

DIPLOMAMUNKA

Általános tomográfiás rekonstrukciós szoftver fejlesztése GPU-ra

Deli Gábor

Témavezető:

Légrády Dávid

egyetemi docens

BME Nukleáris Technikai Intézet

Nukleáris Technika Tanszék

BME

2014

TARTALOM

Önállósági nyilatkozat	3
Bevezetés.....	4
Rendelkezésre álló eszközök.....	5
Fan-beam szinogramkészítés.....	7
Fan-beam rekonstrukció.....	9
Matematikai háttér.....	10
Megvalósítás GPU-n	13
Tesztelés szimulált adatokon.....	16
Mérési adatok rekonstruálása	19
Cone-beam rekonstrukció.....	27
Matematikai háttér.....	27
Megvalósítás GPU-n	30
Mérési adatok rekonstruálása	32
Összefoglalás, kitekintés	38
Köszönetnyilvánítás	39
Irodalomjegyzék.....	40

ÖNÁLLÓSÁGI NYILATKOZAT

Alulírott Deli Gábor, a Budapesti Műszaki és Gazdaságtudományi Egyetem Fizika alapszak (BSc) hallgatója kijelentem, hogy ezt a diplomamunkát meg nem engedett segítség igénybevétele nélkül, saját magam készítettem. Minden olyan szövegrészt, adatot, diagramot, ábrát, vagy bármely más elemet, amelyet vagy szó szerint, vagy azonos értelemben, de átfogalmazva másoktól vettem át, a forrás megadásával egyértelműen megjelöltem.

Budapest, 2014. június 1.

.....

Deli Gábor

BEVEZETÉS

A diplomamunka célja egy olyan GPU-ra írt tomográfiás rekonstrukciós szoftver létrehozása, amellyel rekonstruálhatók többek közt a BME Nukleáris Technika Tanszéken található NTIuCT cone-beam CT által biztosított adatok.

A feladat megoldását egy C nyelven írt fan-beam szinogram készítő szoftver létrehozásával kezdtem, amellyel mérési adatok szimulálhatók a kétdimenziós rekonstrukció számára. Ezt a „Fan-beam szinogramkészítés” című fejezetben ismeretem. Készítettem CUDA nyelven egy GPU-n futó fan-beam rekonstrukciós szoftvert, melyet a „Fan-beam rekonstrukció” fejezetben mutatok be. Ezt a szoftvert tesztelhetjük egyrészt a szimuláció segítségével, matematikai fantomokon („Tesztelés szimulált adatokon” alfejezet), másrészt az NTIuCT-vel mért adatokon is („Mérési adatok rekonstruálása” alfejezet).

A fan-beam rekonstrukciós kódot fejlesztettem tovább háromdimenziós esetre, melynek részleteit a „Cone-beam rekonstrukció” fejezet tartalmazza. A kúpsugaras szűrt visszavetítésen alapuló tomográfiás rekonstrukció gigabájt nagyságrendű adatmennyiségből képes néhány perc alatt 3D tomográfiás képet alkotni a videokártyás párhuzamosításnak köszönhetően. Az implementációt a „Megvalósítás GPU-n” alfejezetben mutatom be. A „Mérési adatok rekonstruálása” alfejezetben ismertetem a valós mérési adatokkal történő tesztelés eredményeit.

RENDELKEZÉSRE ÁLLÓ ESZKÖZÖK

Az orvosi diagnosztika egyik leggyakoribb képalkotási eljárása a tomográfias felvételek készítése. Bár az eljárás matematikai hátterét már 1917-ben kidolgozta Johann Radon, a gyakorlatban való elterjedéshez a számítástechnika néhány évtizedes fejlődésére volt szükség. Ma pedig már a videokártyák általános célú programozhatóságának köszönhetően gyakorlatilag egy hétköznapi számítógép segítségével is akár néhány perc alatt végezhetünk 3D tomográfias rekonstrukciót.

Az általunk megvalósított szoftver a számításigényes műveleteket párhuzamosítva, a videokártyán végzi el. Ennek tesztelésére egy Ubuntu 12.04 operációs rendszert futtató számítógép állt rendelkezésünkre, amelyben GeForce GTX 690 típusú videokártya található. A kártya 2 db duplamagos GPU-ból (graphics processing unit) áll, így összesen 4 GPU-mag érhető el, bár ebből a szoftver csak egyet használ. Az egyes GPU-k 2 GB memóriával rendelkeznek, de később látni fogjuk, hogy a tomográfias képalkotás során akár ennél is nagyobb adatmennyiséget kell feldolgoznunk. A szoftvert az Nvidia által létrehozott CUDA nyelven írtuk, amely a párhuzamosítást ún. szálakkal valósítja meg.^[3] Ezek a szálak nem túl gyorsan hajtják végre a feladatokat, viszont egy időben nagyon sok futhat egymástól függetlenül. A szálakat 3 dimenziós blokkokba (angolul block) strukturáljuk, a blokkokat pedig egy 3 dimenziós rácsba (angolul grid, amely régebbi kártyák esetén csak 2 dimenziós lehetett).^[3] A block és grid maximális mérete kártyánként változik, az általunk használt GeForce GTX 690 GPU-n az első két block-dimenzió mérete maximum 1024 lehet, a harmadik dimenzióé pedig maximum 64, de egy blockban összesen legfeljebb 1024 szál lehet. A grid-re nincsenek ilyen szigorú korlátozások, a mérete az egyes dimenziók mentén $2^{32}-1$ lehet. A forráskódban kernelnek hívjuk azokat a függvényeket, amelyek a GPU memóriájában lévő adatokkal végeznek műveleteket, és ezt több szál egyidejű indításával párhuzamosan teszik.

Lehetőségünk volt a rekonstrukciós szoftvert valós mérési adatokon is tesztelni, a Nukleáris Technika Intézetben épült és jelenleg is ott működő NTIuCT kúpsugaras (cone-beam) CT készülékkel. Ennek legfontosabb elemei a Dexela 1207 CMOS sík detektorpanel, Source-ray Inc. SB-80-1K típusú mikrofókuszos röntgencső, egy léptetőmotor a vizsgálandó tárgy mozgatására, illetve az ezeket vezérlő szoftver.

Cone-beam CT esetén a pontforrásból előre kilépő röntgensugarak gyengülését egy sík, egyenközűen osztott detektorral mérjük, így az információ egy 3 dimenziós kúp (kisebb, téglalap alakú detektor esetén gúla) alakú térfogathoz származik projekciónként. Ha csak a középső detektorsor által mért jelből végezzük a képalkotást, akkor legyező-leképezésről, fan-beam CT-ről beszélünk, ugyanis ekkor az információ egy kétdimenziós legyező (sík detektor esetén háromszög) alakú területből származik projekciónként. Parallel-beam vagy párhuzamos nyalábú CT esetén kiterjedt forrásból lépnek ki a röntgensugarak, melyek kollimálva érkeznek a detektorra.

A cone-beam CT jellemző geometriai paraméterei közül a dolgozatban az alábbiakat fogjuk legtöbbször használni:

- SAD: a röntgenső által meghatározott pontszerűnek tekinthető röntgenforrás és a forgástengely távolsága (source-axis-distance)
- ADD: a forgástengely és a detektor távolsága (axis-detector-distance)
- SDD: a pontszerűnek tekinthető röntgenforrás és a detektor távolsága (source-detector-distance)

A CT-k működése a Lambert-Beer törvényen alapszik, melynek átrendezett alakja heterogén gyengítési együtthatójú tárgy esetén a következő:

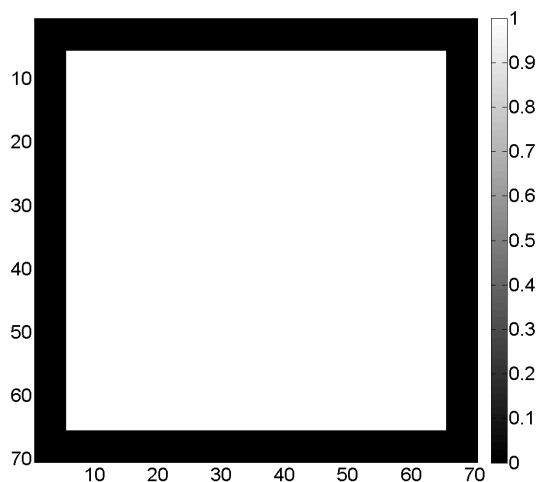
$$-\ln \frac{I}{I_0} = \int_{LOR} \mu(r) dr \quad (1)$$

Ahol I a detektált intenzitás, I_0 a kilépő intenzitás a röntgen-forrásnál, μ a tárgy gyengítési együtthatója, az integrálást pedig a pontforrás és a pontszerű detektor által meghatározott egyenesen, a válaszegyenesen (line of response, LOR) végezzük. Ezt az integrált minden egyenesre elvégezve megkapjuk a tárgy Radon-transzformáltját.

Ha a Radon-transzformáltat a válaszegyenes szögének és eltolásának függvényében ábrázoljuk, akkor megkapjuk a tárgy szinogramját.

FAN-BEAM SZINOGRAMKÉSZÍTÉS

A fan-beam CT sajátosságainak megértéséhez, illetve a rekonstrukció teszteléséhez szükségünk volt egy fan-beam szinogram készítő programra, vagyis digitális Radon-transzformációt kellett implementálnunk fan-beam geometria esetén. A fan-beam szinogram jellemzőinek bemutatásához egy négyzet fantomot (1. ábra) fogunk használni.



1. ábra

A program bemenetei a szimulálandó fantom-mátrix, a detektor adatai (pixelszám, pixelméret) illetve a geometriai paraméterek (forrás-tengely-távolság, tengely-detektor-távolság). Kimenete pedig az elkészült szinogram, esetünkben egy 360x1536 méretű mátrix, amely fokenként tartalmazza az egyes projekciókat, vagyis az 1536 db detektorpixel által "mért" jelet. A tanszéken lévő CT detektora 864 sorból, soronként pedig 1536 pixelből áll, így a valódi mérésből származó szinogram is 1536 adatot fog tartalmazni projekciónként, hiszen az a detektor megfelelő sorának jeleiből készül majd.

A szinogram egy pontjának értéke a Φ szögű projekcióhoz és Y indexű detektorpixelhez tartozó Radon-transzformáltat jelenti. Ez a két paraméter egyértelműen meghatározza a válaszegyeneset, amely összeköti a pontszerűnek feltételezett röntgen-forrást az adott detektorpixellel (2. ábra). A Radon-transzformáltat ezen egyenesre vett vonalintegrál adja. Tehát a szimuláció során összegeznünk kell a válaszegyenes mentén a fantom-mátrix

értékeit, amelyek a fantom helyfüggő gyengítési együtthatói. A vonalintegrált az OF tankönyvben található képlettel közelítjük ^[4]:

$$Rg(t_k, \vartheta_l) \approx \sum_{i=1}^N g \left(x_0 + i\Delta x, t_k \sin \vartheta_l + \frac{t_k \cos \vartheta_l - (x_0 + i\Delta x)}{\sin \vartheta_l} \cos \vartheta_l \right) \left| \frac{1}{\sin(\vartheta_l)} \right| \Delta x \quad (2)$$

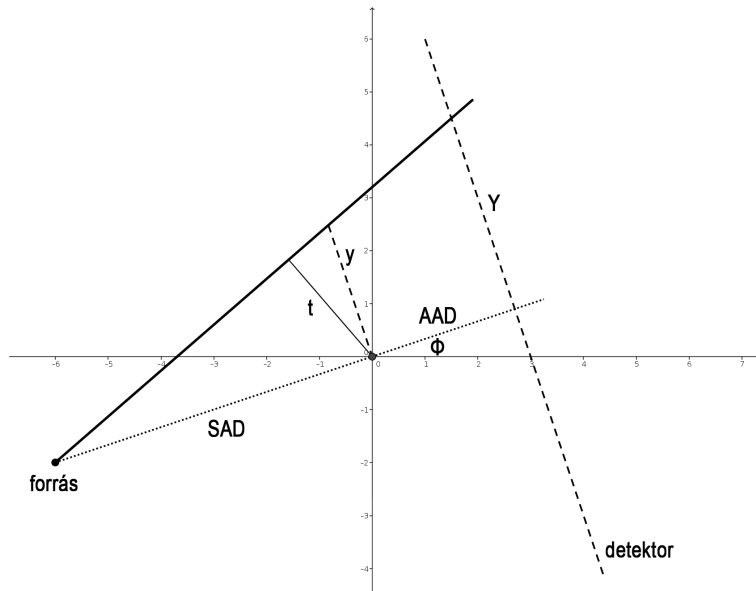
Ahol t_k és ϑ_l a válaszegyenes paraméterei, g a fantom-mátrix. (Ha $\sin \vartheta_l \leq \frac{1}{2}$, akkor az y koordináták szerint haladunk egyenletes lépésközzel.)

A művelet közvetlenül elvégezhető, hiszen Φ és ϑ_l valamint Y és t_k közt az alábbi összefüggések állnak fenn:

$$t_k = \frac{y \cdot SAD}{\sqrt{y^2 + SAD^2}}, \text{ ahol } y = \frac{SAD}{SAD + ADD} Y \quad (3)$$

Illetve

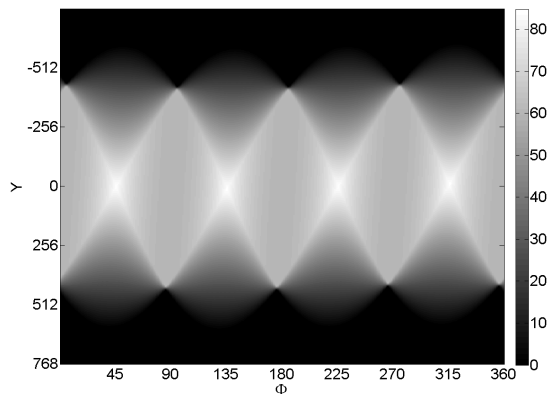
$$\vartheta_l = \Phi + \frac{\pi}{2} + \arctan \left(\frac{Y}{SAD + AAD} \right) \quad (4)$$



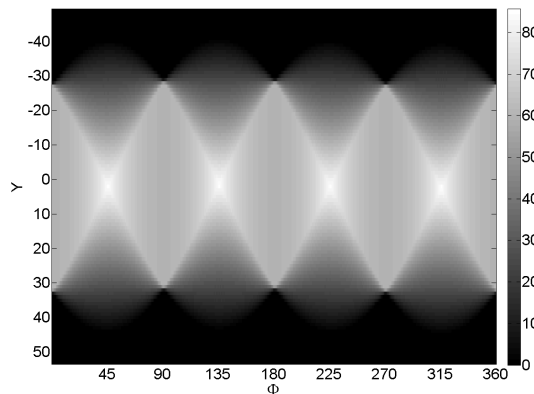
2. ÁBRA

A (3) és (4) egyenletek alapján azt a következtetést vonhatjuk le, hogy ugyanazon objektumról készített szinogramok fan-beam (3. ábra) és parallel-beam (4. ábra) esetben is

ugyanazokat az értékeket fogják tartalmazni (a Radon-transzformált definíciója miatt), de az értékek más helyeken szerepelnek majd. Míg az affin paraméter szerint csak skálázásról van szó, addig a szögváltozóban olyan Y -függő eltolást látunk, amelynek irányát Y előjele határozza meg, mértékét pedig Y abszolút értéke.



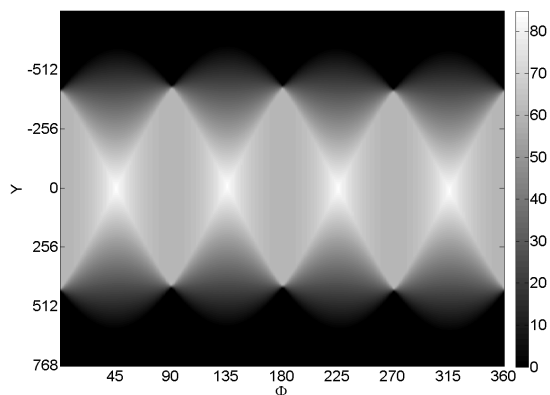
3. ábra



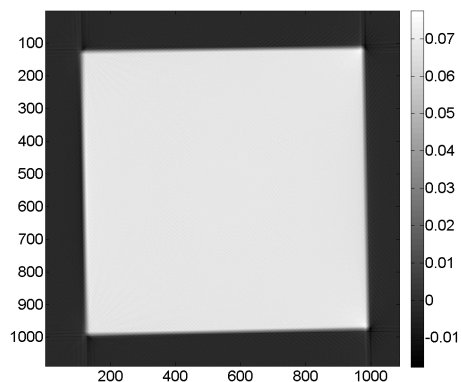
4. ábra

FAN-BEAM REKONSTRUKCIÓ

A szinogram fenti tulajdonsága lehetőséget ad arra, hogy egy fan-beam CT által mért szinogramot átmintavételezés után párhuzamos nyalábúként kezeljünk. Ezt a feltevést a MATLAB beépített függvényeivel sikerült is igazolni. A korábban is használt négyzet szimulált szinogramját (3. ábra) a MATLAB *fan2para* függvényével átmintavételeztük (5. ábra), majd az *iradon* függvényével rekonstruáltuk (6. ábra).



5. ábra



6. ábra

Lehetséges a fan beam adatok direkt rekonstrukciója is, melyhez a hagyományos szűrt visszaveítési eljárást ^[4] némileg módosítani kell. Ezt a direkt, azaz átmintavételezés nélküli eljárást valósítottuk meg GPU-környezetben. Az átmintavételezési eljárással mindeközben egy olyan tesztkörnyezet született, mellyel a GPU kódot matematikai fantomokon ellenőrizhettük.

MATEMATIKAI HÁTTÉR

A szimulációhoz hasonlóan a mérés során is a tárgy Radon-transzformáltját kapjuk meg. Ezekből az adatokból kell visszaállítanunk a vizsgált tárgy gyengítési együtthatójának helyfüggését a tárgy egyik sík metszetére. Ehhez a Radon-féle inverziós formulából indulunk ki, amelyet polárkoordinátákban felírva az (5) egyenletben láthatunk.

$$f(r, \varphi) = \frac{1}{4\pi^2} \oint_{-\infty}^{+\infty} \frac{1}{r \cos(\vartheta - \varphi) - t} \frac{\partial}{\partial t} p(t, \vartheta) dt d\vartheta \quad (5)$$

Ahol f a rekonstruált kép, r és φ a kép polárkoordinátái, t és ϑ a szinogram változói (válaszegyenesek paraméterei), p a kép Radon-transzformáltja, a mért adat.

A (5) egyenletből levezethető ^[1] a szűrt visszaveítés képlete fan-beam geometriára, melynek a visszaveítést leíró tagja:

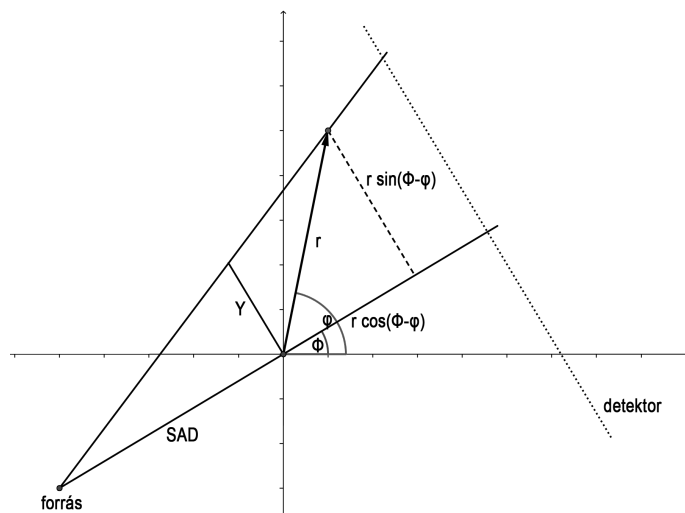
$$f(r, \varphi) = \frac{1}{4\pi^2} \oint \frac{SAD^2}{[SAD + r \cos(\varphi - \Phi)]^2} \tilde{P}_\Phi [Y(r, \varphi)] d\Phi \quad (6)$$

Ahol Φ a projekció szögét jelöli, \tilde{P}_Φ az ún. szűrt szinogram (a szűrést a 9. egyenletben ismertetjük) Φ projekcióhoz tartozó értékeit, Y pedig azt a görbét adja meg, amelyre az integrálást el kell végeznünk. Vagyis a rekonstruált kép egy (r, φ) pontját a szűrt szinogram megfelelő $Y(r, \varphi)$ görbéjén vett vonalintegrál adja (eltekintve a geometriai faktortól). Ez a görbe általában nem rácspontokon fog átmenni, emiatt a visszaveítés során interpolációra van szükség.

Az $Y(r, \varphi)$ görbe megadja, hogy egy adott Φ projekció és (r, φ) rekonstruálandó pont esetén melyik detektorpixel által mért adatot kell az integrálás során figyelembe vennünk (7. ábra). Minden projekcióban adott (r, φ) ponton pontosan egy válaszegyenes megy át, és a nagyítás figyelembevételével meghatározható a szükséges detektorpixel (amely természetesen szintén rajta van a válaszegyenesen).

$$Y(r, \varphi) = \frac{SAD \cdot r \cdot \sin(\varphi - \Phi)}{SAD + r \cdot \cos(\varphi - \Phi)} \quad (7)$$

A továbbiakban Y -ra legtöbbször úgy fogunk hivatkozni, mint egy detektorpixel-indexre, de igazából ez egy távolságot jelent a detektorral párhuzamos, forgástengelyen átmenő egyenesen. Ugyanakkor a nagyítás és pixelméret ismeretében egyértelműen azonosítható Y a detektor egy adott pixelével.

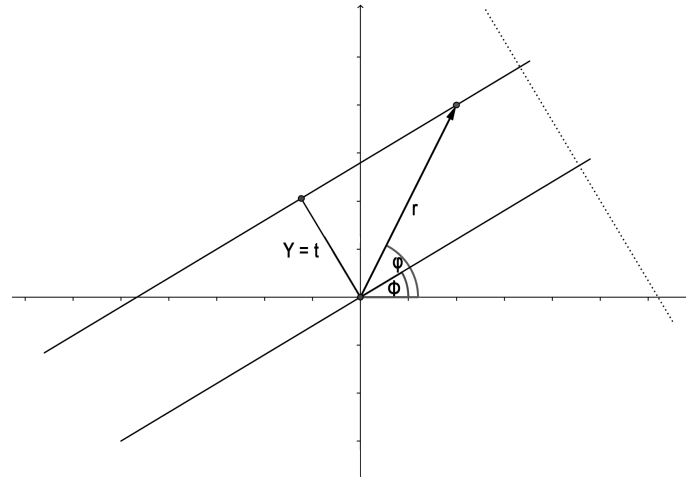


7. ÁBRA

Szemléletes megvizsgálni az $SAD \rightarrow \infty$ határesetet (8. ábra), ugyanis ekkor megkapjuk a párhuzamos nyalábú visszavetítés képletét:

$$Y(r, \varphi) = r \cdot \sin(\varphi - \Phi) = t \quad (8)$$

Ez azt jelenti, hogy megfelelő szűrőfüggvény választással a rekonstrukciós algoritmus használható parallel-beam CT berendezés esetén is, amennyiben SAD értékét nagyon nagyra állítjuk.



8. ábra

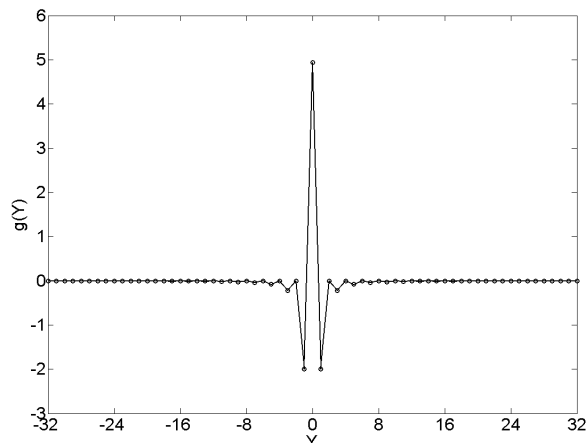
A szűrés gyakran Fourier-térben történő szorzással valósul meg^[4], de mi a (8) képlet szerinti valós térben történő konvolúciót választottuk, amely nem jelent elvi különbséget, ha a megfelelő szűrőfüggvényt használjuk.

$$\tilde{P}_\phi(Y) = \int_{-\infty}^{+\infty} \frac{SAD}{\sqrt{SAD^2 + Y'^2}} P_\phi(Y') g(Y - Y') dY' \quad (9)$$

Az [1] cikkben javasolt szűrőt implementáltuk, amelyet a (10) egyenlet ír le. A 9. ábrán a szűrő középső 65 egység szélességű szakaszát ábrázoltuk.

$$g(Y) = \text{Re} \int_0^{\omega_{y0}} \exp(i\omega Y) \omega d\omega = \frac{\omega_{y0} Y \sin(\omega_{y0} Y) + \cos(\omega_{y0} Y) - 1}{Y^2} \quad (10)$$

Ahol a célszerű választás $\omega_{y0} = \pi$ a cikk javaslata alapján.



9. ábra

MEGVALÓSÍTÁS GPU-N

Az általunk GPU-n megvalósított fan-beam rekonstrukció a következő főbb lépésekből áll, melyek közül a kerneleket később részletesen is bemutatjuk:

1. szinogram beolvasása bináris fájlból, szűrőt tartalmazó tömb inicializálása
2. szinogram és szűrő átmásolása a GPU globális memóriájába
3. szinogram szűrése (1. kernel)
4. szűrt szinogram visszavetítése (2. kernel)
5. a polárkoordinátákban mintavételezett rekonstruált kép Descartes-féle koordinátarendszerbe való transzformációja (3. kernel)
6. rekonstruált kép átmásolása a host memóriába, majd bináris fájlba írás

A szűrést végrehajtó kernelben minden szál a szűrt szinogram egy pontjának értékét számolja ki. A pont egyik koordinátája a projekció szöge, a másik pedig a konvolúcióban szereplő eltolás mértéke. Az egyes szálak a szűrőfüggvény és egy eltolt projekció skaláris szorzatát számolja ki (11. egyenlet).

$$\tilde{P}_\Phi(Y) = \sum_{Y'=0}^{PixelNum-1} P_\Phi(Y')g(Y-Y') \quad (11)$$

Itt Y jelöli az eltolás mértékét, Y' az egyes detektorpixeleket járja be, $PixelNum$ pedig a pixelek számát (vagyis a detektor méretét, amely esetünkben 1536) jelenti.

A (9) egyenletben szereplő $\frac{SAD}{\sqrt{SAD^2 + Y'^2}}$ faktort elhanyagoltuk, hiszen a forrás-tengely távolság a mi esetünkben sokkal nagyobb a detektor méreténél, ezért a tört értéke jó közelítéssel 1.

A szűrt szinogram indexeit az adott szál indexei adják, ezért a grid és block méreteket ennek megfelelően kell definiálnunk.

Visszavetítés során minden szál a rekonstruált kép egy pontját számolja ki. A kernelben lévő ciklus a (7) egyenlet alapján minden egyes projekcióban meghatározza azt az Y detektorpixel-indexet, melynek értékét vissza kell vetítenünk. Ez általában nem rácspontra esik (a kiválasztott index nem egész), ezért a két szomszédos pixel között lineáris interpolációval számítjuk ki a visszavetítendő értéket.

$$f(r, \varphi) = \frac{1}{4\pi^2} \sum_{\Phi=0}^{360} \tilde{P}_\Phi[Y] \quad (12)$$

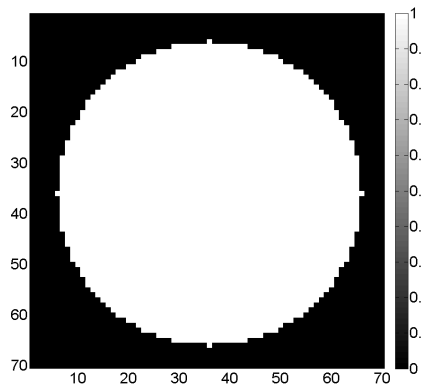
A lineáris interpolációt a rácspontok egységnyi távolsága miatt az alábbi módon végezhetjük:

$$\tilde{P}_\Phi[Y] = \tilde{P}_\Phi[Y_1] + (Y - Y_1) \cdot (\tilde{P}_\Phi[Y_2] - \tilde{P}_\Phi[Y_1]) \quad (13)$$

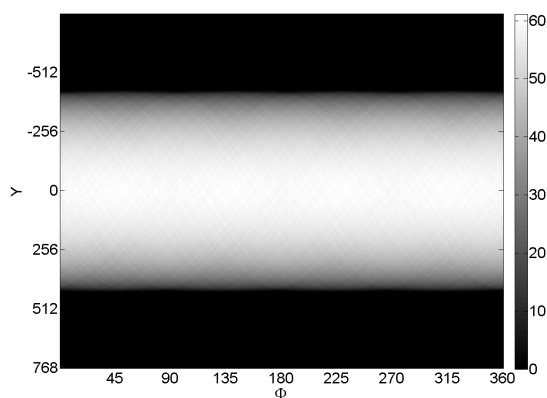
Ahol Y_1 és Y_2 a kiválasztott Y alsó illetve felső egészrésze.

Az (6) egyenletben szereplő $\frac{SAD^2}{[SAD + r \cos(\varphi - \Phi)]^2}$ faktort elhanyagolhatjuk, mivel esetünkben a forrás-tengely távolság sokkal nagyobb a rekonstruálandó terület nagyságánál, ezért a tört értéke jó közelítéssel 1.

A következőkben egy centrált homogén körlappal (10. ábra) szeretnénk illusztrálni a harmadik kernel szükségességét. A kör szinogramján (11. ábra) megfigyelhető, hogy a forgásszimmetria miatt minden projekciója megegyezik.

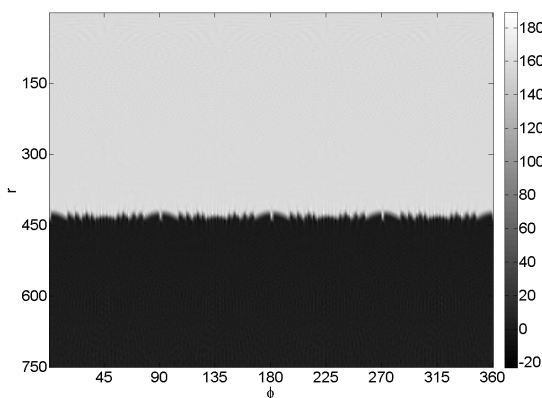


10. ábra

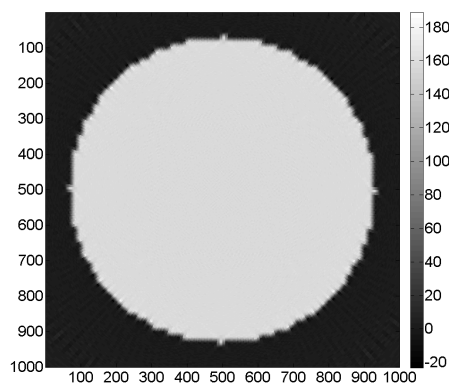


11. ábra

A visszavetítést végrehajtó kernel kimenete az (r, φ) térben regulárisan mintavételezett pontok halmazát tartalmazó mátrix (12. ábra). Ezen pontokat célszerű átmintavételezni Descartes-koordináták szerint, hogy összetettebb esetekben is könnyen értelmezhető képet kapjunk (13. ábra). A kernelben minden szál egy-egy pontnak feleltethető meg a Descartes-féle koordináta-rendszerben, melyekhez egyértelműen hozzárendelhető a megfelelő polárkoordináta. Az r koordinátát Pitagorasz-tétellel, a φ koordinátát pedig a GPU-n is használható atan2 függvénnyel^[3] határozzuk meg.



12. ábra

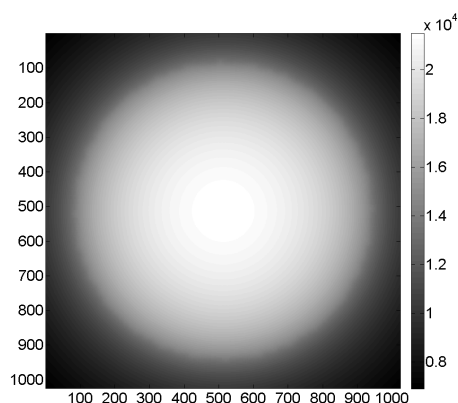


13. ábra

Általában r és φ sem esik rácspontra, de interpoláció helyett kerekítést alkalmaztunk. Ennek később nem lesz jelentősége, ugyanis egy másik eljárást fogunk használni, amelyben nem szerepel az iménti transzformáció.

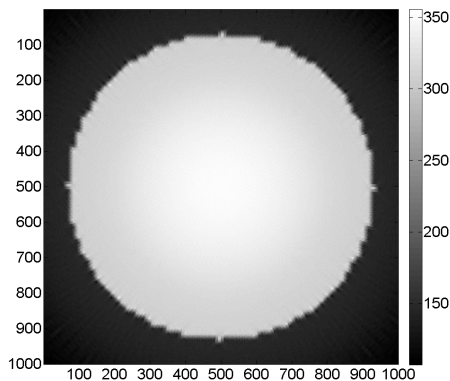
TESZTELÉS SZIMULÁLT ADATOKON

Ha az előzőekben is használt körlap szinogramját szűrés nélkül vetítjük vissza, akkor várakozásainknak megfelelően homályos képet kapunk (14. ábra). A képminőség jelentős javulása érdekében tehát szükség van a szinogram szűrésére a visszavetítés művelete előtt.

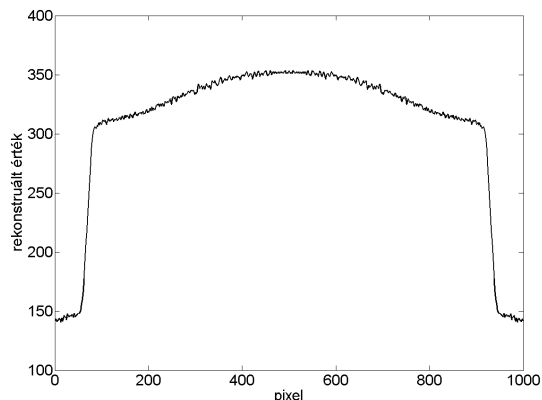


14. ábra

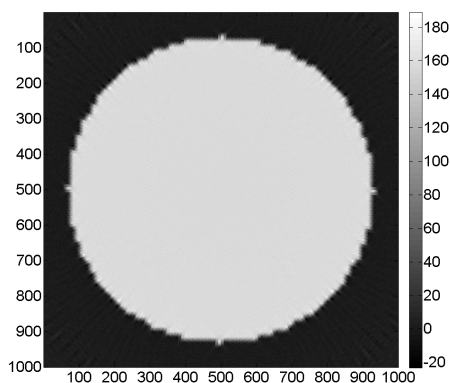
Az [1] cikk nem ad javaslatot a szűrőfüggvény tartójának véges méretére. Ezért a korábban is ábrázolt (9. ábra) élesen levágó sinus cardinalis jellegű függvényt kezdetben 257 helyen mintavételeztük, a $(-128, +128)$ tartományon, egyenletes lépésközzel, hiszen azt vártuk, hogy a többi tag elhanyagolható járulékot ad majd a konvolúciónál. Ilyen szélességű szűrő esetén viszont a rekonstruált képen a homogén területek nem lettek homogének (15. ábra). Különösen jól megfigyelhető az effektus, ha a kép középső sorát külön ábrázoljuk (16. ábra). A függvény tartóját és a mintavételek számát növelve viszont csökkent az artefaktum hatása (17. és 18. ábra). Végül a szűrőfüggvényt a $(-768, +768)$ tartományon definiáltuk, 1537 helyen mintavételezve, és a későbbiek folyamán így használtuk.



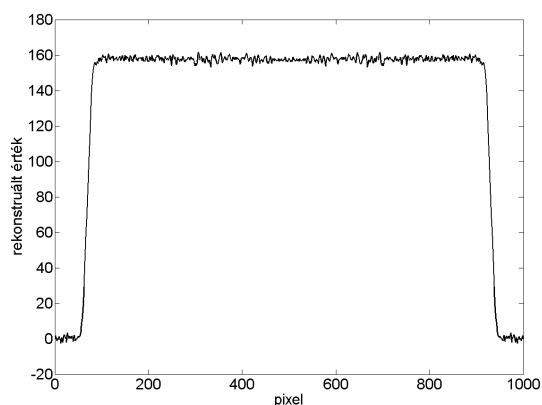
15. ábra



16. ábra



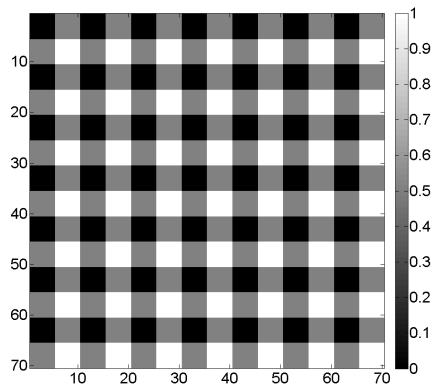
17. ábra



18. ábra

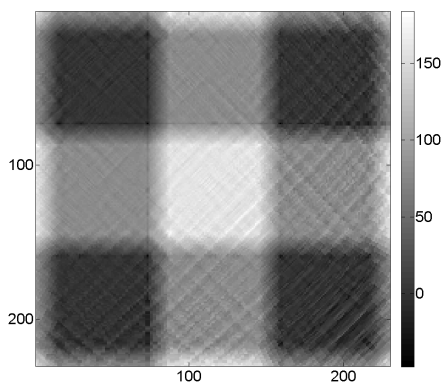
A szimuláció során állandó 1536x360 méretű szinogramokat készítettünk. A szűrés nem változtatja meg a projekciók számát, de az eltolások száma több mint ahány detektorpixelünk volt, ezért a szűrt szinogram egy nagyobb méretű mátrixot fog alkotni, mint az eredeti. Mivel összesen $1536 + 1537 - 1 = 3072$ olyan eltolás lehetséges, amikor a konvolúció eredménye különbözhet zérustól, ezért a szűrt szinogram mérete 3072x360 lett. Tehát a szűrést végző kernelünkkel összesen 3072x360 db szálat kellett indítanunk, hiszen minden szál a szűrt szinogram egy pontját számolta ki. Ezért a tesztelés során 48x45 méretű gridet választottunk, amelyben minden block 64x4 db szálat tartalmazott.

A visszavetítéssel kapcsolatos észrevételeinket az alábbi négyzetes fantommal (19. ábra) mutatjuk be.

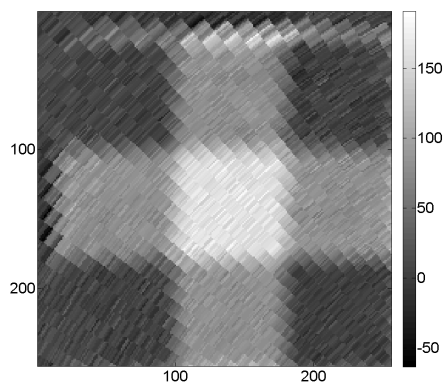


19. ábra

A visszavetített, majd Descartes-féle koordináta-rendszerbe transzformált kép közepét (20. ábra) és bal felső sarkát ábrázoltuk nagyítva (21. ábra). A kép közepétől távolodva egyre erősebb sugárirányú artefaktum jelenik meg, a kép széle felé haladva a kis négyzetek határai egyre „recésebbek”. A műterméket az okozza, hogy a visszavetítést követően minden r -hez 360 db φ koordináta tartozik, vagyis az (r, φ) rendszerben regulárisan mintavételezett pontok a Descartes-rendszerben r növelésével egyre távolabb esnek egymástól.

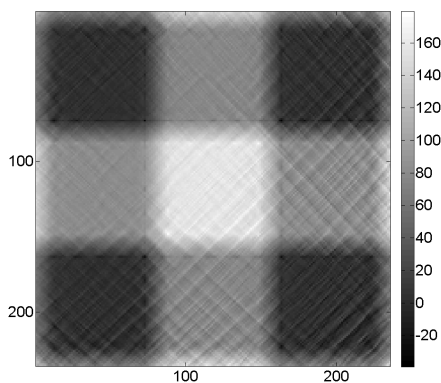


20. ábra

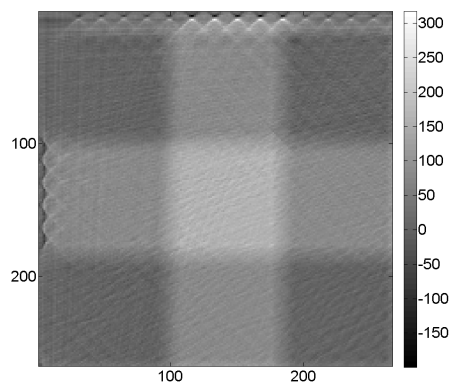


21. ábra

Ennek megoldása érdekében a következőképp módosítottuk a visszavetítés algoritmusát. A rekonstruált kép pontjait Descartes-koordináták szerint mintavételezzük regulárisan, meghatározzuk ezekhez a pontokhoz tartozó polárkoordinátákat, és ezeket fogjuk behelyettesíteni a visszavetítéshez szükséges (7) egyenletbe. Az ily módon végrehajtott visszavetítés után a kép közepén (22. ábra) nem javult a felbontás, hiszen azon a területen korábban is sok ismert pontunk volt, viszont a bal felső sarokban teljesen eltűnt a sugárirányú artefaktum (23. ábra).



22. ábra



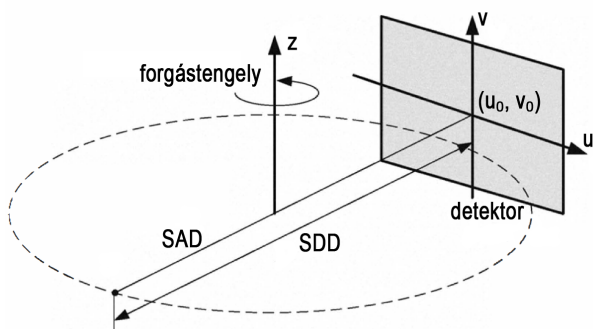
23. ábra

MÉRÉSI ADATOK REKONSTRUÁLÁSA

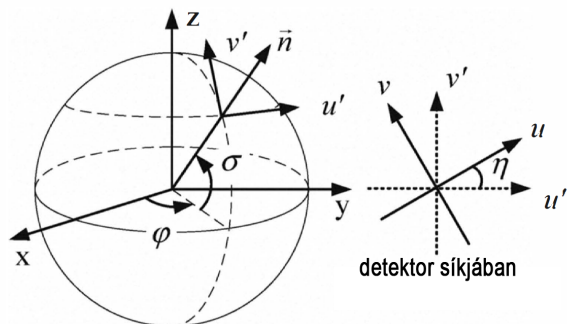
A mérési adatok rekonstrukciójához elengedhetetlen a geometria pontos ismerete. Tölgyesi Botond diplomamunkájának egyik főbb feladata a geometriai kalibrációhoz szükséges algoritmus fejlesztése volt. A [2] forrásban ismertetett módon kerülnek meghatározásra az alábbi paraméterek (24. és 25. ábra a [2] cikkből):

- detektor dőlésszöge (η)
- főnyaláb és a detektor metszéspontja (u_0, v_0)
- forrás-tengely-távolság (SAD)
- forrás-detektor-távolság (SDD)

Ezeket a paramétereket a rekonstrukció során nem tudjuk felülbírálni, csak az elkészült kép alapján lehet arra következtetni, hogy valamelyik paraméter esetleg pontatlan volt.

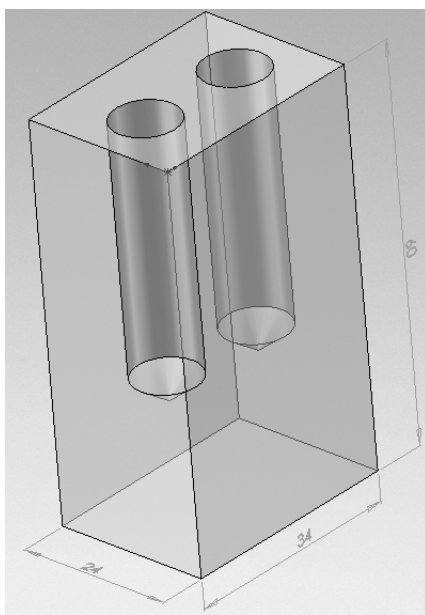


24. ábra



25. ábra

A teszteléshez egy plexi fantomot használtunk, melynek geometriája teljesen ismert. A fantom téglatest alakú, tartalmaz két hengeres furatot, melyek szimmetriatengelye párhuzamos a hasáb leghosszabb oldalával (26. ábra).



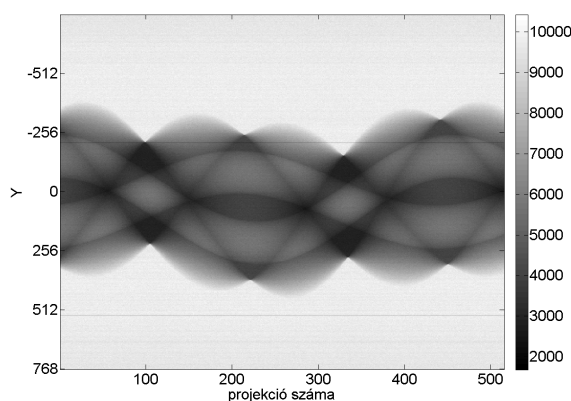
26. ábra

A fan-beam rekonstrukcióval ennek a hasábnak egy metszetét tudjuk előállítani. Ennek érdekében meg kell határoznunk a nyers mérési adatok összességének azt a részalmazát, amelyekre ténylegesen szükség van a rekonstrukcióhoz, illetve megfelelő formátumúra kell ezeket alakítani. Vagyis készítenünk kell egy fan-beam szinogramot a mérési adatokból, amely a rekonstrukciós szoftver bemenete lesz.

Szinogramkészítés mérési adatokból

A szinogram a tárgy egy adott szeletének Radon-transzformáltját tartalmazza. Az általunk használt mérési elrendezésben csak a főnyalábot tartalmazó síkot tudjuk rekonstruálni, hiszen csak ennek a metszetnek ismerjük elegendően sok projekcióját. A szinogramhoz tehát ki kell választanunk a detektornak azt a sorát, amely többek közt a főnyaláb gyengülését méri. Vagyis a fan-beam szinogram esetünkben a geometriai kalibráció v_0 paramétere által kijelölt detektorsorból készül. (A 2D rekonstrukcióhoz erre a paraméterre a továbbiakban nincs szükség, csak a szinogram előállításánál van szerepe.)

Az összes projekció v_0 által meghatározott sorát egy mátrixban ábrázolva szinogramhoz hasonló képet kapunk (27. ábra).

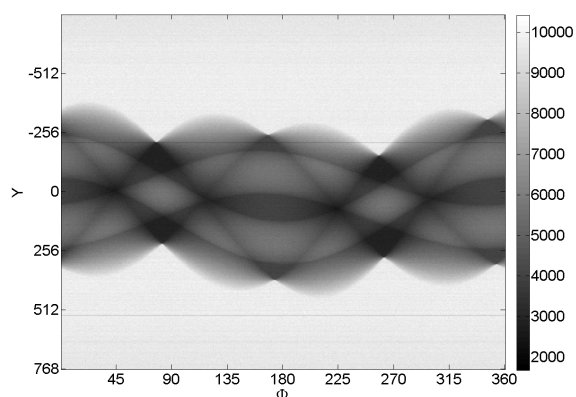


27. ábra

A fantomot úgy pozícionáltuk, hogy a főnyalábot tartalmazó szeleten látható legyen a két hengeres furat is. A projekciókon ez világosabb területként jelenik meg.

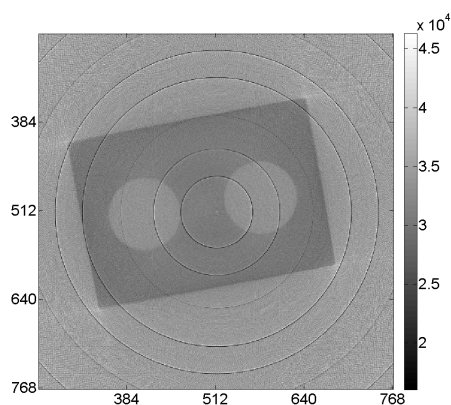
A tanszék berendezése a rövid mérési idő érdekében nem step-and-shoot üzemmódban működik, hanem állandó Röntgen-expozíció mellett adott időközönként történik a detektorból a jelek kiolvasása. Ezen kívül nincs lehetőségünk arra, hogy a mérendő tárgyat pontosan 360 fokban forgassuk körbe, így minden alkalommal egy teljes fordulatnál többről vettük fel a projekciókat, majd utólag kellett eldönteni, hogy hány projekció készült mire 360 fokot fordult a tárgy. Erre két lehetőségünk is van, használhatunk egyrészt egy erre a célra készített MATLAB szkriptet, másrészt a mozgatás paramétereinek ismeretében kiszámítható a kérdéses projekciósám. A megfelelő számú projekció v_0 -adik sorát ábrázolva és a projekciók sorszámait szögekre átszámolva megkapjuk a rekonstrukció számára bemenetként szolgáló szinogramot (28. ábra).

A detektor dőlésszögét (η) a szinogramkészítés során elhanyagoljuk, hiszen jellemzően 1 fok alatti értékről van szó, és a rekonstruált képre nincs számottevő hatással.



28. ábra

Ebből a szinogramból a pontos geometriai adatok ismeretében rekonstruálható a fantom „középső” szelete (29. ábra).



29. ábra

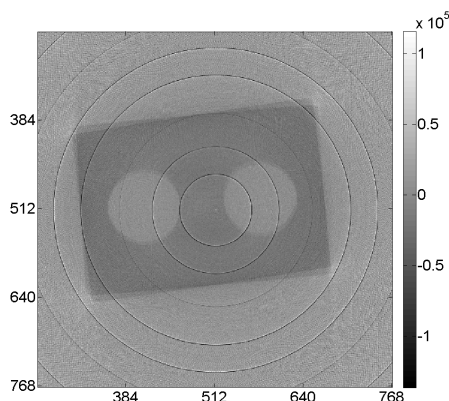
Különböző méretű szinogramok szűrése

Az elkészült szinogram mérete minden mérés esetén más volt, hiszen a projekciók száma nem volt állandó, ezért a rekonstrukció során a szűrésénél nem használhattunk statikus számú szálát. Ehelyett a következőképp jártunk el. Meghatároztuk azt a legkisebb $ProjNum_{10}$ számot, amely a projekciószámnál nagyobb és 10-zel osztható. Mivel a szűrés bemeneteként szolgáló szinogram és a szűrt szinogram „magassága” is statikus, ezért ezen dimenzió mentén

a korábbihoz hasonlóan maradhatott a grid mérete 64, a block mérete pedig 48. Ugyanakkor a másik dimenzió mentén a grid méretét $ProjNum_{10}/10$ -nek választottuk, míg a block méretét 10-nek. A grid mérete $ProjNum_{10}$ definíciója miatt biztosan egész szám lett. Ez tehát azt is jelentette, hogy a szűrt szinogram mérete a másik dimenzió mentén is megnőtt, mintha több projekciót tartalmazna, mint az eredeti, mérésből származó szinogram. Ez igazából nem jelent gondot, mert a visszavetítés során nem használtuk az ily módon megjelenő új „projekciókban” lévő értékeket, ezért nem lényeges, hogy ide milyen értékek kerülnek.

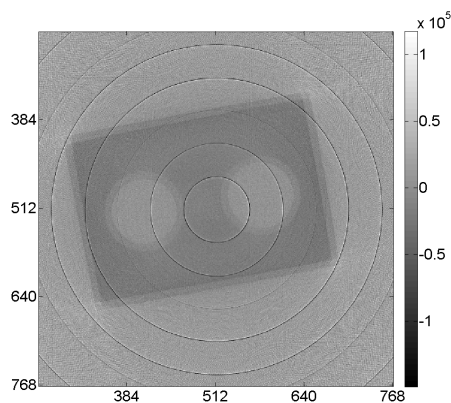
Rekonstrukciótól független képhibák

Amennyiben a 360 fokot lefedő projekciók számát nem sikerül elég pontosan meghatározni, akkor szellemképes lesz a végeredmény, amely már 2%-os eltérés (459 projekció helyett csak 449-ből készítve a szinogramot) esetén is jelentős (30. ábra).



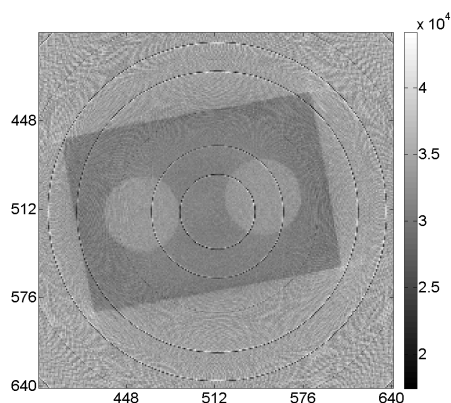
30. ábra

A szinogramon a forgástengely vetületét az u_0 koordináta jelöli ki, mivel feltételezzük, hogy a tengely a főnyaláb vonalában található. A megfelelő képminőség eléréséhez szükség van ennek a paraméternek a pontos ismeretére is. Már 1%-os tévedés (u_0 értékét a rekonstrukció során 802 helyett 810-re állítva) esetén is artefaktum jelenik meg (31. ábra).



31. ábra

Az SAD és SDD paraméterek akár 50%-os hibája is csak minimális szellemkép-műterméket eredményez, viszont a rekonstruált képen a tárgy mérete jelentősen megváltozik (32. ábra).

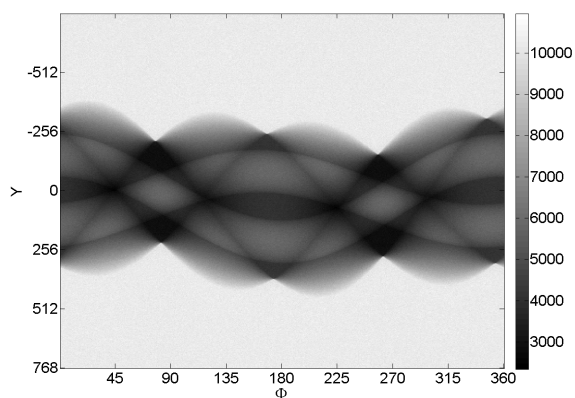


32. ábra

Természetesen csak akkor kapunk helyes méreteket, ha a megfelelő SAD = 315 mm távolságot használjuk a visszavetítés során. Ebben az esetben a plexihasáb oldalainak hossza a rekonstruált képről leolvastva a következőnek adódott: 33,82 mm illetve 23,80 mm. A „hivatalos”, tolmérővel mért értékek 34 mm és 24 mm, vagyis 1%-nál kisebb eltérést kaptunk. Ha a rekonstrukciót SAD = 165 mm értékkel végezzük, akkor a plexihasáb oldalainak hosszát tévesen tudjuk csak meghatározni (17,82 mm és 12,50 mm), viszont a képminőség nem változik számottevően.

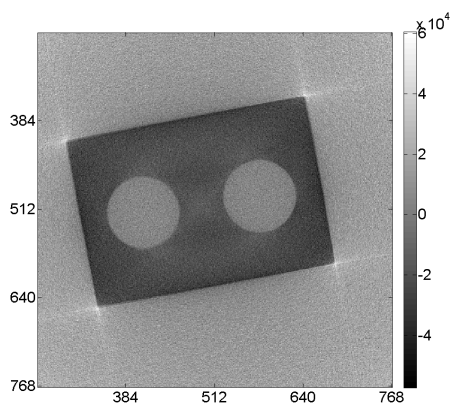
Gain korrekció hatása

A képeken látható koncentrikus köröket azok a detektorpixelek okozzák, amelyek az egyes detektormodulok határainál helyezkednek el, és nem adnak jelet. Jól megfigyelhetők a szinogramon is vízszintes vonalak (33. ábra), melyek hatását rekonstrukció során a szűrés művelete tovább növeli. Ezeket a hibákat hivatott kompenzálni a gain (erősítés) korrekció, amelynek elkészítése szintén Tölgyesi Botond feladata volt. A gain korrigált szinogramon megfigyelhető többek közt a vízszintes vonalak eltűnése (33. ábra).



33. ábra

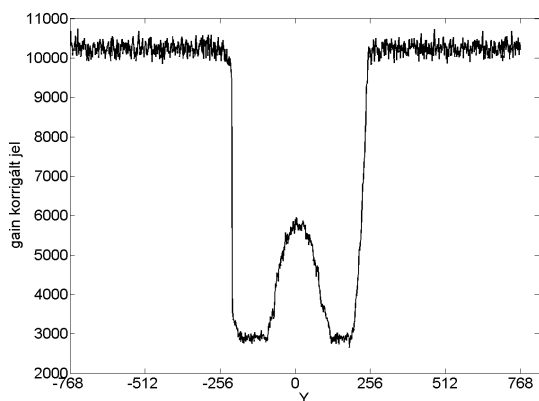
A gain korrekció a rekonstrukciós algoritmust nem befolyásolja, de a képminőségen jelentősen javít, gyakorlatilag teljesen eliminálja a köröket, illetve növeli a kép homogenitását (34. ábra).



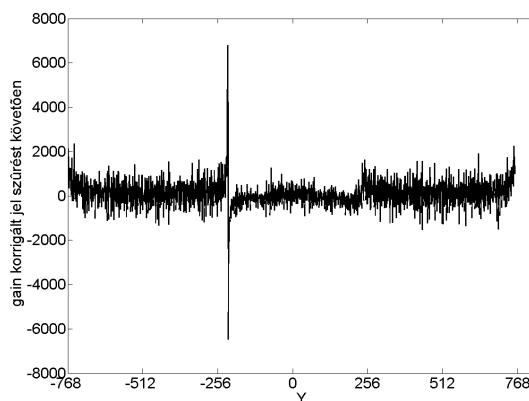
34. ábra

Rekonstrukció által okozott artefaktum

A hasáb sarkainál lévő világos foltokat a rekonstrukció szűrése eredményezi. Ugyanis 90° -onként találunk olyan projekciókat, ahol rövid tartományon belül (néhány pixel) a mért jel a többszörösére változik (a szinogramon éles fekete-fehér átmenet, 33. ábra). Ezt a különbséget az élkimelő hatású szűrő tovább növeli. A jelenség illusztrálása érdekében külön ábrázoltuk a még szűrés nélküli gain korrigált szinogram 99. projekcióját, amely a $77,65^\circ$ -hoz tartozik (35. ábra), illetve ugyanezt a szűrést követően is megtettük (36. ábra). Azok a rekonstruált pontok lesznek tehát túl „világosak”, amelyekhez tartozó szinogram-térbeli görbék áthaladnak a szűrést követően keletkező nagy intenzitású pontokon.



35. ábra



36. ábra

CONE-BEAM REKONSTRUKCIÓ

A cone-beam rekonstrukcióhoz az eddigiekben tárgyalt fan-beam rekonstrukciós eljárást terjesztjük ki három dimenzióra. Az a célunk, hogy a korábban is használt mérési adatokból a tárgynak ne csak egy kétdimenziós szeletét állítsuk elő, hanem a CT látóterébe eső teljes térfogatot.

Ez a feladat viszont nem triviális. A pontszerűnek feltételezett Röntgen-forrás és a detektor egy tetszőleges sora meghatároz egy ferde síkot. Ez a sík akkor rekonstruálható a Radon-féle inverziós formulával, ha végtelen sok (illetve elég sok) projekcióját ismerjük. Cone-beam CT esetében viszont ez csak a középső, főnyaláb által kijelölt síkra igaz (amelyre tulajdonképpen az előzőekben a fan-beam rekonstrukciót végeztük), a többi síknál csak 1-1 projekció áll a rendelkezésünkre. Azt is mondhatjuk ez alapján, hogy minden rekonstruálandó ponthoz egy egyedi sík-halmaz tartozik, amelyeknek egy közös metszéspontjuk van: a rekonstruálandó pont.

MATEMATIKAI HÁTTÉR

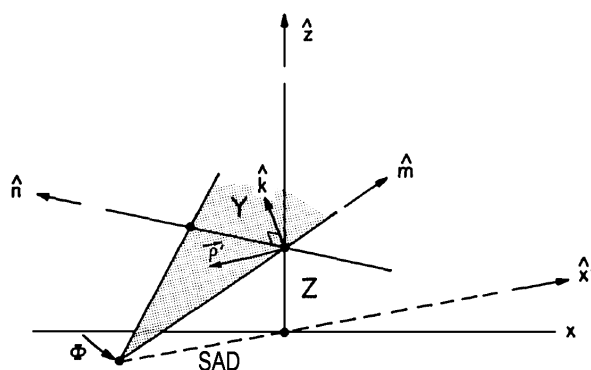
A problémára egy heurisztikus megoldást javasol az [1] cikk, melyet az alábbiakban részletesen ismertetünk. A (14) egyenlettel a kétdimenziós szűrt visszavetítés összevont képletéből indulunk ki, amelynek elemi transzformációit az (6), (7), (9) és (10) egyenleteknél már bemutatottuk.

$$f(r, \varphi) = \frac{1}{4\pi^2} \operatorname{Re} \oint d\Phi \frac{SAD^2}{[SAD + r \cos(\varphi - \Phi)]^2} \times \int_0^\infty \omega d\omega \int_{-\infty}^{+\infty} dY \frac{SAD}{\sqrt{SAD^2 + Y^2}} P_\Phi(Y) \exp \left[i\omega \left(\frac{SAD r \sin(\varphi - \Phi)}{SAD + r \cos(\varphi - \Phi)} - Y \right) \right] \quad (14)$$

Vezessünk be néhány új változót: r és φ helyett jelölje a rekonstruálandó pontot a ρ helyvektor, melynek megfelelő vetületeit \hat{m} és \hat{n} egységvektorokkal képzett skaláris szorzata adja. Az \hat{m} vektor mutasson a forrástól a detektorsor középső pixele felé, az \hat{n} pedig erre merőlegesen, a középső pixeltől a detektorsor szélén lévő pixel felé (37. ábra, de itt még $Z = 0$, az [1] cikkből átvéve).

$$\begin{aligned}
 f(\rho) &= \frac{1}{4\pi^2} \operatorname{Re} \oint d\Phi \frac{SAD^2}{[SAD + \rho \cdot \hat{m}]^2} \\
 &\times \int_0^\infty \omega d\omega \int_{-\infty}^{+\infty} dY \frac{SAD}{\sqrt{SAD^2 + Y^2}} P_\Phi(Y, Z=0) \exp \left[i\omega \left(\frac{SAD \rho \cdot \hat{n}}{SAD + \rho \cdot \hat{m}} - Y \right) \right]
 \end{aligned} \tag{15}$$

Következő lépésben már nem a 2 dimenziós eljárással is rekonstruálható síkot fogjuk vizsgálni, hanem az egyik ferde síkot (37. ábra, $Z \neq 0$). A sík normálvektora legyen \hat{k} . Ekkor a három egységvektor jobbsodrású rendszert alkot: $\hat{k} = \hat{m} \times \hat{n}$.



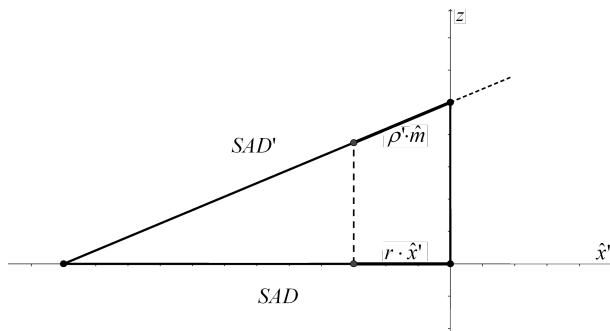
37. ábra

Ez a ferde sík akkor lenne rekonstruálható, ha a forgástengely a z tengellyel adott szöget bezárva, a \hat{k} vektor irányába mutatna, és ekörül vennénk fel elegendően sok projekciót a tárgyról. Explicit rekonstrukciós képlet helyett azt határozzuk meg, hogy ennek a ferde síknak egy pontjához mekkora járulékot ad egy elemi $\delta\Phi'$ szögű projekció ($\delta\Phi'$ a \hat{k} körüli forgatásra utal). Az adott pontba mutató helyvektor $\rho' + Z\hat{z}$ alakba írható, ahol ρ' a ferde síkra illeszkedik, és a forgástengelytől mutat a rekonstruálandó pontba, \hat{z} a z -irányba mutató egységvektor, Z pedig a megfelelő koordináta. A sík és a forgástengely metszéspontja ekkor SAD' távolságra van a pontszerű forrástól.

$$\begin{aligned}
 \delta f(\rho' + Z\hat{z}) &= \frac{1}{4\pi^2} \operatorname{Re} \delta\Phi' \frac{SAD'^2}{[SAD' + \rho' \cdot \hat{m}]^2} \\
 &\times \int_0^\infty \omega d\omega \int_{-\infty}^{+\infty} dY \frac{SAD'}{\sqrt{SAD'^2 + Y^2}} P_\Phi(Y, Z) \exp \left[i\omega \left(\frac{SAD' \rho' \cdot \hat{n}}{SAD' + \rho' \cdot \hat{m}} - Y \right) \right]
 \end{aligned} \tag{16}$$

Hajtsunk végre néhány újabb változócsere, fejezzünk ki minden lehetséges paramétert az $r = (x, y, z) = \rho' + Z\hat{z}$ helyvektorral! Belátható, hogy a \hat{k} és \hat{z} tengely körüli elemi

szögelfordulások között az alábbi kapcsolat áll fenn: $\delta\Phi' = \delta\Phi \frac{SAD}{\sqrt{SAD^2 + Z^2}}$. Az \hat{n} egységvektor pedig az elforgatott y tengely irányába mutat, tehát $\hat{n} = \hat{y}'$. Az elforgatott x tengely irányába mutató egységvektor pedig legyen \hat{x}' .



38. ábra

Háromszögek hasonlóságából (38. ábra) az alábbi összefüggés következik:

$$\rho \cdot \hat{m}' = \frac{SAD'}{SAD} r \cdot \hat{x}' \quad (17)$$

Ezek alapján a (16) egyenletet a következő alakba írhatjuk:

$$\begin{aligned} \delta f(r) &= \frac{1}{4\pi^2} \text{Re} \delta\Phi \frac{SAD^2}{[SAD + r \cdot \hat{x}']^2} \quad (18) \\ &\times \int_0^\infty \omega d\omega \int_{-\infty}^{+\infty} dY \frac{SAD}{\sqrt{SAD^2 + Y^2 + Z^2}} P_\Phi(Y, Z) \exp \left[i\omega \left(\frac{SAD r \cdot \hat{y}'}{SAD + r \cdot \hat{x}'} - Y \right) \right] \end{aligned}$$

Ezzel megkaptuk egy adott projekció járulékát az r rekonstruálandó ponthoz. A pont rekonstruált értékéhez minden lehetséges projekcióra összegeznünk kell a (18) egyenletet. A kapott összefüggést elemi transzformációkra bontva azt alábbi összefüggéseket kapjuk.

$$f(r) = \frac{1}{4\pi^2} \oint \frac{SAD^2}{[SAD + r\hat{x}']^2} \tilde{P}_\Phi[Y(r), Z(r)] d\Phi \quad (19)$$

$$\tilde{P}_\Phi(Y, Z) = \int dY' \int dZ' g_y(Y - Y') g_z(Z - Z') P_\Phi(Y', Z') \frac{SAD}{\sqrt{SAD^2 + Y'^2 + Z'^2}} \quad (20)$$

A (19) egyenlet írja le a 3D visszavetítést a (6) képlettel analóg módon, míg a (20) egyenlet a projekció szűrését a (9) képlettel analóg módon. A visszavetítéshez szükség van még $Y(r)$ és $Z(r)$ definiálásra, amely a (7) képlettel analóg:

$$Y(r) = \frac{SADr \cdot \hat{y}'}{SAD + r \cdot \hat{x}'} \quad (21)$$

$$Z(r) = \frac{SADr \cdot \hat{z}'}{SAD + r \cdot \hat{x}'} \quad (22)$$

Az Y irányú szűrő a fan-beam rekonstrukció során is használt sinc jellegű függvény, míg a Z irányú szűréshez [1] az alábbi függvényt javasolja:

$$g_z(Z) = \frac{\sin \omega_{z_0} Z}{\pi Z} \quad (23)$$

Ahol ha $\omega_{z_0} = \pi$, akkor lényegében nem történik szűrés.

A 3D szűrt visszavetítés tehát hasonló elven történik, mint 2 dimenzióban: a projekciók szűrése után a visszavetítés fogja meghatározni egy rekonstruált pont értékét a térben. A visszavetítéskor ezúttal is minden projekció esetén az adott ponton átmenő válaszegyenes jelöli ki azt az egy detektorpixel, amelynek a jelét vissza kell vetítenünk. Az eddigiekkel ellentétben a rekonstrukcióhoz használt projekciók már 2 dimenziós mátrixok, ezért mindkét dimenzióban lehetséges a szűrés. A tapasztalataink illetve az [1] cikk szerint a korábban is használt Y irányú szűrés elengedhetetlen a jó felbontású rekonstruált képhez, míg a Z irányban történő szűrésnek nincs akkora hatása. A szűrést ezúttal is valós térben végeztük, konvolúcióval.

MEGVALÓSÍTÁS GPU-N

A cone-beam rekonstrukcióhoz 2-3 nagyságrenddel nagyobb adatmennyiség feldolgozására, tárolására van szükség, mint a kétdimenziós esetben. Az 1536x864 pixel méretű projekciók a mérés közben 2,5 MB-os bináris fájlba írónak, melyekből egy mérés során több száz keletkezik. Gyakran több mint 700 projekciót készítettünk a vizsgált tárgyról, ez több mint 1750 MB memóriaterületet igényelne. A rekonstruált 3 dimenziós 512x512x512 voxelből álló kép mérete 512 MB. Ezekon kívül a szűrt projekciókat is tárolnunk kellene, viszont a rendelkezésre álló GPU memóriája csak 2 GB. Tehát nem tudjuk párhuzamosan végezni a rekonstrukciót egyszerre az összes mérési adat felhasználásával. Ehelyett az egyes projekciókat egy CPU-n futó ciklusban egymás után dolgozzuk fel, ebből hívjuk meg a szükséges kerneleket.

A program indulásakor létrehozuk az y és z irányú szűrőket, illetve lefoglaljuk a szükséges memóriaterületeket, majd a rekonstrukciót végző ciklus következik, mely az alábbi lépéseket tartalmazza:

1. adott projekció beolvasása bináris fájlból, majd átmásolása a GPU memóriájába
2. adott projekció y irányú szűrése (1. kernel)
3. adott projekció z irányú szűrése (2. kernel)
4. szűrt projekció visszavetítése (3. kernel)

Az y irányú szűrés lényegében a fan-beam esettel ekvivalens módon történik, formailag annyi különbséget találhatunk, hogy míg két dimenzióban a szinogram 1536 elemű oszlopaire hajtottuk végre a konvolúciót, addig itt a projekció 1536 elemű soraira kell ugyanezt megtennünk. A korábbi esethez hasonlóan az első, y irányú konvolúciót követően a projekció mérete 1536x864 pixelről 3072x864 pixelre növekszik, ezért ebben a kernelben összesen 3072x864 szálát kell indítanunk. Ehhez 128x32 db block-ot definiáltunk, melyek mindegyékben 24x27 db szál volt.

A z irányú szűrőfüggvény tartójára sem ad iránymutatást az [1] cikk, ezért az Y -szűrőhöz analóg módon a Z -szűrőt a $(-432, +432)$ tartományon, 865 helyen mintavételezve definiáltuk. A (23) egyenletben szereplő ω_{z_0} paramétert széles tartományon változtatva sem tapasztaltunk jelentős műterméket a rekonstruált képen. A második, z irányú konvolúciót követően a szűrt projekciót tartalmazó mátrix mérete 3072x864 pixelről 3072x1728 pixelre növekszik, ezért a block és grid méreteket ennek megfelelően kell megválasztanunk, vagyis összesen ennyi szála lesz szükségünk. Ennek érdekében a grid-et 128x192 méretűre választottuk, a block-ok mérete pedig 24x9 volt. Az y , majd z irányban is megszürt projekció szolgál a visszavetítés bemeneteként.

A rekonstrukció eredményét tartalmazó 512MB méretű tömböt a GPU globális memóriájában tároljuk, és a CPU-n futó ciklus minden egyes iterációja során, az egyes szűrt projekciók visszavetítésével ennek elemeit inkrementáljuk a megfelelő értékkel. Minden szál a voxeltömb egy elemét módosítja, ezért összesen 512x512x512 szálát indítunk, melyhez 64x64x64 méretű gridet, és 8x8x8 méretű block-okat definiáltunk. A szálak indexei fogják meghatározni az r helyvektor koordinátáit a Descartes-féle koordináta-rendszerben. Továbbá $Y(r)$ és $Z(r)$ kiszámításához szükségünk van még az \hat{x}' és \hat{y}' egységvektorokra, amelyek a Φ

szögből közvetlenül számolhatók. A kernel a projekció Φ szögét a CPU-n futó ciklustól kapja paraméterként, a ciklusban pedig az aktuális projekció sorszámából és a teljes projekciószámból határozzuk meg azt. Mivel $Y(r)$ és $Z(r)$ általában nem egész számok, ezért kerekítést alkalmazunk.

MÉRÉSI ADATOK REKONSTRUÁLÁSA

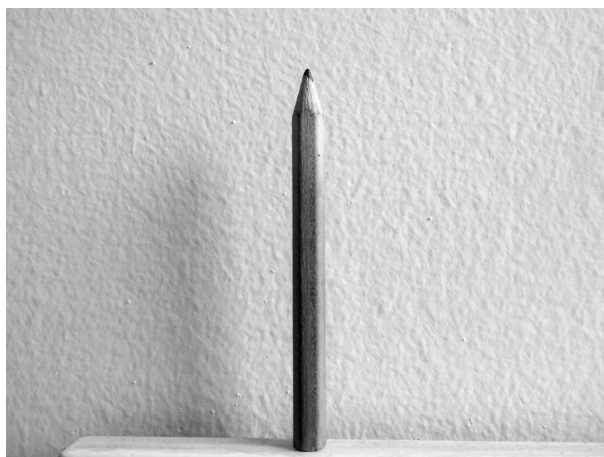
A kétdimenziós esettel ellentétben itt nincs szükség a mérési adatok előzetes feldolgozására, ugyanis nem kell szinogramot készítenünk, csak a 360° -ot lefedő projekciók számát kell tudnunk pontosan.

A mért projekciók szűrése a fent ismertetett módon történik.

A 3D rekonstrukció során a visszavetítéshez elsősorban ki kell választanunk a szűrt projekció megfelelő $\tilde{P}_\Phi[Y(r), Z(r)]$ értékét. Ezt a (21) és (22) egyenletek segítségével, valamint a geometriai kalibrációból kapott paraméterek felhasználásával tesszük meg. Először meghatározzuk az aktuális szálindexek alapján a rekonstruálandó pont valós térbeli koordinátáit, illetve a projekció szögéből az \hat{x}' és \hat{y}' egységvektorokat. Ez után a (21) és (22) egyenletek alapján kiszámítjuk, hogy melyik pixelértéket kellene visszavetítenünk, ha a főnyaláb pontosan a detektor középső pixelére mutatna ($u_0 = 768, v_0 = 432$), valamint a detektor ferdesége elhanyagolható lenne ($\eta = 0^\circ$). Végül a kapott eredményeket korrigáljuk a geometriai adatokkal, és egésze kerekítjük, így a szűrt projekciót tartalmazó tömb megfelelő (Y, Z) indexű elemét már ki tudjuk olvasni, és az adott voxel értékét meg tudjuk ezzel növelni.

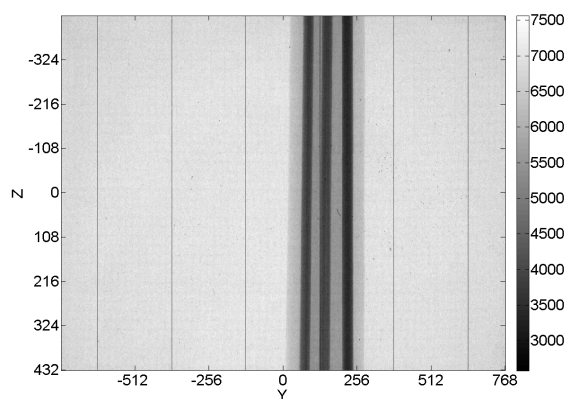
Geometriai paraméterek vizsgálata

3D rekonstrukciónál a geometriai kalibráció által biztosított adatok közül az u_0 , SAD, SSD paramétereknek, illetve azok hibájának ugyanolyan hatása van, mint a kétdimenziós esetben. Ezen kívül itt is ugyanolyan fontos a 360 fokot lefedő projekciók számának pontos meghatározása. Továbbá cone-beam rekonstrukciónál megnő a szerepe az η paraméternek, melyet 3 db hagyományos grafitceruza rekonstrukciójával mutatunk be.



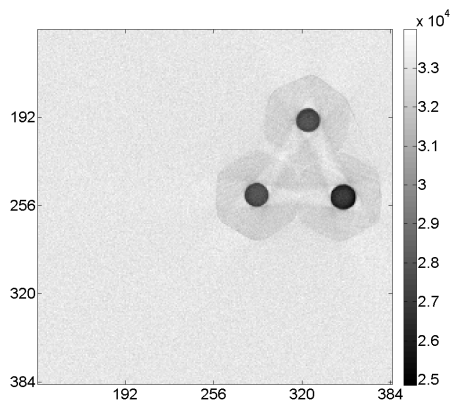
39. ábra

A vizsgált ceruza egy fából készült hatszög alapú hasáb (39. ábra), melynek közepén egy grafitenger található, a ceruzákról készült projekción megfigyelhető a nagyobb gyengítési együtthatóval rendelkező grafit, illetve az ezt körülvevő, „világosabb” fa (40. ábra).

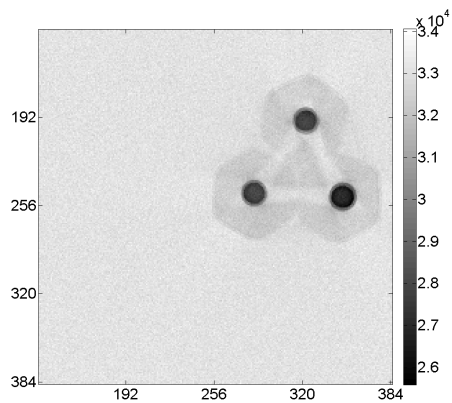


40. ábra

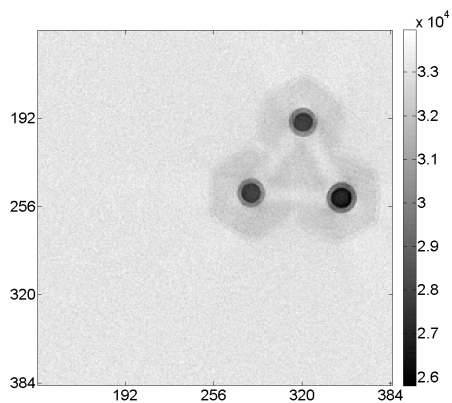
Ha nem vesszük figyelembe a detektor ferdeségét, akkor a rekonstruált 3D kép axiális szeletein z -től függő mértékben tapasztalunk szellemkép-artefaktumot (41. ábra: 1. szelet, 42. ábra: 128. szelet, 43. ábra: 256. szelet, 44. ábra: 384. szelet, 45. ábra: 512. szelet).



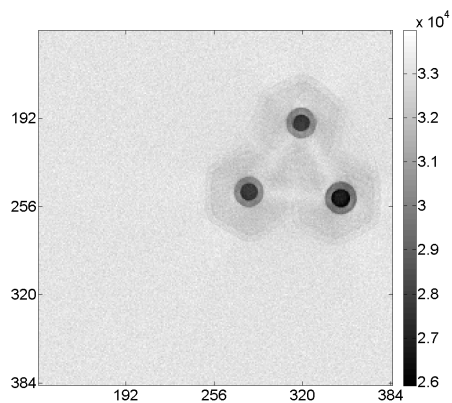
41. ábra



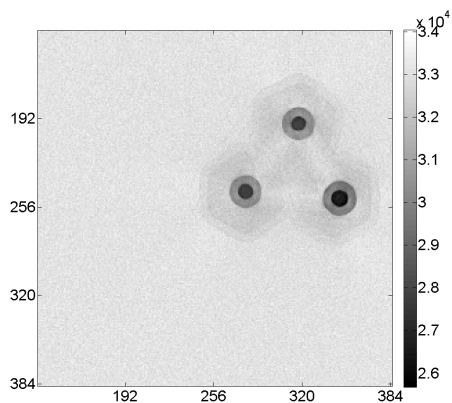
42. ábra



43. ábra

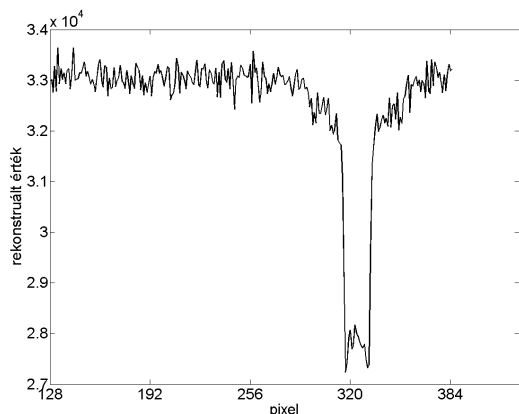


44. ábra

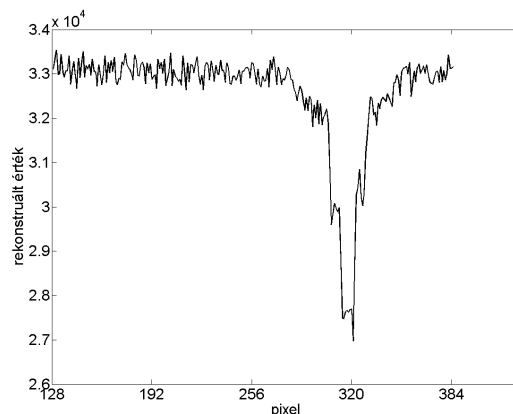


45. ábra

A $z = 1$ (46. ábra) és $z = 512$ (47. ábra) szeletek egyik grafitot is tartalmazó sorát ($Y = 192$) külön is ábráztunk, az utóbbi esetben a grafikonon is jól látható a grafitrúd szélén kialakuló szellemkép, előbbinél pedig az artefaktum hiánya.



46. ábra



47. ábra

Tapasztalataink szerint η paraméter értékében 10%-os tévedés még nem okoz jelentős műterméket, 25%-os hiba fölött viszont már a rekonstruált képen is látható mértékű a jelenség.

A fenti szeleteken a grafitokat összekötő sávok mentén világos területeket látunk. Ennek oka a spektrumfelkeményedés, ezért a két grafitrúd nagy elnyelése miatt az őket metsző válaszegyeneseken a ténylegesnél kisebb gyengítési együtthatót rekonstruálunk. Ezt a képhibát sajnos a szűrt visszavetítés nem tudja kezelni.

A z irányú szűrés vizsgálata

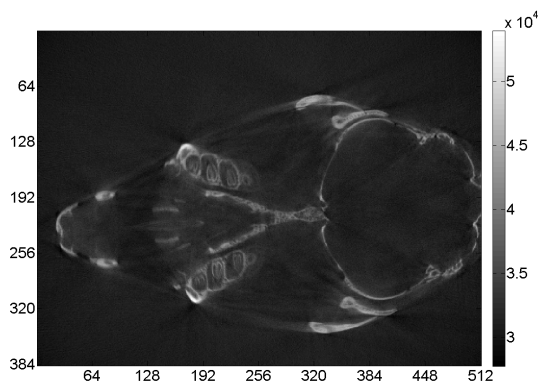
A z irányú szűrő hatását egy nyúlkoponya 3D rekonstrukcióján mutatjuk be (48. ábra).



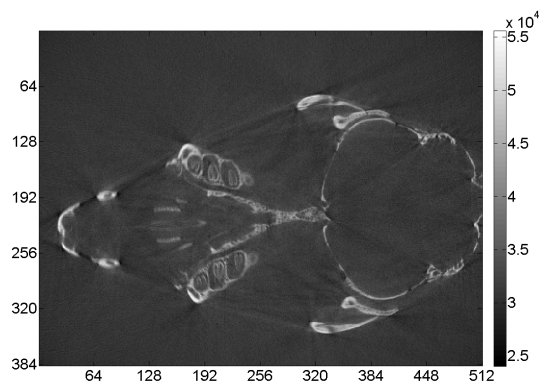
48. ábra

A koponyáról készült rekonstrukció közepső, 256. szeletén illusztráljuk a szűrőfüggvény ω_{z_0} paraméterének változását 0,05 és 150 között. Azt tapasztaltuk, hogy kis

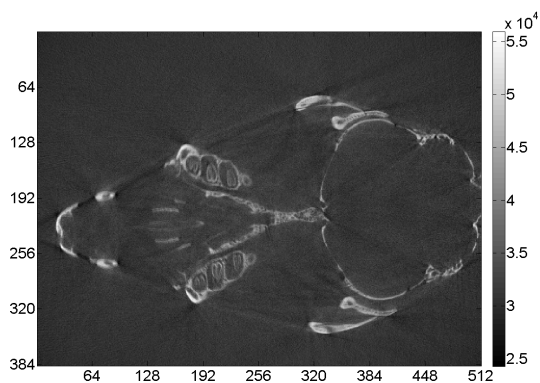
ω_{z_0} értékeknél minimális mértékű elmosódás történik, míg a Nyquist frekvencia fölött várakozásainknak megfelelően egyre jelentősebb zaj jelenik meg (49. ábra: $\omega_{z_0} = 0,05$; 50. ábra: $\omega_{z_0} = 1$, 51. ábra: $\omega_{z_0} = \pi$, 52. ábra: $\omega_{z_0} = 10$, 53. ábra: $\omega_{z_0} = 150$).



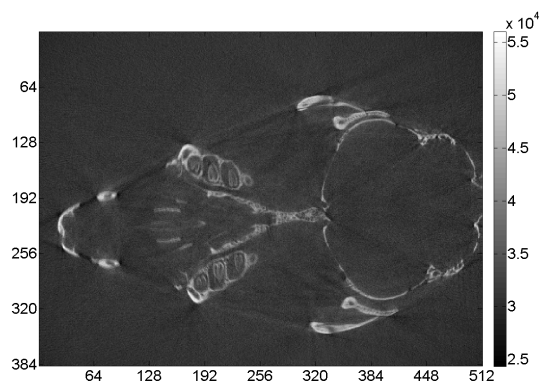
49. ábra



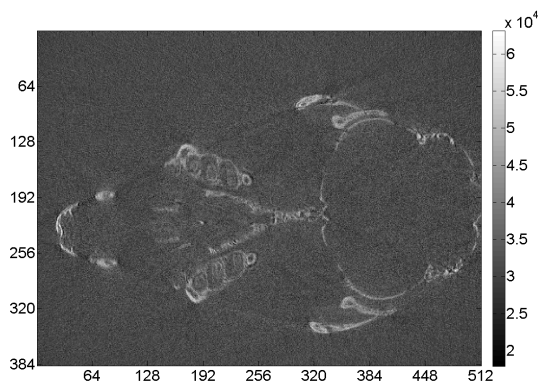
50. ábra



51. ábra

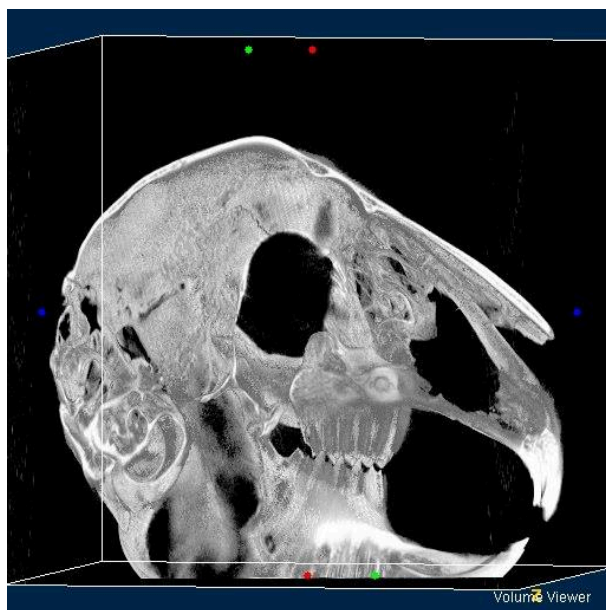


52. ábra



53. ábra

A koponyáról készült háromdimenziós vetületi kép látható az alábbi ábrán (54. ábra).



54. ábra

ÖSSZEFOGLALÁS, KITEKINTÉS

A dolgozat során létrehoztam egy GPU-alapú 3D cone-beam rekonstrukciós szoftvert, melyet sikeresen teszteltem mérési adatokon.

A feladat megoldása során készítettem C nyelven egy fan-beam szinogram készítő szoftvert, amellyel szimulálhatók a mérési adatok. Ezáltal a 2D rekonstrukciós szoftvert matematikai fantomokkal is tudtuk tesztelni. Létrehoztunk egy GPU-n futó, CUDA nyelven írt fan-beam rekonstrukciós szoftvert, amely a videokártyás párhuzamosításnak köszönhetően kb. 1 másodperc alatt állít elő több száz projekciót tartalmazó szinogramból egy 1024x1024 pixel méretű rekonstruált képet. Ezt továbbfejlesztettük három dimenzióra, a szintén GPU-n futó cone-beam rekonstrukció általában gigabájt nagyságrendű adatmennyiségből alkot néhány perc alatt 512x512x512 voxel méretű háromdimenziós tomográfiás képet. A rekonstrukciós szoftvereket a BME Nukleáris Technika Tanszéken található NTIuCT berendezés által biztosított mérési adatokkal teszteltük.

A szoftver futásideje tovább rövidíthető a memóriakezelés optimalizációjával. Mivel a gain korrekció is egy könnyen párhuzamosítható feladat, ezért kézenfekvő volna ezt is GPU-ra implementálni, esetleg a rekonstrukció részévé tenni. Elméletileg nem szükségesek redundáns adatok a rekonstrukcióhoz, elegendő volna valamivel több, mint 180°-ról készült projekciókat felhasználni, ezáltal a mérési idő is csökkenthető volna.

Nem volt lehetőségünk a tanszék további eszközei (Optikai CT, Gamma Tomográfia, MRI) által mért adatok rekonstruálására, ezért a jövőben érdemes volna ezekkel az adatokkal is tesztelni a szoftvert.

KÖSZÖNETNYILVÁNÍTÁS

A szoftverek fejlesztése és tesztelése során kapott segítséget ezúton is köszönöm témavezetőmnek, Légrády Dávidnak. A mérési adatokat Kleizer Gábornak és Tölgyesi Botondnak köszönhetjük, hiszen a NTIuCT berendezés megépítése elsősorban Gábor érdeme, a geometriai paraméterek és gain-korrekció pedig Botond munkája. Továbbá köszönöm, hogy részt vehettem a Légrády Dávid által életre hívott „CT-szakkörökön” Tölgyesi Botond, Kleizer Gábor, Molnár Balázs, Rosta Gergely, Berze Noémi és Hegedüs Tamás társaságában, akik a CT képalkotás valamely területével foglalkoznak, illetve foglalkoztak az elmúlt időszakban.

IRODALOMJEGYZÉK

- 1. L. A. Feldkamp, L. C. Davis, J. W. Kress, 1984.** Practical cone-beam algorithm. Optical Society of America, Vol. 1, No. 6, 612-619
- 2. Dufan Wu, Liang Li, Li Zhang, Yuxiang Xing, Zhiqiang Chen, Yongshun Xiao, 2011.** Geometric Calibration of Cone-beam CT with a Flat-panel Detector. Nuclear Science Symposium Conference Record, 2952-2955
- 3. NVIDIA CUDA, 2014.** NVIDIA CUDA C Programming Guide
- 4. Kári Béla, Karlinger Kinga, Légrády Dávid, Bérczi Viktor, Czifrus Szabolcs, 2011.** OF Tankönyv