

Monte Carlo módszerek a pozitron emissziós tomográfiában

TDK dolgozat

Gudics Péter

2013.10.14.

Konzulens: Szirmay-Kalos László, IIT

Tartalomjegyzék

1.	A PET vizsgálatok	3
2.	A mérés menete	3
3.	Iteratív rekonstrukció	7
4.	A feladat leszűkítése	7
5.	A rendszermátrix előállítása	8
6.	Monte Carlo algoritmus.....	9
7.	A rekonstrukció menete	11
8.	Direkt kontribúciós változat	14
9.	Nem korrigált változat	17
10.	Korrigált kombinált változat.....	18
11.	Woodcock tracking.....	20
12.	LOR térkép elkenése.....	22
13.	Párhuzamosíthatósági lehetőségek	23
14.	Összefoglalás	23
	Irodalomjegyzék	24

1. A PET vizsgálatok

A PET (Pozitron Emissziós Tomográfia) vizsgálatok célja [RZ07], hogy a szervek működésbeli sajátosságait, például a vérkeringést és anyagcserét ábrázolják. Daganatokat lehet kimutatni vele, illetve agyi károsodások-, és az Alzheimer kór felismerésében is segít.

A vizsgálat úgy történik, hogy a páciens injekció formájában kap izotópokat, amelyeknek rövid a felezési ideje. Ezután a páciensnek körülbelül 1 óra hosszát kell várakoznia, amíg az izotóp felszívódik a szervezetben, végül pedig a henger alakú készülékbe fektetik, ami fél-egy óra alatt elkészíti a kívánt képet.

Az izotóp bomlása során pozitronokat bocsát ki. A pozitron az elektron anti-részecske párja, így amikor a testben lévő elektronokkal kölcsönhatásba lép, akkor a kölcsönhatásban résztvevő elektron, és önmaga megsemmisül (elektron-pozitron annihiláció).

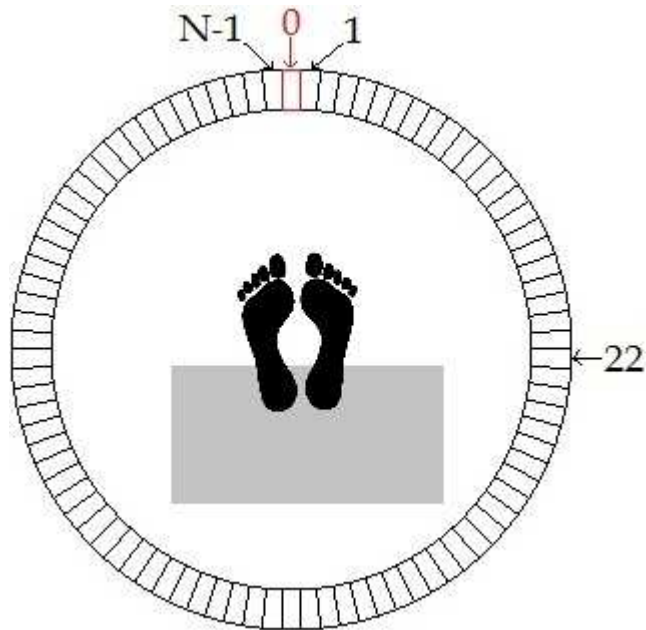
A megsemmisülés következtében két gamma foton keletkezik, és ha az elektron és pozitron sebességkülönbsége relatíve kicsi volt az ütközés pillanatában, akkor az energia és az impulzus megmaradás miatt a fotonok kvázi ellentétes irányban távoznak, 511 keV energiával.

A vizsgálatához használt készülék falán detektorgyűrűk vannak, melyek nagyjából egyidejűleg beérkező fotonpárok esetén továbbítanak egy jelet a képet elkészítő számítógépnek. A detektoroknak sok jellemzője van (koincidencia szélesség, holtidő, érzékenység, stb.), de az algoritmus szempontjából csak a becsapódó elektronok detektálásának alsó energiaküszöbe érdekes.

2. A mérés menete

Az algoritmus bevezetése előtt néhány elnevezéssel, konvencióval, és a mérés által előállított adathalmaz jellegével meg kell megismerkednünk.

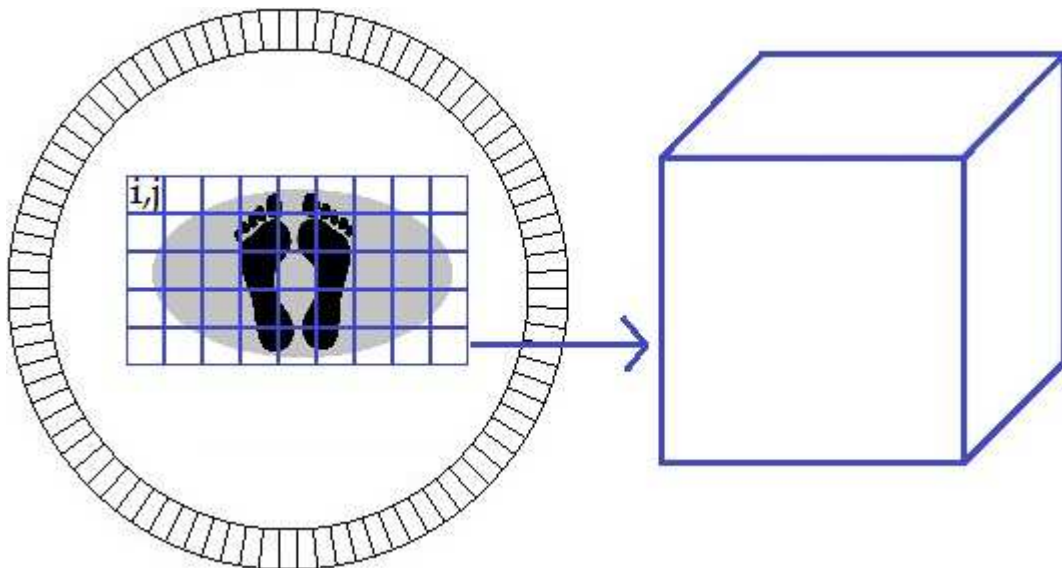
A körgyűrűkön elhelyezkedő detektorokat 0-tól N-1ig számozzuk, ahol a 0. egy tetszőleges referencia, és hozzá képest az óra járása szerinti irányba növekedik a számozás.



1. ábra. A detektorok számozása egy példa körgyűrűn.

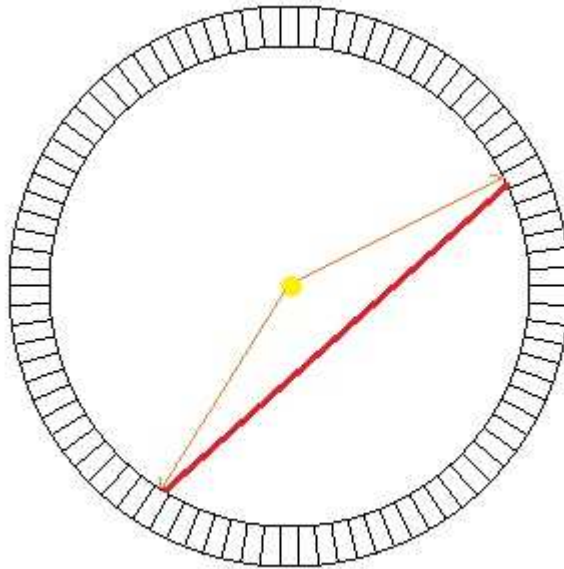
A detektorokat a körgyűrűkön egy referenciához képest, az óra járásával egyezően számozzuk.

A számítógép véges felbontással dolgozik, ezért a testet sok virtuális, egyenlő méterű kockára osztjuk fel, melyeket voxelnek (volume element) nevezünk. Egy voxel a legkisebb térfogati egység, mely a későbbi számításokban részt fog venni.



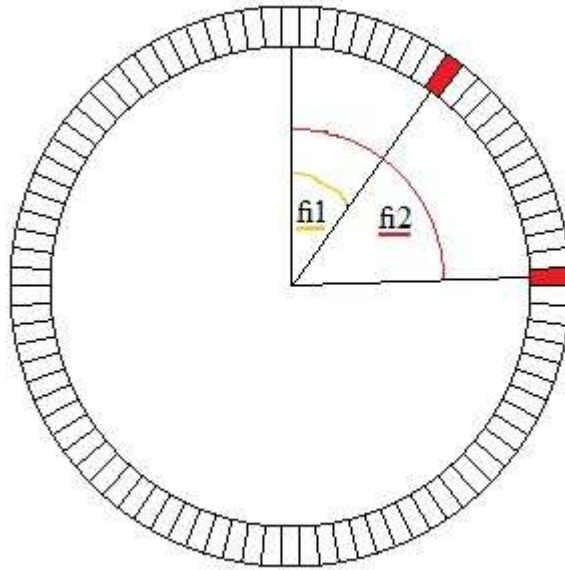
2. ábra, a térfogat virtuális felosztása, a voxel.

A mérés során a becsapódó elektronpárok detektorait összekötő egyeneseket válaszegyeneseknek (LOR – line of response) nevezzük. A LOR-okhoz egy számot rendelünk, aszerint hogy melyik két detektort találták el. Ezt sokféleképpen megtehetnénk, mi a detektorok irányának egy referenciáirányhoz képesti szöge alapján választottunk indexet.



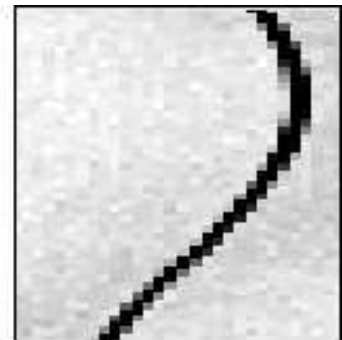
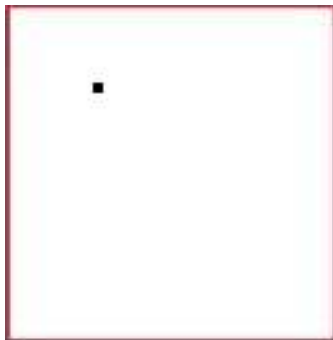
3. ábra. A válaszegyenes, azaz LOR.

A sárga pontban történt annihiláció két fotonja a barnával jelzett utak végén csapódik be, melyeket a piros válaszegyenes köti össze.



4. ábra. A két detektor mely a fotonpárt felfogta (pirossal), és a két szög, melyek közül a kisebb lesz a f_1 , a kettő különbsége pedig $d\phi$.

A mérést követően megfigyelési síkonként az összes voxelen megszámloljuk, hogy hány LOR megy keresztül, és erről egy szinogramot készítünk.



5. ábra. A bal oldalt az eredeti aktivitás térkép, melyen egyetlen voxelen sok bomlás történt. Jobb oldalt látható a mérés végén, a projekció után kapott szinogram. Egy voxelen történt béta bomlások a mérési területen való elhelyezkedésétől függő szinuszos görbét fog eredményezni a szinogramon. Ez jellemzi, hogy az egyes LOR-okhoz mekkora volt a kontribúciója. A sötét szín nagy kontribúciót jelent, a szürke részek a szóródásból származó "zaj".

Éles rendszerekben egy-egy körgyűrű nem csak az ő síkjában történő ütközéseket jegyezheti fel, hanem különböző gyűrűk között is. Ez azt jelenti, hogy minden felvenni kívánt síkhoz tartozik egy szinogram, tehát ha n detektorgyűrűnk van, akkor mindegyikhez tartozik 1 ilyen sík, illetve $n-1$ darab gyűrűpár közötti síkot tudunk még felvenni. Ez összesen $2n-1$ szinogramot eredményez.

3. Iteratív rekonstrukció

Az iteratív rekonstrukció folyamán a képről egy előzetes becslést csinálunk, majd ennek a vetítését összehasonlítjuk a mért adatok vetítettjével. Ha eltérés van a becslésünk és a mért adatok vetítettje között, akkor korrekciós eljárásokat alkalmazunk a különbség csökkentésére, és a következő iterációban vizsgáljuk az újabb becslésünk konvergenciáját. Ezt addig ismételjük, amíg egy bizonyos elfogadható hibahatár alá nem jutunk. A megfigyelt érték maximalizálását megcélzó stratégiát MLEM - Maximum Likelihood Expectation Maximization-nek hívják [SV82].

Összehasonlításra az úgynevezett előre vetítést (forward projection) használjuk, ahol minden LOR-ra meghatározzuk a beütésszám várható értékét egy adott aktivitás becslés alapján:

$$y_{LORL} = \sum_{V=1}^{NVoxels} A_{LV} * x_V$$

Az y vektor L . eleme az L . LOR várható beütésszám értéke, ahol x_V a V -edik voxel aktivitása, és A_{LV} az a valószínűség, hogy az L -edik LOR átmegy a V -edik voxelen.

Legyen a mért adatok projekciója Y , és a becslés projekciója y . Az összehasonlítás négyzetes hibával történik.

$$Hiba = \sum_{L=1}^{NLors} (Y_L - y_L)^2$$

A korrekció a visszavetítés (back-projection) során történik. Itt az n . becslésünket korrigáljuk a mért érték (Y), és az általunk becsült várható érték (y) arányával.

$$x_V^{(n+1)} = x_V^{(n)} * \frac{\sum_L A_{LV} * \frac{Y_L}{y_L}}{\sum_L A_{LV}}$$

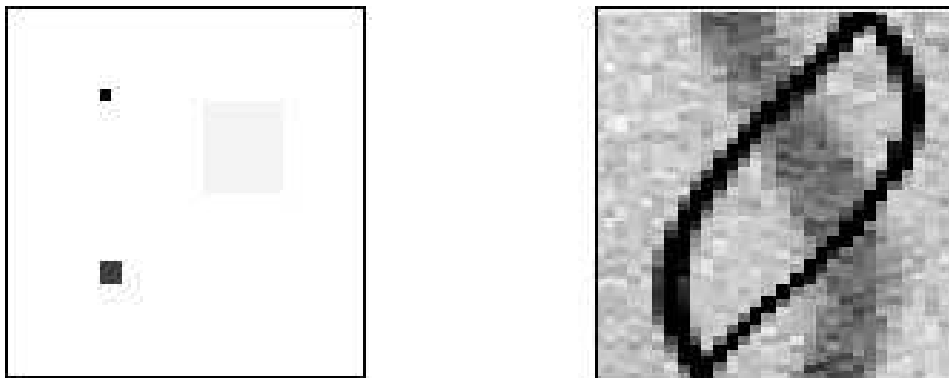
Az A_{LV} a rendszermatrix, mely az összes A_{LV} valószínűséget tartalmazza, hogy az L . LOR átmegy a V -edik voxelen. Előállításával egy későbbi fejezetben foglalkozom.

4. A feladat leszűkítése

n detektorgyűrűt használva akár $2n-1$ szinogram is lehet a mérés utáni kimenet. Ekkora adathalmazból a háromdimenziós rekonstrukció akár napokat is igénybe vehet, mely nem csekély idő. Célunk a rekonstrukciós algoritmus változatai közötti pontosság és jellegzetességek felderítése volt. Az algoritmus jellegzetességei miatt a feladat két

dimenzióban is megegyezik azzal, mint ami három dimenzióban lenne, ezért két dimenziót, és egyetlen detektorgyűrűt használtunk a szimulációhoz. Így egy-egy eredmény elkészítése ritkán tartott egy óránál tovább. A detektorgyűrűn 90 darab detektorral, illetve a mérni kívánt területen 32×32 voxelrel dolgoztunk.

A kiindulási állapotban a bomlásokat egy voxeltérképen az aktivitásukkal jellemezzük. Az aktivitás az egy egységnyi térfogatra vonatkoztatott bomlások száma másodpercenként. A voxelek mérete állítható a szimuláció elején, ezért később az aktivitást fel kell arányosítani vele. A szimulációban a következő 32×32 -es térkép használjuk a voxelek aktivitásának megadásához:



6. ábra. A szimulációhoz használt 32×32 voxeles adathalmaz (balra), és a belőle készített szinogram (jobbra). Az adathalmazon a bal felső voxel a legnagyobb, 64 aktivitású, alatta a kisebb négyzet 16, végül a leghalványabb négyzetnek 1 az aktivitása.

5. A rendszermátrix előállítása

A rendszermátrix, melyet A_{LV} -vel jelölünk, tartalmazza azt az információt, hogy az egyes LOR-ok mekkora valószínűséggel mennek át a voxeleken. Ez függ a mérni kívánt médium sűrűség-eloszlásától, hiszen különböző sűrűségű anyagokban más a mértéke a fotonok szóródásának, illetve abszorpciójának. Az emberi testben (azaz vízben) 511 keV-os foton energiaszinten a foton-anyag interakció 95%-ka szóródás, míg 5% abszorpció, így inkább a szóródás érdemel figyelmet.

A rendszermátrix feltöltéséhez a következő integrál kell kiszámítanunk:

$$A_{LV} = \frac{1}{V} \int_V T(v \rightarrow L) dl$$

Szemléletes jelentése, hogy egy V voxel tartományból kiinduló összes lehetséges fotonút hányad része jut el az L LOR-ba. Ez egy nagy dimenziójú integrált eredményez, mivel egy kiindulási pontot 3 dimenzióban határozunk meg, ezután 2 szöggel mondjuk meg melyik irányba fog elindulni, ami már 5 dimenzió, és minden egyes szóródás alkalmával 3-al nő a

dimenziók száma. A hagyományos integrál kvadratúrák esetén az adott hibaküszöb eléréséhez szükséges minták száma exponenciálisan nő a dimenziók számával. Ennél egy jobb megközelítés a Monte Carlo algoritmus, ahol véletlen mintákat véve a kvadratúra csak a minták számának a négyzetgyökével fordítottan arányos, mégpedig a dimenziótól függetlenül.

6. Monte Carlo algoritmus

Az algoritmus során minden voxelből egy megadott számú fotonpárt véletlen paraméterekkel indítunk, és végigkövetjük útjukat, amíg el nem nyelődnek, vagy a detektorokba be nem csapódnak [Jea04]. Ha mindkét fotonpár eléri a detektorokig, akkor kiszámítjuk hogy melyik LOR-hoz tartoznak, majd a rendszermátrix megfelelő elemébe elmentjük a kontribúciójukat, mely a foton energiája osztva a minták számával. Nézzük meg az algoritmus lépéseit:

- Véletlen kezdőpont sorsolása
A szimulációban a voxelnek nekünk két dimenzióban egy négyzethálós rácsot jelentenek. A bomlás kezdőpontját a voxel belsejében véletlen X és Y koordinátaként sorsoljuk ki.
- Véletlen irány sorsolása
Az első fotonnak kisorsoljuk melyik irányba fog menni. Ez két dimenzióban a $[0, 2\pi)$ intervallumot jelenti. A másik foton a párból ehhez képest ellentétes irányba fog menni.
- Szabad úthossz kiértékelése
Ahogy a foton halad a testen, előbb-utóbb ütközni fog a testen belüli részecskékkel. Hogy megállapítsuk ez mikor fog megtörténni, a következő egyenletet kell kiértékelnünk:

$$P(S) = 1 - \exp\left(-\int_0^S \sigma_t(s) ds\right)$$

ahol S a szabad úthossz, σ_t pedig a szóródás és az abszorpció tényezőinek összege. Az egyenletet átrendezve:

$$-\log(1 - r) = \int_0^S \sigma_t(s) ds$$

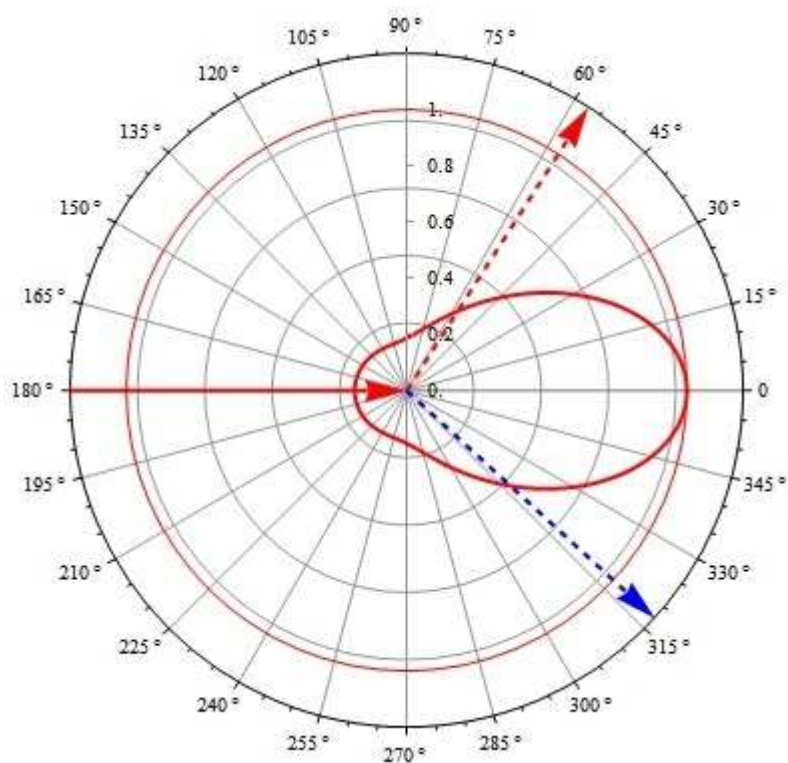
ahol r egy $[0, 1)$ között választott véletlen szám lesz. Ezt az integrált téglány szabállyal számoljuk ki úgy, hogy ds lépésenként összegzünk, amíg az integrál értékét el nem érjük.

$$-\log(1 - r) > ds \sum_i^N \sigma_t(s_i)$$

Itt megjegyzendő, hogy σ_t értéke mindig az aktuális voxelből kerül ki, ahol éppen a foton jár.

- Szóródás, abszorpció megállapítása

Ha a szabad úthossz alatt a foton nem ért ki a testből, akkor szóródni, vagy elnyelődni fog. Ez több változással is jár. Először is a végső kontribúciójának a súlya csökken. Kezdetben 1 súlyról minden szabad úthossz végén $\sigma_s / (\sigma_s + \sigma_a)$ részére csökken, ahol σ_s a szóródás, σ_a az abszorpció esélye. Ha elnyelődött akkor a teljes energiáját elveszti, és nem "követjük tovább". Ha szóródott, akkor csökken az energiája, és irányt vált. Az irányt a Klein-Nishina formula, más néven Khan algoritmus mondja meg, az energia változását pedig a Compton szóródás. Ezt a továbbiakban részletesen nem tárgyalom, a <http://demonstrations.wolfram.com/KleinNishinaFormulaForComptonEffect/> oldal részletesebb leírást tartalmaz róla. A Khan algoritmus két tulajdonsága, hogy szóródás után a foton legnagyobb valószínűséggel nem vált irányt, illetve az energia csökkenése arányban van az irány változásával.



7. ábra. Klein-Nishina fázis függvény. A szóródás utáni irány legnagyobb valószínűséggel változatlan.

A lépéseket addig ismételjük, amíg a fotonok nem nyelődnek el, vagy a detektorokba csapódnak.

7. A rekonstrukció menete

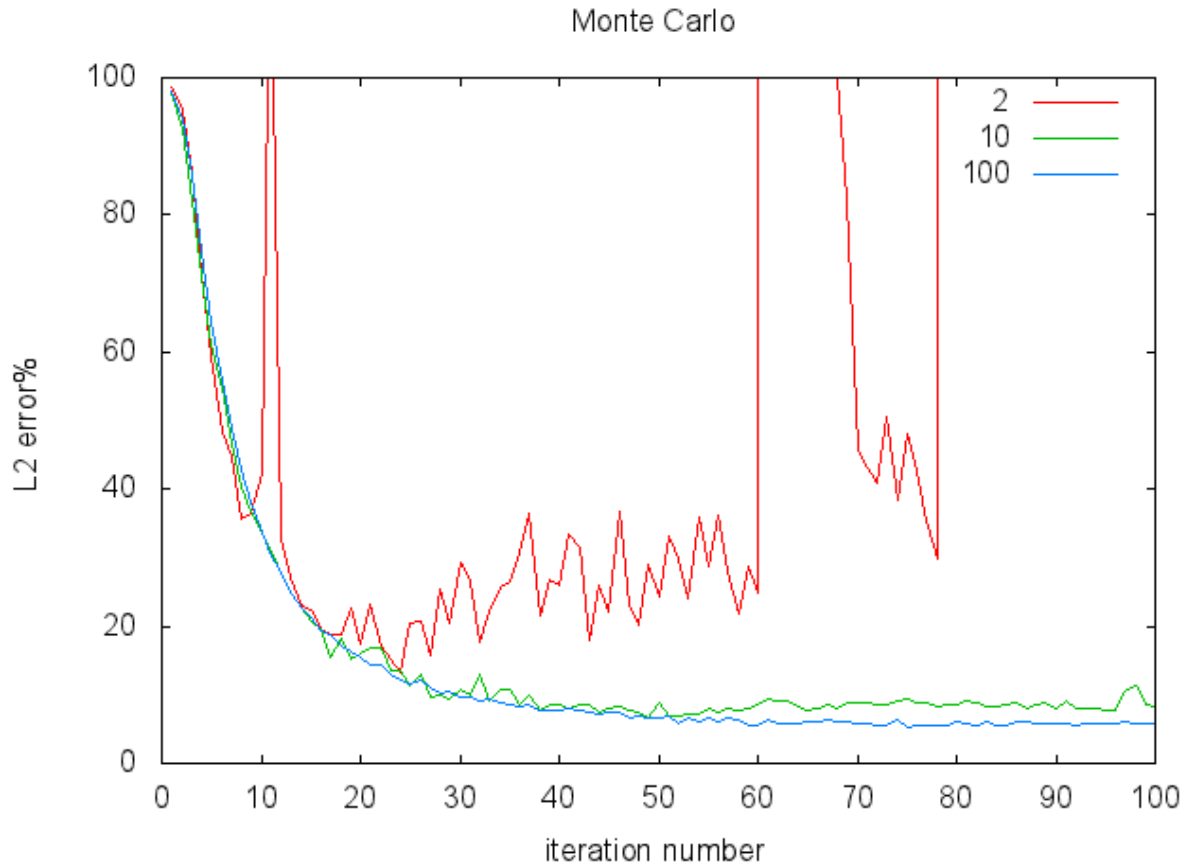
Most hogy értjük a rekonstrukció összes részletét, foglaljuk össze a sorrendet:

0. A szimulációban használt méréshez nagy mintaszámú rendszermátrix betöltése.
1. A voxeltérképből mért adat elkészítése előre vetítéssel, a nagy mintaszámú rendszermátrixot használva. Ez lesz Y .
2. Előzetes becslés elkészítése (például az összes találatot egyenletes eloszlással szétosztani y -ra)
3. Rendszermátrix (újra)számolása Monte Carlo módszerrel
4. Becslés előrevetítése a rendszermátrixszal.
5. Becslés visszavetítése (korrekció) a rendszermátrixszal.
6. A hiba mérése.
7. 3-6. ismétlése 100-szor.

Megjegyzések:

A 0. lépésre azért van szükségünk, mivel mi szimulációt írtunk, a gyakorlatban valódi méréssel történik az Y meghatározása. A nagy mintaszám a mi szimulációnkban 200.000 mintát jelent, ennek a rendszermátrixnak az előállításához körülbelül egy órát vesz igénybe a számítógép, míg a későbbi kevesebb mintaszámú rendszermátrix 1000 mintával is csak 10 percet vesz igénybe. A gyakorlatban a mérés mintaszámát a sugárdózis nagysága határozza meg. Bizonyos mértéknél nagyobb a páciens egészségének nem tesz jót, illetve a mérőberendezésnek is van egy küszöbe, aminél többet nem képes mérni.

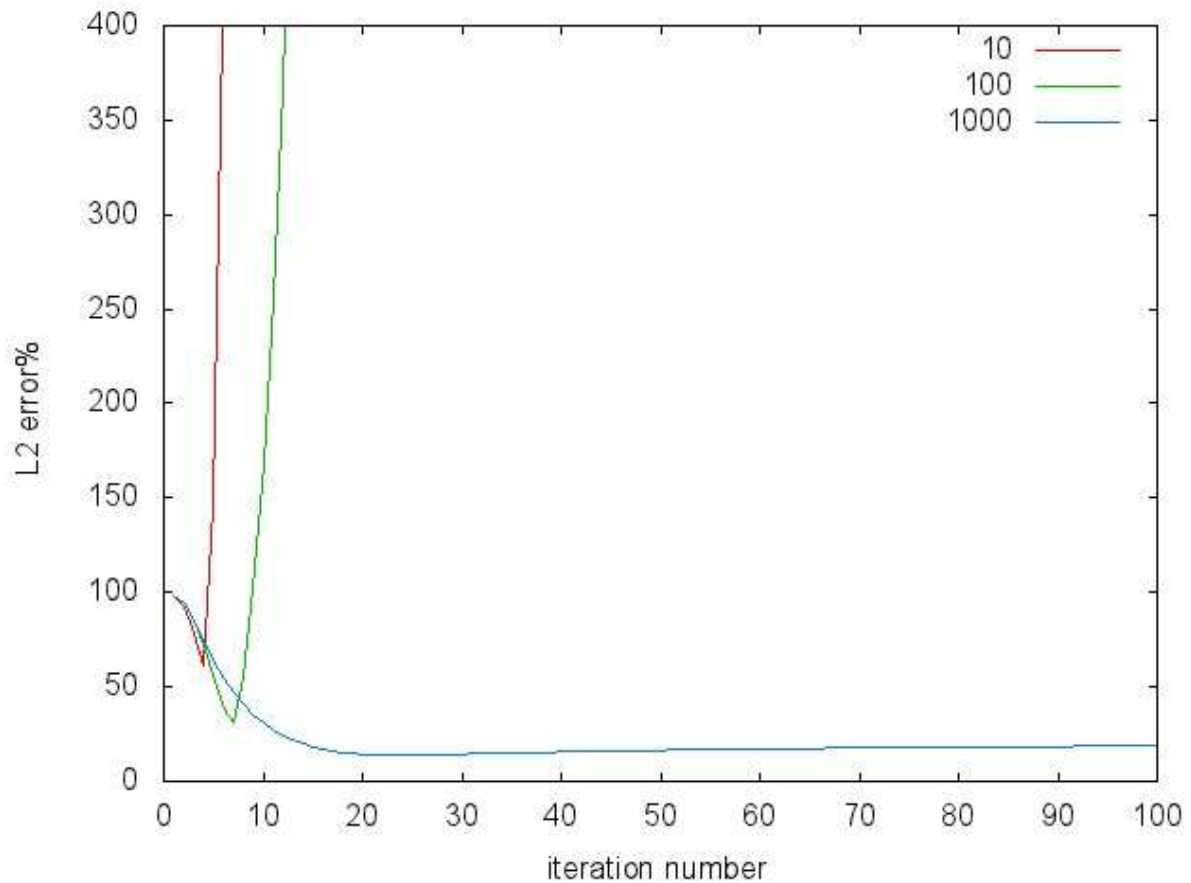
Az alább látható egy rekonstrukció alatt készült hibagörbe, melyet a négyzetes hiba alapján számolunk, amit relatív skálán, százalékos formában adunk meg a referenciára vonatkoztatva. A hibagörbék elkészítéséhez a program egy szöveges fájlba mentette el iterációnként a hiba abszolút értékét, majd az ábrát a gnuplot nevű ingyenes alkalmazással készítettük.



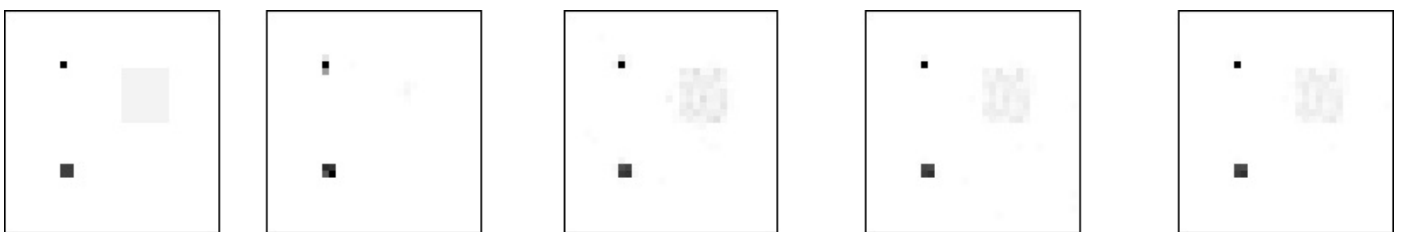
8. ábra. rekonstrukciós hibagörbék különböző mintaszámú Monte Carlo módszerrel számolt rendszermátrixok használatával. A vízszintes tengelyen az iterációk száma, a függőlegesen az hiba látható. A jobb felső sarokban a különböző színű görbékhez tartozó mintaszámok találhatóak.

Látható az ábrán, hogy az eredmény először gyorsan, majd egyre lassabban konvergál, és van egy alsó határa a hibának, ezen ábrán az 5, ami alá nem jut el 100 iteráció alatt. Viszont az algoritmus lépésszáma $O(\text{voxelek száma} \cdot \text{minták száma} \cdot \text{iterációk száma})$, így szeretnénk minimális számú mintát használni a megfelelő képminőséghez. Túl kevés minta esetén, mint ahogy azt a görbén 2 mintával láthatjuk, az algoritmus akár nem is lesz konvergens.

Megjegyzés: Fontos, hogy minden iterációhoz újrageneráljuk a rendszermátrixot, hiszen az egy becslés, amivel alá, vagy fölé fogunk becsülni, és ha ezt 100 iteráción keresztül használjuk, akkor a hiba akkumulálódik. Ha minden iterációban új mátrixot használunk, akkor a számítási hibák kompenzálják egymást. Ipari alkalmazásokban úgyszólván olyan nagy (több petabyte-os) a rendszermátrix mérete a LORok és voxelek száma miatt, hogy nem lehet tárolni, így mindig az éppen szükséges elemét (újra) kiszámítják.



9. ábra. Ugyanannak a rendszermátrixnak az újrafelhasználása az összes iterációban. Az algoritmus 10 és 100 minta használatával rövidesen elszáll. 1000 mintával már stabil, viszont 15 alá nem megy a hiba, ami közel háromszorosa annak, mintha minden iterációban újragenerálnánk a rendszermátrixot, akár csak 100 vagy 10 mintával is.



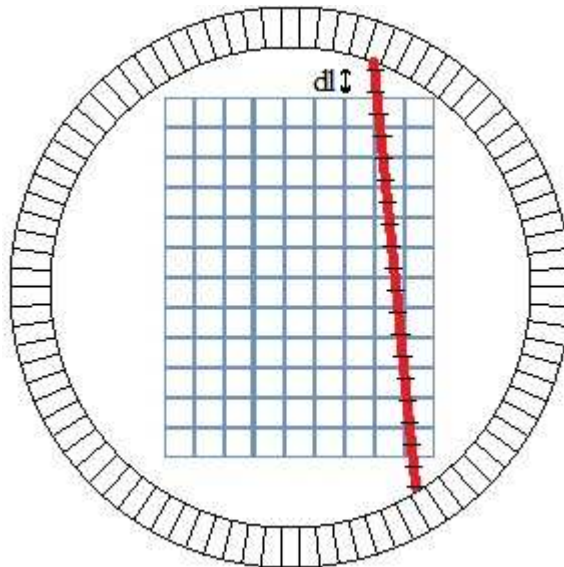
10. ábra, a rekonstrukció eredménye

A rekonstrukció eredménye a 10. ábrán látható. A baloldali képen látható az eredeti bomlástérkép. Tőle jobbra a 2 mintával készített rekonstrukció, ahol a legkisebb aktivitású szürke téglalap majdnem hiányzik. Őket követi a 10, 100 végül 1000 mintával készített rekonstrukció.

8. Direkt kontribúciós változat

Felmerül a kérdés, hogy hogyan lehetne javítani, vagy gyorsítani a Monte Carlo algoritmuson. A direkt (szóródás-mentes) kontribúció kiszámításának az integrálja már nem olyan magas dimenziószámú, így azt tudjuk pontosabban számolni más módon [Jos82, Sid85, MB04]. Innentől kezdve LOR módszerként fogok rá hivatkozni.

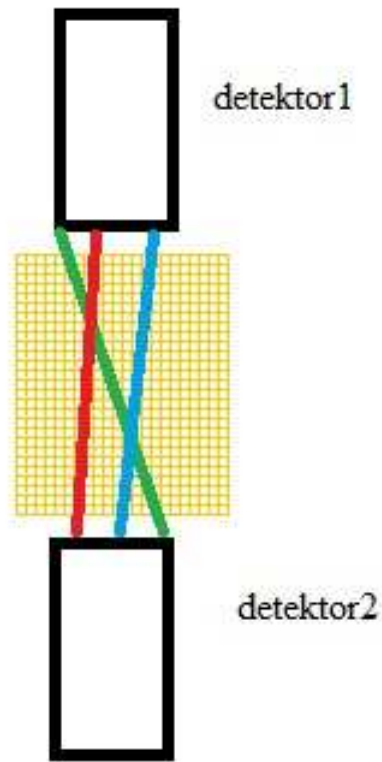
Egy LOR kontribúciója a voxelekre melyeken áthalad a következő egy vonalintegrállal kapható meg, ahol az aktivitást és a kioltási tényezőket kell integrálni.



11. ábra. A LOR menti integrálás a direkt hozzájárulás számításához.

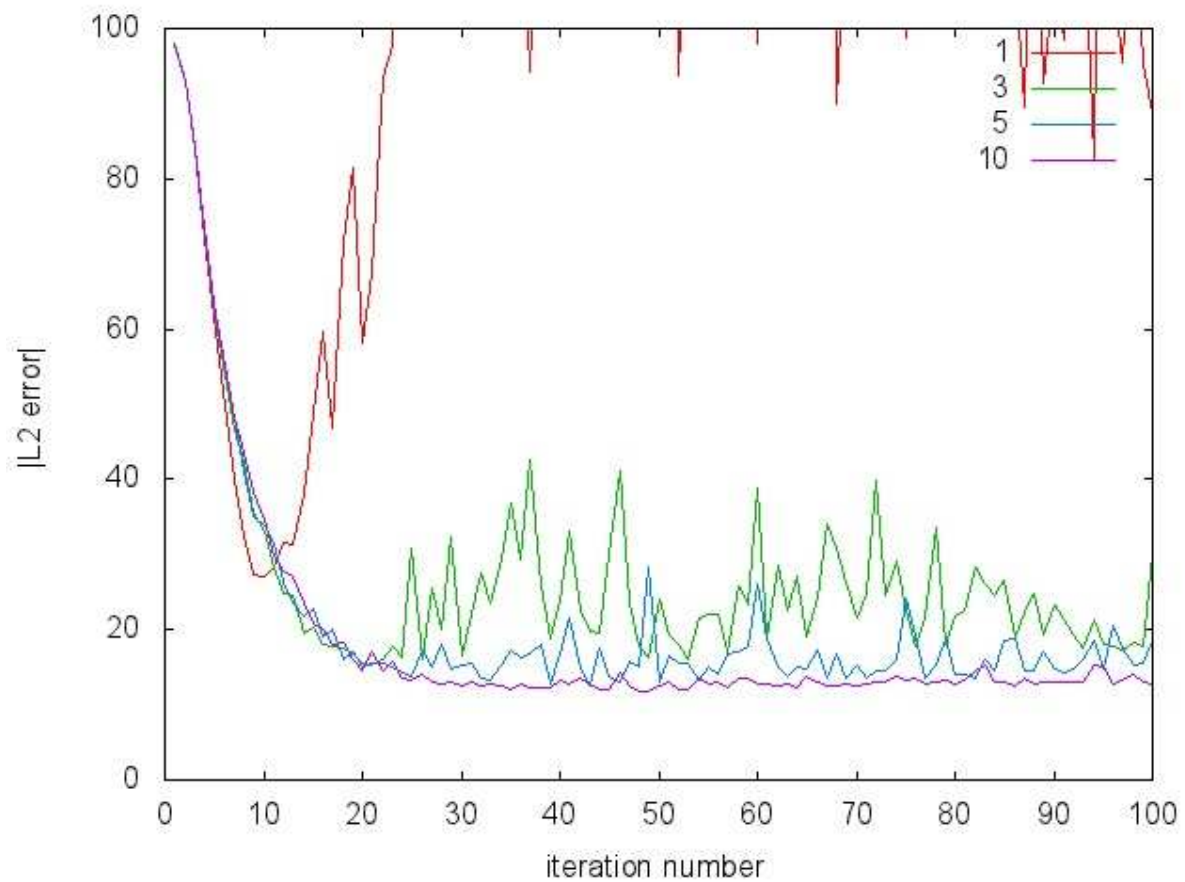
Az aktuális LOR-on dl távolságonként mintavételezzük az aktivitást és a σ_i kioltási tényezőt azon a voxeleken, melyen éppen áthalad, és a teljes l hosszú egyenesen összegezve kapjuk meg az integrálok értékét.

Sajnos itt sem elég egy egyenes becslését alkalmazni, ezért az integrálokat több egyenes súlyozott összegeként számítjuk ki.

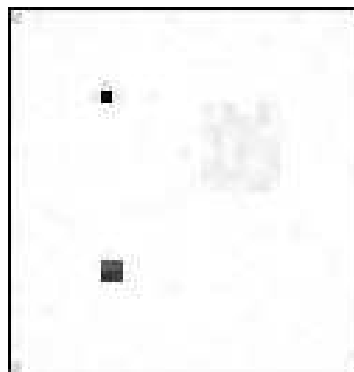


12. ábra. A jobb becslés érdekében a piros, zöld és kék egyenes súlyozott összegét vesszük.

Végül egy mérés 1, 3, 5 illetve 10 mintával:



13. ábra. LOR integrálós módszerrel történő mérés hibagörbéje.

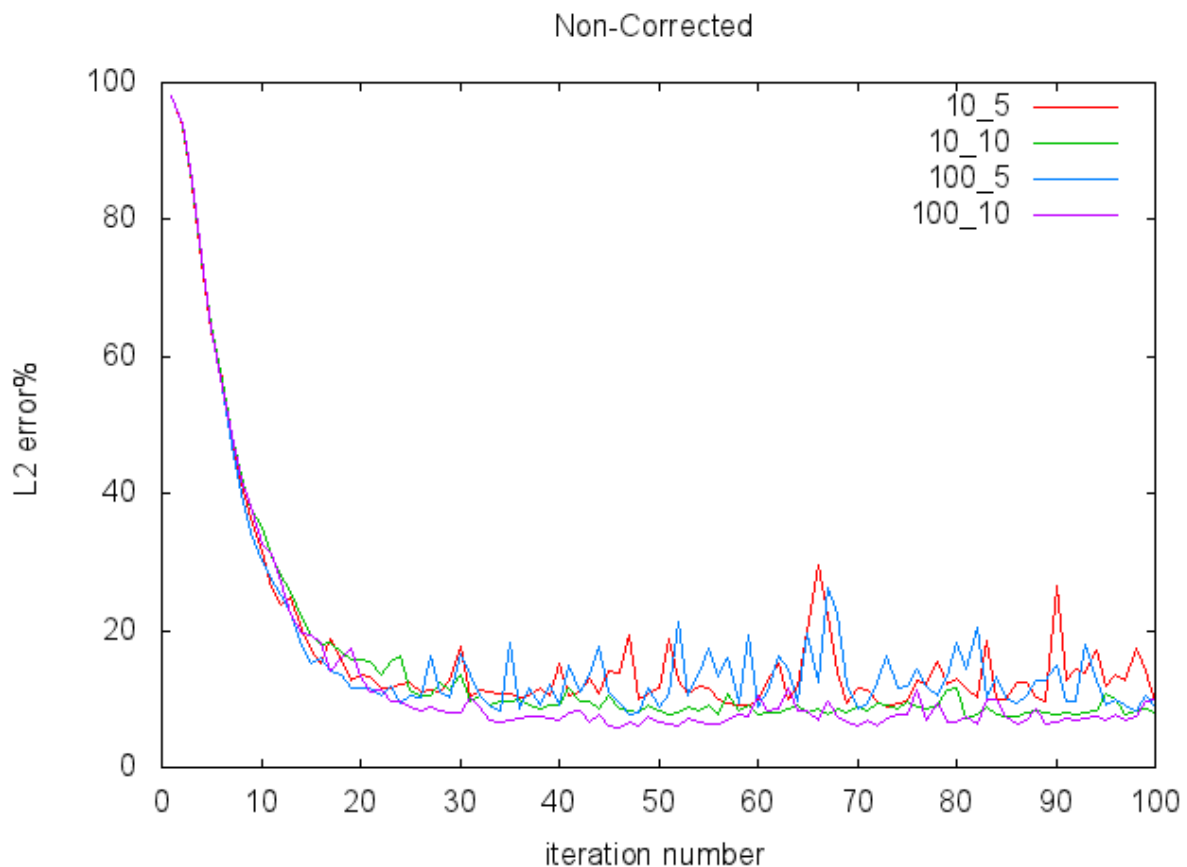


14. ábra. LOR integrálós módszerrel történő rekonstrukció eredménye.

Ez a módszer pontosan számítja a direkt kontribúciót, de rekonstrukcióra önmagában alkalmatlan, ahogy a 14. ábrán is látszik.

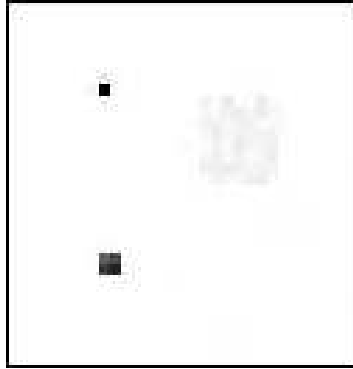
9. Nem korrigált változat

Hogy megőrizzük a Monte Carlo pontosságát, de növeljük a sebességét, egy új ötlethez folyamodunk. Tudjuk hogy a LOR változat a direkt kontribúciót jól számolja, de a végeredmény szempontjából kevésnek bizonyul. Ötvözzük a kettőt, a LOR módszerrel számoljuk a direkt kontribúciót, és a Monte Carlo algoritmussal pedig a szóródó fotonok kontribúcióját. Ehhez csak annyit kell hozzátennünk a Monte Carlo algoritmushoz, hogy számolja a fotonpárok szóródását, és ha egyik foton sem szóródott a párból, akkor eldobjuk őket. Tulajdonképpen a két számítást egymástól függetlenül is megtehetjük, majd végül összeadjuk az eredményeiket. Ezt hívjuk mi "nem korrigált" kombinációnak.



15. ábra. Nem korrigált módszerrel történő mérés hibagörbéje.

A bal felső sarokban először a Monte Carlo által használt mintaszám, majd a LOR módszer által használt mintaszám látható. Bár a tiszta Monte Carlo módszerhez képest többet fluktuál, a 100 Monte Carlo és 10 LOR mintával készült görbe a 40. iteráció körül eléri azt a szintet amit a 100 tiszta Monte Carlo mintával készült csak 100 iteráció után. A nem korrigált módszerben az összes Monte Carlo-val számolt fotonpárnak csak a 60%-a szóródott.



16. ábra. A nem korrigált módszer 100 MC és 10 LOR mintával készült rekonstrukció eredménye.

10. Korrigált kombinált változat

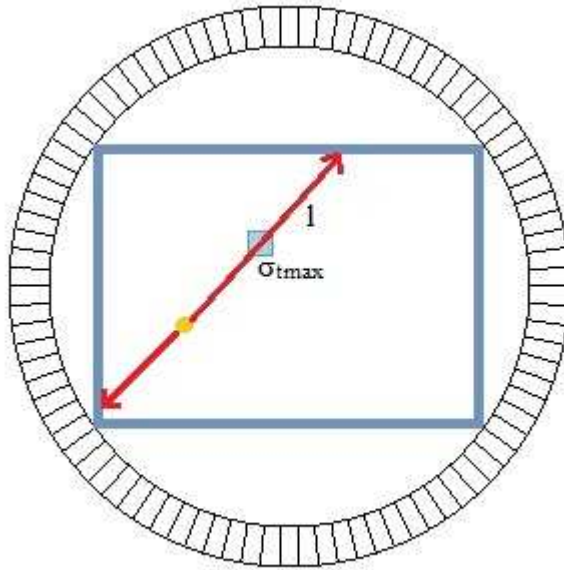
A nem korrigált változatnál a minták egy részét eldobjuk, ami veszteség és a hibát növeli. A korrigált változatban nem dobjuk el a mintákat, hanem súlyozva összegezzük, vigyázva arra, hogy az eredmény várható értékben korrekt legyen (torzítatlan becslés). Ehhez figyelembe vesszük, hogy a két módszer által generált minták sűrűsége különböző, és mindkét algoritmusnál az eddig számolt kontribúciót az összes résztvevő módszer sűrűségével kompenzáljuk [VG95].

A korrigált kombinált változat megoldotta az több módszer optimális kombinációját. Viszont a közegben a szóródás mértékétől függően sok-sok processzoridő elveszhet a Monte Carlo algoritmusban olyan fotonok követésére, melyek végül nem fognak szóródni. A korrigált változat lényege, hogy a Monte Carlo algoritmusban az r véletlen szám értékét úgy módosítjuk, hogyha az első foton nem szóródott, akkor a másodikkal ez mindenképp megtörténjen, így ténylegesen szóródó fotonpárokat fogunk kapni.

Ehhez a fotonpár első fotonjánál ki kell számolnunk, hogy mi a szabad úthossza. Ehhez a következő összefüggést értékeljük ki:

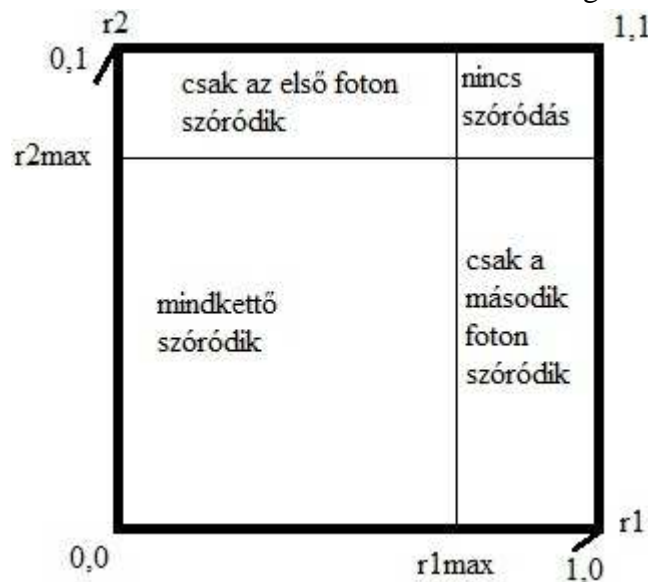
$$r_{max} = 1 - \exp(-\sigma_{tmax} * l)$$

ahol σ_{tmax} az útközben előforduló legnagyobb sűrűség, és l a testben megtehető leghosszabb egyenes szakasz. Fontos észrevenni számunkra, hogy r_{max} is a $[0,1]$ intervallumból kerül ki.



17. ábra. Az ütközés kieröltetése a mintavételezés során

A sárga voxelben történt bomlás után az első foton kilövési irányában kiszámoljuk l hosszát, majd megkeressük a legnagyobb σ_r -vel rendelkező voxel, hogy ki tudjuk számítani $r_{1\max}$ -ot. Ezután sorsolunk egy r $[0,1]$ véletlen számot, majd megvizsgáljuk, hogy r kisebb-e mint $r_{1\max}$. Ha kisebb, akkor biztos, hogy a fotonpár első tagja szóródni fog. Ha nem kisebb, akkor a második fotonnál fogunk korrigálni úgy, hogy kiszámítjuk $r_{2\max}$ -ot az előbb leírt módon, majd miután új r -t sorsoltunk, azt megszorozzuk $r_{2\max}$ -al, így biztosítva, hogy biztosan szóródni fog l távon belül. A következő ábra szemlélteti a lehetőségeket:

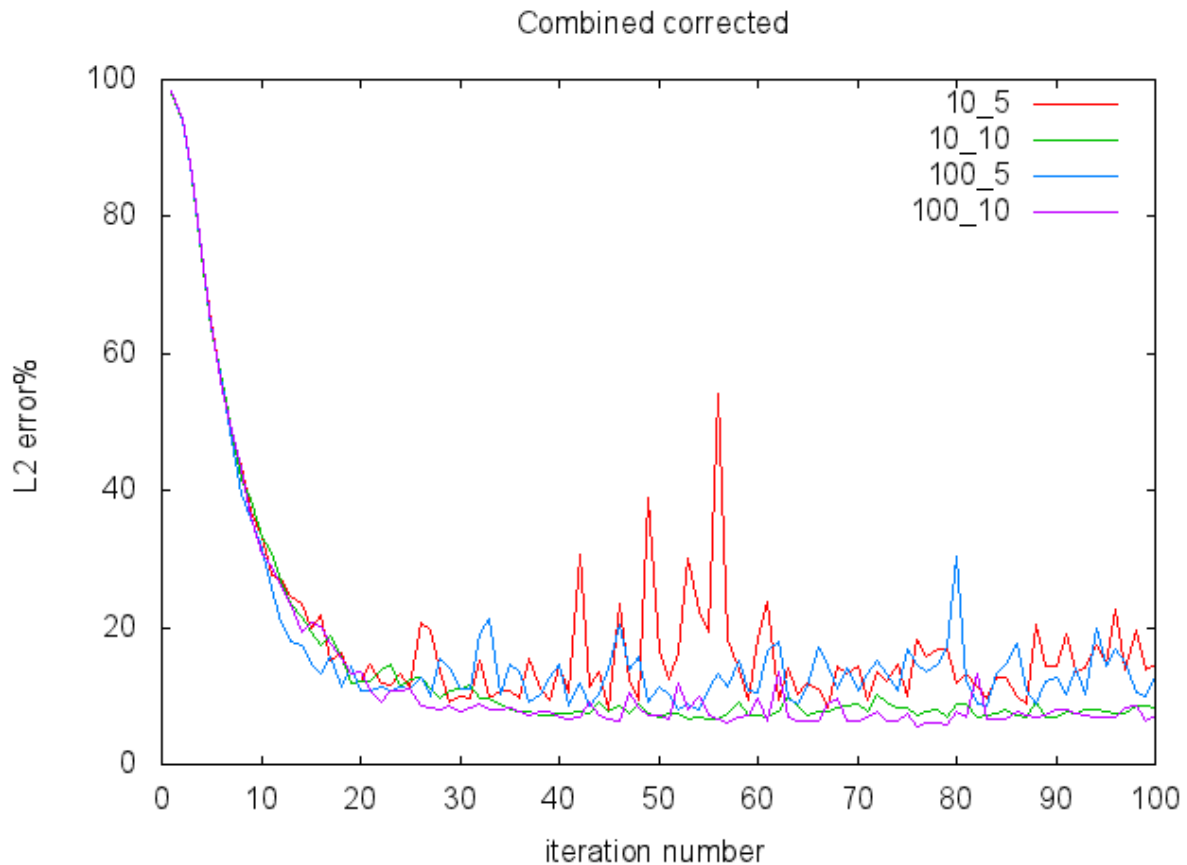


18. ábra. Az ütközés kieröltetéséhez tartozó döntési térkép.

A vízszintes tengelyen van az első fotonnak sorsolt r_1 , a függőlegesen a második fotonnak sorsolt r_2 . Ha első sorsolásnál $r_1 < r_{1\max}$, akkor r_2 értéke mindegy, mivel vagy a "mindkettő szóródik" vagy az "első szóródik" régióba fog esni. Ha első sorsolásra $r_1 > r_{1\max}$, akkor r_2

sorsolásakor $r_{2_{\max}}$ -al való szorzás a $[0, r_{2_{\max}}]$ tartományra szűkíti r_2 értékészletét, így mindenképp a "második szóródik" régióba fog esni. Így teljesül a célunk, hogy legalább az egyik foton szóródjon.

Persze ha a vizsgált területünk négyzetrácsának valamelyik sarkához nagyon közel van a bomlás helye, akkor lehetséges hogy a korrigálásunk ellenére sem fognak szóródni a fotonok, ezeket mikor beírnánk kontribúció szerint a rendszermátrixba, akkor eldobjuk. Az ilyen fotonpárok aránya az összeshez elenyésző.



19. ábra. A kombinált korrigált szimuláció hibagörbéje

A Monte Carloval számolt fotonok 96%-a szóródott, tehát a módszer jól működik. Ez az arányú szóródás túl magas, és hatalmas tüskéket okoz.

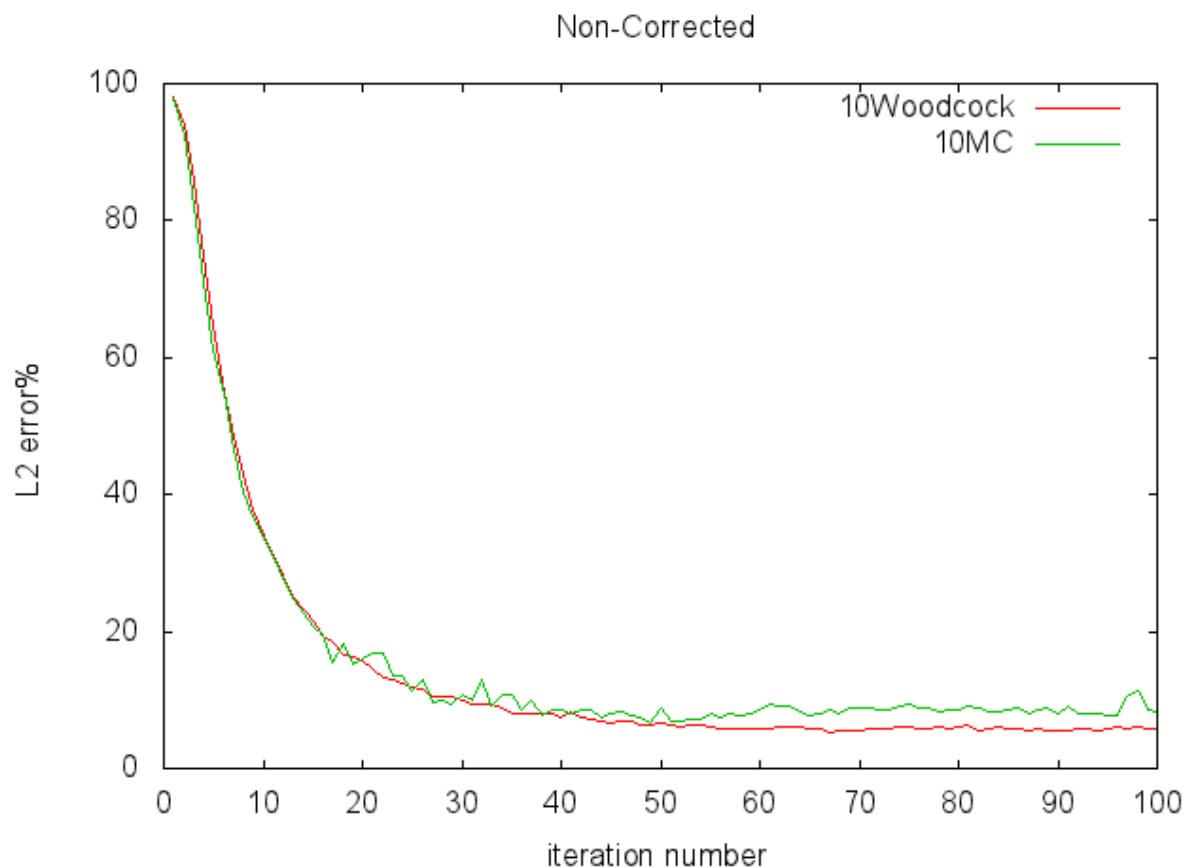
11. Woodcock tracking

A Woodcock tracking a Monte Carlo szabad úthossz sorsolás egy változata, mely a szabad úthossz számítását egyszerűsíti a következő ötlet szerint [SzT11]:

A mérni kívánt közeget töltsük fel úgy "virtuális részecskékkel", hogy mindenhol egyenletesen σ_{max} sűrűséggel legyenek részecskék, ahol σ_{max} egy felső becslése a mérni kívánt közeg legnagyobb sűrűségének.

A szimuláció lépésköze legyen $ds = \log(1 - r)/\sigma_{max}$, és ha ütközés történik, akkor σ_i/σ_{max} valószínűséggel ütközünk "valós" részecskének, és akkor az energia, illetve szóródási irányt számolunk, illetve $1 - \sigma_i/\sigma_{max}$ valószínűséggel ütközünk virtuális részecskének, mely nem változtat se az irányon, se az energián. Ezt akár úgy is értelmezhetjük, hogy az energiát az eggyel szorozzuk, és az irányt pedig egy Dirac-deltával.

Ezt a módszert használva számos voxel kiolvasását megspórolhatjuk, mely sokat számít, ha masszívan párhuzamosítjuk az algoritmust, például Cuda vagy OpenCL környezetben, melyekről később említést teszek. A módszer gyengepontja σ_{max} felső becslése lehet, ha például a testben egy apró régióknak kiugró σ_i értéke van a többihez képest (például valamilyen fém implantátum az agyban, vagy csavar valamilyik porcban), akkor környezetéhez viszonyítva nagyon sűrű lesz a virtuális közeg, és nagyon apró lépésközzel fog futni a szimuláció, ami nagyobb lépésszámot eredményez.



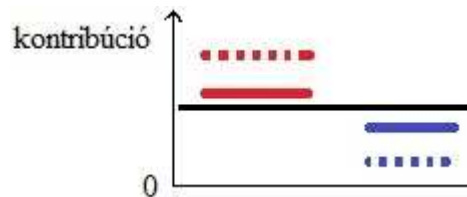
20. ábra. Woodcock összehasonlítása tiszta Monte Carlo módszerrel.

A viszonylag homogén mérési adatok miatt 10 másodperccel gyorsabb volt, és 1%-kal pontosabb, ami az ábrán nagy különbségnek tűnik, de annyira nem forradalmi.

12. LOR térkép elkenése

Képzeljünk el két LOR-t, mindkettő D1 detektorból indul, de az első D2-ben végződik, a második pedig D2 szomszédjában. Nagyon hasonló utat tesznek meg, viszont mi külön becsüljük meg őket, például az elsőt alul, míg a másodikat pedig felül. Ha a véletlen hiba nagyságrendileg nagyobb, mint a két szomszédos LOR közötti kontribúció különbsége, akkor ha átlagolnánk őket, akkor mindkettő LOR becslése közelebb kerülne a valós kontribúcióhoz.

Illusztrálva a kontribúció értékével:



21. ábra. A LOR térkép elkenésének szemléltetése.

Tegyük fel, hogy a képen látható piros az egyik LOR kontribúciója, a szaggatott vonallal jelzett pedig a becslés amit adtunk rá. A másik LOR kék színnel, becslése szaggatott vonallal. A kettejük között lévő fekete vonal az átlaguk, ami mindkét esetben közelebb áll a valós kontribúciójukhoz, mint a becslöttjük.

Az algoritmus szintjén, amikor kiszámoljuk a végső kontribúciót, akkor mindkét detektor oldalán a szomszédokhoz is hozzáadjuk azt, pontosabban mindkét oldalon 3-3 detektorról van szó, ami összesen 9 LOR-t jelent, így mindegyik LOR a kontribúció kilencedét kapja meg.

13. Párhuzamosíthatósági lehetőségek

Mivel a minták egymástól függetlenek, így az algoritmus lehetőséget ad a fotonutak számításának a párhuzamosítására. Ezt meg lehet valósítani Nvidia Cuda, vagy OpenCL-el is, melyek egy kiegészítésként szolgálnak a C++ nyelvhez. A sebesség szempontjából a voxel olvasások kritikusak. Ha tudnánk tárolni a rendszermátrixot, akkor a kontribúció visszaírásánál kellene még a konzisztenciára ügyelni, atomi író műveletek használatával.

14. Összefoglalás

A dolgozat a Pozitron Emissziós Tomográfias rekonstrukció direkt Monte Carlo szimulációra építő megoldását tárgyalta. A Monte Carlo eljárások fő problémája a szórás, illetve a zaj, amelyet a véletlen becslés okoz. A dolgozatban sokféle szóráscsökkentő eljárást vizsgáltunk meg a PET rekonstrukció során, és megmutattuk, hogy ezek alkalmazásával a pontosság jelentősen javítható a mintaszám és így a számítási idő növekedése nélkül.

A dolgozat célja éppen ezen szóráscsökkentő eljárások összehasonlítása volt egy egyszerűbb, 2D-s környezetben, ahol a kísérletek elvégzése gyorsan végrehajtható volt. A további lépés ezen algoritmusoknak a teljes 3D-s GPU alapú rendszerbe való beépítése lesz, amelyet az IIT-n fejlesztenek a Mediso Kft. számára [Mag10].

Irodalomjegyzék

- [Mag10] M. Magdics et al. Tera-Tomo project: a fully 3D GPU based reconstruction code for exploiting the imaging capability of the NanoPET/CT system. In *World Molecular Imaging Congress*, 2010.
- [GMDH08] N. Gac, S. Mancini, M. Desvignes, and D. Houzet. High speed 3D tomography on CPU, GPU, and FPGA. *EURASIP Journal on Embedded Systems*, 2008. ID 930250.
- [Jea04] S. Jan and et al. GATE: A simulation toolkit for PET and SPECT. *Phys. Med. Biol.*, 49(19):4543–4561, 2004.
- [Jos82] P.M. Joseph. An improved algorithm for reprojecting rays through pixel images. *IEEE Trans. on Med. Imaging*, 1(3):192–196, nov. 1982.
- [KY11] K. S. Kim and J. C. Ye. Fully 3D iterative scatter-corrected OSEM for HRRT PET using GPU. *Phys. Med. Biol.*, 56:4991–5009, 2011.
- [MB04] B. De Man and S. Basu. Distance-driven projection and backprojection in three dimensions. *Phys. Med. Biol.*, 49:2463–2475, 2004.
- [Med10a] Mediso. Anyscan-PET/CT, www.mediso.com/products.php?fid=1,9&pid=73.
- [Med10b] Mediso. Nanoscan-PET/CT, www.mediso.com/products.php?fid=2,11&pid=86.
- [Oll96] J. M. Ollinger. Model-based scatter correction for fully 3D PET. *Phys. Med. Biol.*, 41:153–176, 1996.
- [QLC+98] J. Qi, R. M. Leahy, S. R. Cherry, A. Chatziioannou, and T. H. Farquhar. High-resolution 3D Bayesian image reconstruction using the microPET small-animal scanner. *Phys. Med. Biol.*, 43(4):1001, 1998.
- [RZ07] A. J. Reader and H. Zaidi. Advances in PET image reconstruction. *PET Clinics*, 2(2):173–190, 2007.
- [Sid85] R. L. Siddon. Fast calculation of the exact radiological path for a three-dimensional CT array. *Medical Physics*, 12(2):252–257, 1985.
- [SV82] L. Shepp and Y. Vardi. Maximum likelihood reconstruction for emission tomography. *IEEE Trans. Med. Imaging*, 1:113–122, 1982.
- [SzT11] L. Szirmay-Kalos, B. Tóth, and M. Magdics. Free Path Sampling in High Resolution Inhomogeneous Participating Media. *Computer Graphics Forum*, 30(1):85–97, 2011.
- [TB02] D. W. Townsend and T. Beyer. A combined PET/CT scanner: the path to true image fusion. *British Journal of Radiology*, 75:24–30, 2002.
- [VG95] E. Veach and L. Guibas. Optimally combining sampling techniques for Monte Carlo rendering. In *ACM SIGGRAPH '95 Proceedings*, pages 419–428, 1995.
- [WNC96] C.C. Watson, D. Newport, and M.E. Casey. *Three-Dimensional Image Reconstruction in Radiation and Nuclear Medicine*, chapter A single scattering simulation technique for scatter correction in 3D PET, pages 255–268. Kluwer Academic Publishers, 1996.
- [XM07] F. Xu and K. Mueller. GPU-acceleration of attenuation and scattering compensation in emission computed tomography. In *9th Fully Three-Dimensional Image Reconstruction in Radiology and Nuclear Medicine*, 2007.