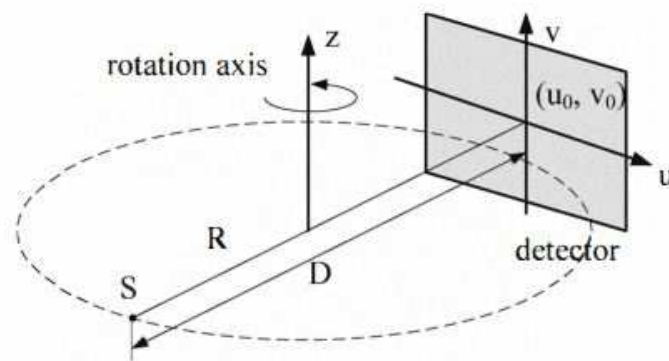
- (ii) Egy másik jelentős probléma a röntgenforrásból származó fotonok energia szerinti eloszlása. A leképezést egy polikromatikus nyalábbal hajtjuk végre, aminek egy, a röntgen-csőre jellemző spektruma van. A röntgen sugárzás lineáris gyengülését jellemző μ gyengítési együttható azonban nemcsak anyag (rendszer), hanem energia-függő is. Vagyis ahhoz hogy érték helyes rekonstrukciót kapjunk, az iteráció során végzett szimulációkban (előrevetítésekben) ugyanazt a spektrumot kell használnunk, mint ami a valódi mérés során jellemezte röntgenforrást.
- (iii) A harmadik lényegi különbség az eddigiekhez képest, hogy a mérési geometria nem ismert, vagy legjobb esetben is, csak véges pontossággal ismert. Ezalatt arra gondolok, hogy a mérés-szimuláció során (amikor matematikai fantomokon végzem el az előrevetítést) alkalmazott geometriát egzaktul felhasználhatjuk a rekonstrukcióban alkalmazott előrevetítésekénél (nincs diszkrétancia a „mérés” és a rekonstrukció között). Természetesen a valós mérési geometriát meg lehet határozni geometriai kalibráció segítségével, azonban a becsült paraméterek hibával terhelt lesznek. Ezen hibák rekonstrukció minőségére vonatkozó hatását meg kell vizsgálni a tesztelés során.
- (iv) Érdekes még megemlíteni az egyes projekciók felvételekor érvényes geometriai viszonyok problémáját. Ha a mérési adatok gyűjtése folyamatos üzemmódban történik (a forgatás nem áll meg az expozíció idejére sem), akkor utólag nem triviális meghatározni azt, hogy az egyes projekciókat milyen szögelfordulások jellemzik, illetve, hogy hányadik projekció jelenti a félfordulatot (180°), vagy éppen a teljeset.
- (v) Végül kiemelném még a mérést terhelő zajokat, amiknek többségét szintén kalibráció segítségével lehet csökkenteni. Az előrevetítési szimuláció során azért nem kapunk tökéletes Radon-transzformáltat a fantomról, mert a részecskék száma Poisson-statisztikát követ, aminek relatív hibája N részecske esetén $1/\sqrt{N}$, viszont más zajforrás

nincs. A valóságban a detektorunk nem tökéletes, gondolhatunk például ún. halott pixelekre (dead pixel), vagy arra, hogy a detektálásnak 1-től különböző hatásfoka van.

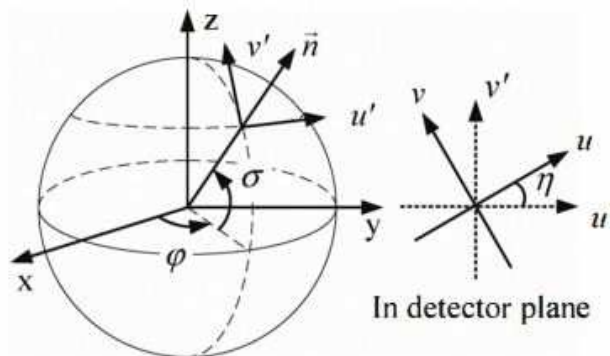
Összefoglalva tehát a rekonstrukció fő kihívásai: részecskeszámok becslése a detektor által mért adatokból, röntgen-spektrum, geometriai kalibráció, folyamatos üzemmódú adatgyűjtés, és a zaj hatása a rekonstrukció minőségére.

6.3 A rekonstrukcióhoz szükséges geometriai adatok

A leképezés során érvényes elrendezést geometriai kalibrációval tudjuk meghatározni. A 30. és 31. ábrán láthatjuk, hogy milyen paramétereket tudunk becsülni az eljárás során. Az előbbi ábrából az egyes elemek elhelyezését jellemző távolságokat, valamint a főnyaláb irányát ismerhetjük meg, utóbbi pedig a detektor dőlésszögeit jellemzi.



30. ábra
A CT berendezés geometriája [11]



31. ábra
Detektor elhelyezkedését jellemző koordináták [11]

[10] szerint a detektor gondos beállítása esetén (közel merőleges a detektorsíkra a főnyaláb) a 31. ábrán látható szög-paraméterek kalibrálatlansága nincs hatással a teljes geometriai

kalibráció pontosságra, így ha ezek a szögek kellően kicsik, akkor elhagyhatók a kalibrációs lépések közül, valamint a rekonstrukció minőségére sem lesznek várhatóan nagy hatással. Mivel a detektor beállítása nagy pontossággal merőleges lett a főnyalábra, így a rekonstrukcióhoz szükséges geometriai paraméterek lényegében a 30. ábrán látható mennyiségek. Ezeket a 2. táblázatba gyűjtöttem ki, ahol megadtam a kalibráció során számolt értéküket is, amiket a 6.5 fejezetben látható rekonstrukcióban használtam (a geometriai kalibráció elvégzését [10] mutatja be).

| Paraméter jele | Definíciója | Értéke (6.5 fejezet rekonstrukciójában) |
|----------------|---|--|
| D | Forrás-detektor távolság | 43,2 cm |
| R | Forrás-forgástengely távolság | 16,5 cm |
| u_0 | Főnyaláb és detektor metszéspontjának vízszintes koordinátája | 802 pixel |
| v_0 | Főnyaláb és detektor metszéspontjának függőleges koordinátája | 432 pxel |

2. Táblázat
A 30. ábra paraméterei és értékei

6.4 Elvárások

A rekonstrukciótól az eddigi tesztek alapján máris lehetnek várakozásaink, főleg a képminőséget illetően. Láttuk ugyanis (pl. 5.3.4 fejezet), hogy a kontúrok elmosódása egy jellemző tulajdonsága a rekonstrukciónak, de a műtermékek megjelenése egy szűrt visszavetítéshez képest nem számottevő, vagyis az algoritmus kevésbé érzékeny a rossz jel/zaj arányra. Észrevehettük azt is, hogy a módszer a kép közepén hamarabb konvergál, vagyis a voxelmátrixban a középponthoz közel helyezkedő voxelek hamarabb elérik a valóságosnak megfelelő értéket, mint a periférián fekvők. Ezekon kívül számíthatunk még esetleg spektrumfelkeményedés okozta műtermékekre is, hiszen jelenleg még nem áll rendelkezésre adat a röntgen-cső spektrumáról.

6.5 Tesztesetek

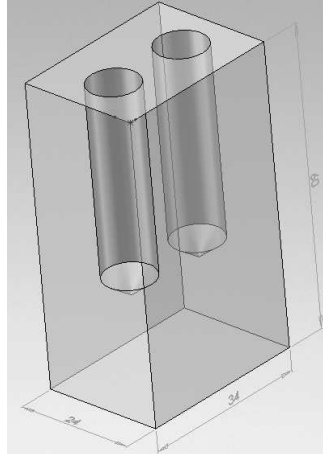
A 6.2 fejezetben áttekintettük, hogy milyen kihívásokkal szembesülünk a valós mérési adatok rekonstrukciója során. A detektor által mért adatok 1 és 2^{14} közötti egész számok, amikről feltételezzük, hogy $\ln\left(\frac{I}{I_0}\right)$ formában felírható mennyiséggel arányosak (vagyis a Radon-transzformáltat becslik). Ahhoz, hogy részecskeszámmal arányos mennyiséget kapjunk, venni kell a mérési adatok exponencializáltját. Ebből kiindulva, a visszavetítésben a forrásból a detektorpixelek felé induló fotonok számát a „háttérből” becsülhetjük, vagyis keressünk olyan pixeleket, amikhez tartozó LoR-ok mentén gyakorlatilag nincs abszorpció, és átlagoljuk a hozzájuk tartozó beütésszámokat. A kapott fotonszámokhoz igazítjuk a szimulációban indított részecskék súlyát is: ha M db fotont akarunk szimulálni, de az adatok „háttérpixeleiből” számolt részecskeszám N db, akkor egy fotont vegyünk figyelembe N/M súllyal a szimulációban.

A röntgenső spektrumának kérdése a TDK-dolgozat elkészüléséig még nyitott maradt, ugyanis jelenleg nem áll rendelkezésre adatsor a spektrumra vonatkozólag. A spektrumot kétféleképp lehet meghatározni:

1. Kísérletileg, aminek mérés technikája igen bonyolult
2. Szimulációval (pl. MCNP), ehhez azonban ismerni kell a forrás felépítését

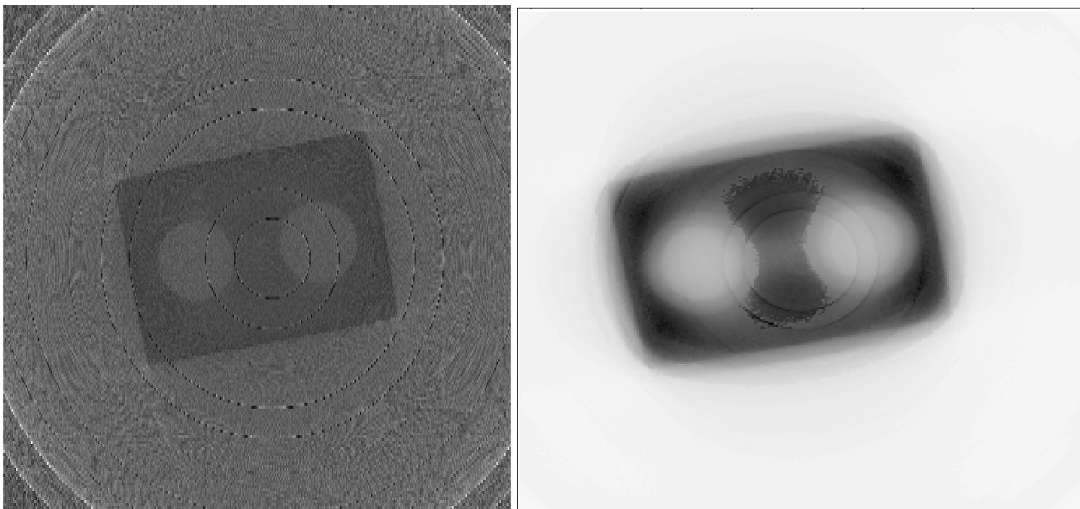
A jelenlegi algoritmusban a röntgensugárzásra vett effektív energiát használtam a spektrum helyettesítésére, vagyis lényegében egy monokromatikus nyalábot indítottam a szimulációban a spektrumban szereplő legnagyobb energia 1/3 részét véve. [12] szerint ugyanis az effektív energiát jól becsli a maximális energia (gyorsító-feszültségnek megfelelő) 1/3-ad része.

A 6.2 fejezetben említett további kalibrációk elvégzése [10]-ben olvasható, ezek eredményeit (2. táblázat) felhasználva a 32. ábrán látható fantom egy szeletét rekonstruáltam (amelyik a főnyaláb síkjába esett). A fantom plexiből készült, benne két, egymással és a szimmetriatengellyel párhuzamos furat található.



32. ábra
A tesztfantom [10]

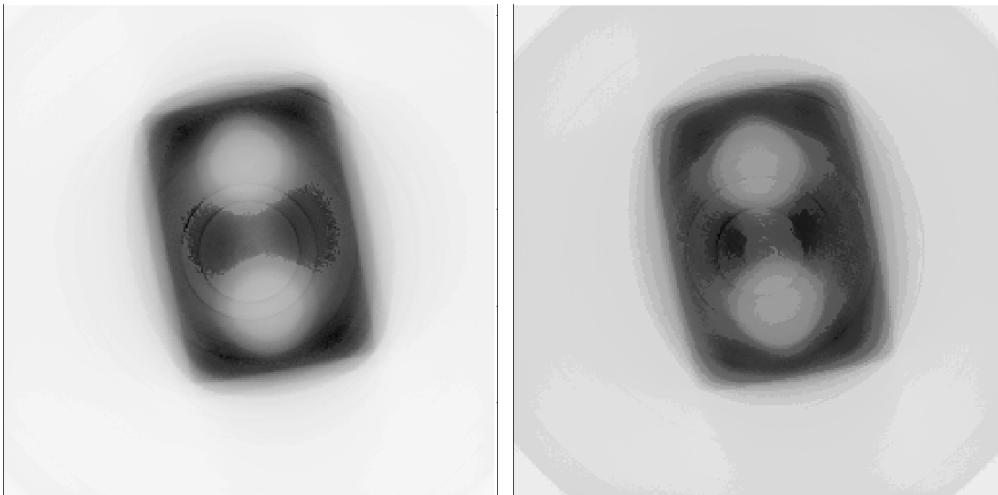
Referencia rekonstrukcióként a Deli G. diplomamunkájában [13] szereplő eredményeket használtam fel. Ez a 33. ábrán balra látható rekonstrukció, ugyancsak GPU-n futó, szűrt visszavetítési algoritmus eredménye. Az általam fejlesztett szoftver, pedig a 33. ábrán jobb oldalt látható képet adta.



33. ábra
Szűrt visszavetítés [13] (balra), ML-EM rekonstrukció (jobbra)

Az két kép összehasonlításából kiindulva több megállapítást is tehetünk. Az első, legszembetűnőbb különbség az ML-EM rekonstrukción a háttér zajmentessége. Másik lényeges különbség, amit el is vártunk az 5. fejezet tesztjei alapján, hogy a kontúrok elmosottak, összességében a jobb oldali kép kevésbé éles. Kontrasztja viszont sokkal jobb, habár ezt rontja a kép közepén megjelenő műtermék. Korábbi tapasztalatok alapján spektrumfelkeményedés okozta műterméknek látszik, de ennek kiderítésére további tesztek szükségesek (spektrummal), ami a TDK-munka időkeretébe már nem fér bele.

Vizsgáltam ugyanezen a fantomon a geometriai paraméterek (ld. 6.3 fejezet) változtatásának hatását. Úgy hajtottam végre a vizsgálatot, hogy a geometriai kalibráció eredményeként kapott paraméterek 10-20%-os környezetében változtattam azokat, és figyeltem a rekonstruált képet. [13]-hez hasonlóan azt a megállapítást tehettem, hogy az U_0 paraméter kivételével a kalibrált paraméter hibahatárán belüli bizonytalanság egyáltalán nincs hatással a képminőségre. A 34. ábrán megfigyelhetjük, hogy a rekonstrukció szellemképes lesz, ha U_0 -t megváltoztatjuk (vagyis eltoljuk a szinogramot).



34. ábra

Rekonstrukció a kalibrált $U_0=802$ paraméterrel (balra), $U_0=830$ esetében (jobbra)

7. Konklúzió, kitekintés

A TDK-munkám során a transzmissziós tomográfias ML-EM képrekonstrukciós algoritmus GPU-n történő implementációját hoztam létre. A dolgozatban ismertettem az ML-EM módszer előnyeit, hátrányait, alkalmazhatóságát, implementációjának lehetőségeit. Több vizsgálatot is elvégeztem a konvergenciára, szabad paramétereinek optimális megválasztására, képminőségre vonatkozólag. A következő lépések között lesz a háromdimenziós rekonstrukció kialakítása, ami lényegében már a forráskód része, de megvalósítása memória-problémákba ütközik. A jövőbeli munkák során a rendelkezésre álló memória jobb kihasználásával nem csak a háromdimenziós képrekonstrukciót kívánom megvalósítani, hanem az előrevetítés gyorsítását is, ami, mint láttuk, a futási időnek meghatározója lehet. A szórás szimulációja, valamint a spektrummal történő rekonstrukció is későbbi munkák része lesz, hiszen ebben az irányban lehet igazán kihasználni az algoritmus pozitív tulajdonságait.

8. Irodalomjegyzék

- [1] K. Lange, R. Carson : "*EM Reconstruction Algorithm for Emission and Transmission Tomography*", **Journal of Computer Assisted Tomography**, 8(2):306-316
- [2] Molnár B. : "*Iteratív ML-EM CT képrekonstrukció GPU-ra*", TDK dolgozat, 2013
- [3] G. Jakab, T. Huszár, B. Csébfalvi: "*Iterative CT Reconstruction on the GPU*", Sixth Hungarian Conference on Computer Graphics and Geometry, Budapest, 2012
- [4] B. De Man : „Iterative Reconstruction for Reduction of Metal Artifacts in Computed Tomography”, 2001
- [5] F. E. Boas, D. Fleischmann : "*CT artifacts: Causes and Reduction Techniques*", **Imaging Med.** (2012) 4(2), 229-240
- [6] Molnár B. : "*Iteratív képrekonstrukciós eljárás gamma tomográfiához: tesztelés és verifikáció*", Szakdolgozat, 2013
- [7] NVIDIA CUDA™, „NVIDIA CUDA C Programming Guide”, 2012
- [8] Guillem Pratx, Lei Xing : „*GPU computing in medical physics: A review*”, **Medical Physics** **38 (5)**, May 2011
- [9] Á. Cserkaszký : "*Monte-Carlo based PET image reconstruction on GPU*", Master's Thesis, 2011
- [10] Tölgyesi B. : "*Röntgen CT készülék építése*", Diplomamunka, 2014
- [11] Dufan Wu, Liang Li, , Li Zhang, Yuxiang Xing, Zhiqiang Chen, Yongshun Xiao: „*Geometric calibration of cone-beam CT with a flat-panel detector*”, 2011 IEEE Nuclear Science Symposium Conference Record, MIC I2.M-79
- [12] Resource manual for compliance test parameters of diagnostic x-ray systems, Diagnostic Devices Branch, Division of Enforcement I, Office of Compliance
- [13] Deli G. : "*Általános tomográfiás rekonstrukciós szoftver fejlesztése GPU-ra*", Diplomamunka, 2014