



Diplomamunka

**Mikro-CT készülék rekonstrukciós és kalibrációs környezetének
létrehozása**

Marinovszki Árpád

Témavezető: Dr. Légrády Dávid
egyetemi docens
BME Nukleáris technika Intézet

BME

2016

Diplomamunka feladat a Fizikus mesterképzési (MSc) szak hallgatói számára

A hallgató neve: **Marinovszki Árpád**

szakiránya: **orvosi fizika**

A diplomamunkát gondozó (a záróvizsgát szervező) tanszék:

Nukleáris Technika Tanszék

A diplomamunka **Budapesti M° szaki és Gazdaságtudományi Egyetem**
készítésének helye:

A témavezető neve: **Dr. Légrády Dávid**

– munkahelye: **BME NTI**

– beosztása: **docens**

– e-mail címe: **legrady@reak.bme.hu**

A konzulens neve:

(külső téma vezető esetén kijelölt tanszéki munkatárs)

– beosztása:

– e-mail címe:

A diplomamunka **Mikro-CT készülék rekonstrukciós és kalibrációs környezetének létrehozása**
címe:

azonosítója:
DM-2016-45

A téma rövid leírása, a megoldandó legfontosabb feladatok felsorolása:

A BME Nukleáris Technikai Intézetében rendelkezésre álló mikro-CT készülék kalibrációs és rekonstruktív szoftverrendszeré alkalmas tomográfiai felvételek készítésére és a képrekonstrukció elvégzésére. A vonatkozó algoritmusok érettségi foka azonban még alacsony szint^o, az eddig megvalósított eljárások a lehető legegyszer^o bb és olykor instabil eljárások. A hallgató feladata egységes, grafikus kártyát (GPU) is igénybe vevi kalibrációs és rekonstruktív szoftver- és algoritmuskönyezet létrehozása. A hallgató feladatai:

- A geometriai kalibráció képfelismerési algoritmusának kidolgozása, robosztus algoritmus létrehozása és implementálása, melynél a geometriai kalibráció felhasználói beavatkozás nélkül, automatikusan elvégezheti. A hallgató feladata a geometriai kalibráció robosztusságának vizsgálata a becsült geometriai paraméterekre nézve és a rekonstruált kép tekintetében is.
- A gain kalibráció robosztus algoritmusának kialakítása és implementálása a halott pixelek és csillanó pontok felhasználói beavatkozást nem igénylő korrekciója.
- Az új szoftverkönyezettel készített felvételek minőségi elemzése, a készülék felbontásának vizsgálata.
- A determinisztikus rekonstruktív felkeményedés-korrekciónak implementálása és összehasonlítása Monte Carlo alapú rekonstruktívval.

A feladat kiadásának időpontja: **2016.02.19.**

Téma vezető vagy tanszéki konzulens aláírása:

A diplomamunka témaírását jóváhagyom
(tanszékvezető aláírása):



Önállósági nyilatkozat

Alulírott Marinovszki Árpád a Budapesti Műszaki és Gazdaságtudományi Egyetem fizikus MSc szakos hallgatója kijelentem, hogy ezt a diplomamunkát 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 vettettem, a forrás megadásával jelöltem.

Budapest, 2016. június 6.

aláírás

Tartalomjegyzék

1. Bevezetés	4
2. Gain korrekció	4
2.1. A gain korrekció által javított artefaktumok bemutatása	4
2.2. A gain korrekció algoritmusai	6
2.3. A gain kalibráció implementálása	9
2.4. A gain korrekció implementálása	12
3. A geometria kalibráció	13
3.1. A síkpaneles CBCT geometriája	13
3.2. A geometriai paraméterek meghatározásának elmélete	15
3.2.1. A paraméterek meghatározása Noo[2] cikke alapján	15
3.2.2. A paraméterek meghatározása Wu[3] cikke alapján	17
3.3. A golyók középpntjának meghatárzoása	19
Hivatkozások	20

1. Bevezetés

2. Gain korrekció

A gain korrekció az elkészült felvétteleken elsődlegesen elvégzendő korrekció, amely felelős egyrészt azért, hogy a detektorpixelek eltérő érzékenysége és zaja következtében fellépő képhibákat eltüntesse. Továbbá, hogy korrigálja a röntgennyaláb mért intenzitásában történő azon gyengülést, amely pusztán azért következik be, mert a detektorpixelek más–más távolságra vannak a forrástól.

A fejezet elején megmutatom, hogy milyen hibákat eredményez a gain korrekció elhanyagolása. Ez után ismertetem a korrekció elvégzéséhez szükséges algoritmusokat, majd bemutatom, hogy implementáltam ezeket az elkészült programban.

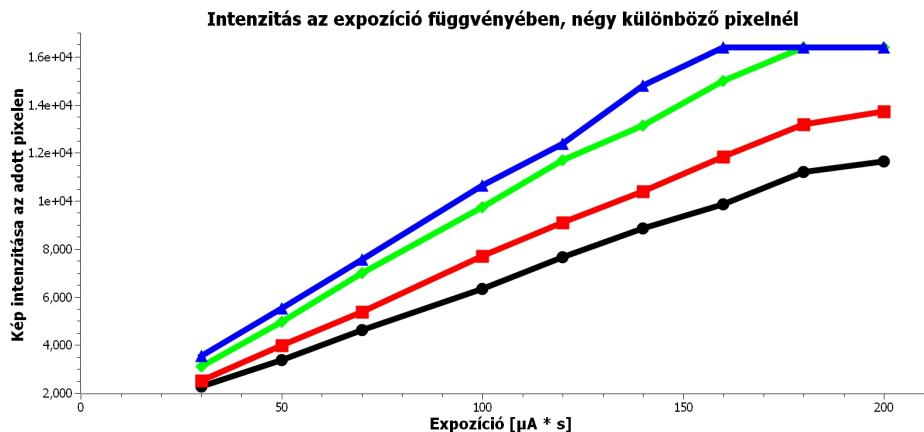
2.1. A gain korrekció által javított artefaktumok bemutatása

A gain korrekció által kiszűrt hibák három részre oszthatóak. Egyrészt belátható, hogy a sík detektoron mérhető fluxus értéke nem egyenletes. Ha olyan felvételt tekintünk, ahol nincs leképezendő objektum, belátható, hogy a legnagyobb fluxus a detektor azon pixelét fogja érni, amely a legközelebb van a röntgenforrás fókuszpontjához. Az ettől távolabbi pixeleken mérhető intenzitás pusztán azért is kisebb lesz, mert a forráspontról távolabb vannak, hiszen a mérhető intenzitás a fókuszponttól számított távolság négyzetének inverzával arányos. A jelenség következtében tehát a detektorpixelek intenzitása a szélek felé csökken, és a csökkenés mértéke az expozíciós beállításuktól – úgy mint az expozíciós idő, valamint a röntgenső feszültsége és árama – független. A kialakult képtörzulást *flatness hibának* nevezzük, utalva arra, hogy ez azért következik be, mert a detektorunk sík – gömbfelületű detektornál ez a hiba nem lépne fel.

Megállapítható továbbá, hogy a detektorpixelek eltérő érzékenysége miatt is fellépnek képhibák. Ezen hibákat két csoportra lehet bontani: *offset hibára*, valamint *gain hibára*. Az elnevezés igen szemléletes. Válasszunk ki a detektoron egy adott pixelt, és vizsgáljuk ennek a pixelnek az mért intenzitását, miközben változtatjuk az expozíciós időt! Az adott pixelen a mért intenzitás – expozíciós idő függvényt ábrázolva olyan görbét kapunk, amely kezdetben igen jó közelítéssel egyenest mutat, míg nem az intenzitás egy adott értéket elérve nem nő tovább. Utóbbi esetet, azaz, amikor az intenzitásérték tovább nem nő, szaturációval hívjuk. Az előbbihez hasonlóan, ha állandó expozíciós idő mellett a röntgenső áramerősségét változtatjuk, és így nem a felvétel idejét növeljük, hanem a detektorra eső nyalábintenzitást, szintén kezdetben lineáris, majd telítődő összefüggést kapunk a pixel által mért intenzitás – alkalmazott áramerősségi függvény tekintetében. Éppen ezért az expozíciós idő és a röntgenső áramerősségének

szorzatának – a továbbiakban *expozíció* függvényében szokás vizsgálni az adott pixel intenzitását. Az így kapott görbe, azaz az intenzitás az expozíció függvényében, adja meg az adott pixel érzékenységét.

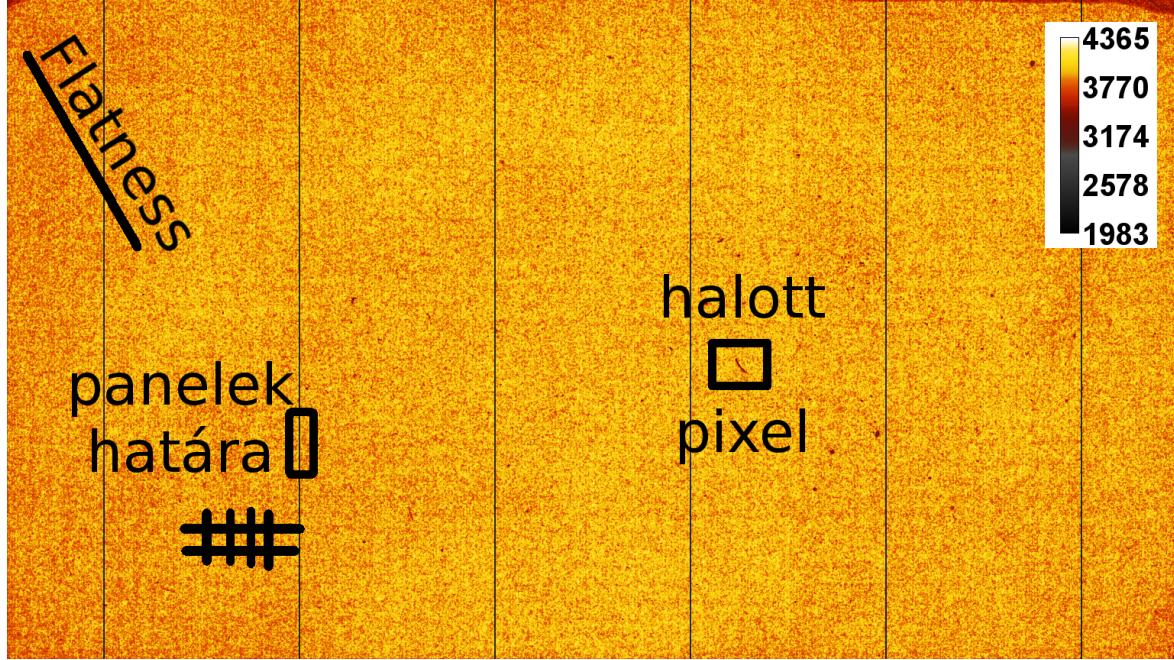
Ez az érzékenység azonban pixelről pixelre változik. Különböző pixelek érzékenységét megvizsgálva észrevehetjük, hogy az egyes érzékenységi görbék lineáris szakaszának meredeksége és tengelymetszete is változik. Változik továbbá a szaturációs expozíció is, azaz az az expozíció érték, amelynél az adott detektor telítésbe megy. Ezt szemlélteti az 1. ábra, amelyen négy különböző pixel által mért intenzitást ábrázoltam, az expozíció függvényében. Az ábrán jól látszódik, hogy a különböző pixelek érzékenysége eltérő meredekségű és tengelymetszetű egyenessel jellemzhető, valamint a szaturációs expozíciójuk is eltérő. Az offset és gain korrekciók a detektorpixelek ezen érzékenységbeli különbségeit korrigálják. Az *offset hibák* javítása jelenti a lineáris szakaszok eltérő tengelymetszetőből származó eltérések kiküszöbölését. Az offset hibák nagysága nem függ tehát az expozíciótól, de függ az expozíciós időtől. Ugyanis kikapcsolt röntgenforrás mellett is detektálunk valamekkora háttérsugárzást, illetve a detektorok zaja is mérhető intenzitást fog eredményezni a felvételen. A teljes, röntgenforrás nélkül mért jel tehát arányos lesz az expozíciós idővel és függeni fog a pixelek érzékenységétől is. A *gain hibák* javítása jelenti az érzékenységek lineáris szakaszainak eltérő meredekségéből, valamint az eltérő szaturációs expozíciókból származó hibák javítását. Ezek tehát az expozíciós beállításoktól függő hibák.



1. ábra. Négy kiszemelt pixel intenzitásának változása az expozíció függvényében.

Az egyes hibák hatását az elkészült felvételeken a 2. ábrán szemléltetem. Az ábrán egy olyan felvétel látható, amely készítésekor a detektor és a forrás között leképezendő objektum nem volt, továbbá a felvétel mindenféle korrekciótól mentes, azaz a mérőszoftver ezt a képet olvasta ki a detektorból. Az ábrán látható, hogy a leképezendő objektum hiányában homogénnek várt kép közel sem homogén. Általánosan megfigyelhető, hogy az egyes pixelek értékei jócskán szórnak, továbbá látható, hogy a képen

négyzetrácsos struktúra rajzolódik ki, amelynek oka valószínűleg az detektorpanelek határainál megváltozik a pixelek érzékenysége. Különösen határozott ez az eltérés a képen látható hat függőleges egyenes mentén. Ezen pixelek az összes felvételen alacsonyabb intenzitással rendelkeznek társaiknál és értékük is kevésbé változik az expozíció növelésével, mint a többi pixel. Megfigyelhető továbbá, hogy a kép egyre sötétebb a sarkok felé, amely a flatness hibáak tudható be.

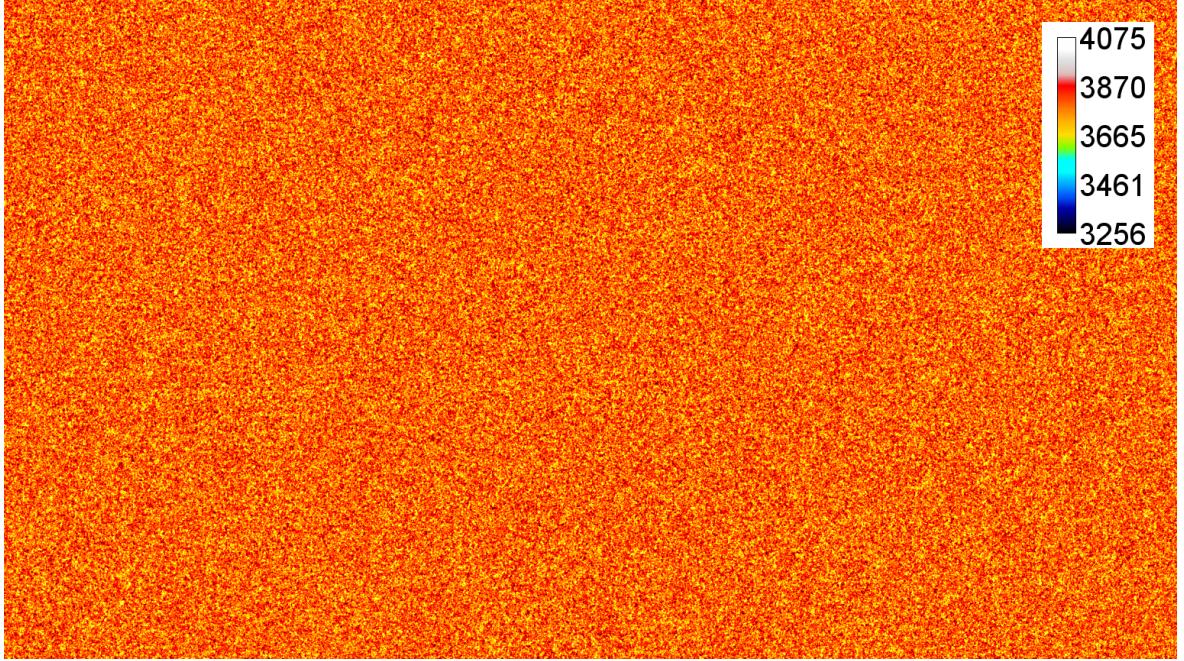


2. ábra. Tárgy nélkül készített felvétel, gain korrigálás nélkül.

A gain korrekció során tehát ezen hibákat távolítjuk el a képről. Gain korrekció után a fenti képből a 3. ábrán látható képet kapjuk. Megfigyelhetjük, hogy itt nem látszódnak az előbb jelölt artefaktumok. Bár a kép továbbra sem teljesen homogén – ezt nem is várjuk el, valamekkora zaj mindenkorában terhelni fogja a méréseinket –, a pixelek intenzitásának szórása jóval lecsökkent, megszűnt a gain korrekció nélküli képet átható négyzetes struktúra, és a kevésbé érzékeny pixelek sem lógnak ki a többi közül. A következőkben a gain korrekcióhoz szükséges algoritmusokat ismertetem.

2.2. A gain korrekció algoritmusai

A felvételek gain korrigálásához először is meg kell határozni az egyes detektorpixelekhez tartozó érzékenységeket, azaz azt, hogy az expozíció növelésével milyen gyorsan változik a pixel intenzitása, valamint azt, hogy az adott pixel milyen expozíció melett telítődik. ezen kívül meg kell határozni, hogy megvilágítás nélkül hogyan változik az adott pixel intenzitása az expozíciós idő függvényében. Ehhez először kalibráló képsorokat kell készíteni. Az offset hibák javításához a röntgenforrás bekapcsolása nélkül,



3. ábra. Tárgy nélkül készített felvétel, gain korrigálás után.

különböző expozíciós időkkel, a gain és flatness hibák javításához bekapcsolt forrással, különböző expozíció értékek mellett.

A képeket először az offset hibáktól kell megszabadítani. Az ehhez szükséges korrekciós faktorok meghatározásához tehát különböző expozíciós időkkel, kikapcsolt forrással készítünk képeket. Az egyes pixelek értékét az expozíció idő függvényében ábrázolva olyan görbét kapunk, amelyre jól illeszthető egyenes. Ez alapján az i . pixel által, forrás nélkül mért I_i^{offset} intenzitás az (1) képlettel írható le, ahol a_i^{offset} és b_i^{offset} a fenti módon előállított képsorozatból, egyenes illesztéssel megkapható és pixelenként változó.

$$I_i^{\text{offset}} = a_i^{\text{offset}} \cdot t_{\text{exp}} + b_i^{\text{offset}} \quad (1)$$

Minthogy ezek az intenzitásértékek a detektort érő hasznos jeltől függetlenül megjelennek, először ezen értékeket kell levonni az elkészült képekből.

Az offset hibák korrigálása után tehát az i . pixel intenzitása a (2) képlet szerint alakul, ahol $I_i^{\text{mérő}}$ az i . pixel eredeti, korrekció nélküli intenzitása, $I_i^{\text{offset korrigált}}$ pedig az offset hibától mentes érték.

$$\begin{aligned} I_i^{\text{offset korrigált}} &= I_i^{\text{mérő}} - I_i^{\text{offset}} \\ &= I_i^{\text{mérő}} - \left(a_i^{\text{offset}} \cdot t_{\text{exp}} + b_i^{\text{offset}} \right) \end{aligned} \quad (2)$$

Ez után javíthatjuk ki a gain és flatness hibákat. A két hiba valójában egy algoritmussal javítható, hiszen a flatness hiba felfogható úgy, mintha a detektor pixeleinek

érzékenysége lenne más a detektor más–más területein. A szükséges korrekciós faktorok meghatározásához olyan kalibráló képsorokat kell készítenünk, amelyek különböző expozícióval készülnek. Az 1. ábrán látottak alapján megállapítható, hogy az egyes detektorpixelek intenzitása kezdetben az expozícióval lineárisan változik, vagyis egyenest tudunk rá illeszteni. Amennyiben tehát nincs tárgy a detektor és a forrás között, az i . pixel által, a detektort érő sugárzás, mint hasznos jel által okozott (tehát offset hibától mentes) $I_i^{\text{offset korrigált}}$ intenzitás értéke a (3) egyenlet szerint számolható az expozícióból. Itt a_i^{gain} és b_i^{gain} jelöli a különböző expozíciós értékek mentén felvett intenzitásértékekre illesztett egyenes paramétereit, E pedig az expozíció értékét, amely a fent leírtak alapján az expozíciós idő és a röntgensű áramának szorzata.

$$\begin{aligned} I_i^{\text{offset korrigált}} &= I_i^{\text{mért}} - I_i^{\text{offset}} \\ &= a_i^{\text{gain}} \cdot E + b_i^{\text{gain}} \end{aligned} \quad (3)$$

Egy olyan felvételen, amely készítésekor valamilyen tárgyat helyezünk a detektor és a forrás közé, a mért intenzitásértékek nyilván el fognak térní a detektor különböző részein, ahogy a leképezni kívánt tárgy lineáris gyengítési együtthatója is eltér a különböző térfogatokban. Ezt úgy vehetjük figyelembe, hogy a (3) képletben az E expozíció érték helyére egy helyfüggő, E_i^{virt} virtuális expozíciót vezetünk be. Ezen érték úgy módosul az eredeti E expozícióhoz képest, hogy figyelembe veszi a röntgen nyaláb gyengülését, miközben az áthalad a leképezni kívánt tárgyon. Vagyis egy adott mérés során az E_i^{virt} virtuális expozíció nem lesz más, mint az a (valós) expozíció érték, amely esetén ugyan akkor intenzitást mértünk volna tárgy nélküli elrendezésben, mint jelen esetben a leképezendő tárggyal. Ezzel a jelöléssel kapjuk a (4) egyenletet, amely immáron általánosan igaz, egy tetszőleges felvételre – míg a (3) egyenlet csak arra az esetre érvényes, amikor nincs semmi a detektor és a forrás között.

$$\begin{aligned} I_i^{\text{offset korrigált}} &= I_i^{\text{mért}} - I_i^{\text{offset}} \\ &= a_i^{\text{gain}} \cdot E_i^{\text{virt}} + b_i^{\text{gain}} \end{aligned} \quad (4)$$

A gain hibák javítása során azt szeretnénk elérni, hogy az egyes pixelek intenzitása ne függjön az adott pixel érzékenységtől, azaz az a_i^{gain} és b_i^{gain} együtthatóktól, csak az adott pixelt ért – virtuális – expozíciótól. Vagyis azt, hogy ha két detektorpixelt egy felvétel során ugyanolyan intenzitású sugárzás ér, és az expozíciós idő is megegyezik, akkor a két pixel intenzitása az elkészült képen is azonos legyen. Ezt úgy végezzük el, hogy először meghatározzuk az egyes detektorpixelek virtuális expozícióját, majd ezekből úgy alakítjuk ki a gain korrigált intenzitásértéket, mintha minden pixel érzékenysége megegyezne.

Először tehát kiszámoljuk az adott offset korrigált intenzitáshoz tartozó virtuális expozíció értékét a (4) egyenlet segítségével. Egyszerű átrendezéssel kapjuk a virtuális expozícióra az (5) egyenletet.

$$E_i^{\text{virt}} = \frac{I_i^{\text{offset korrigált}} - b_i^{\text{gain}}}{a_i^{\text{gain}}} \quad (5)$$

A virtuális expozíció tehát azzal van összefüggésben, hogy mekkora valós fluxus érte az adott pixelt, és mentes az adott pixel érzékenységek torzító hatásától. Ebből úgy határozzuk meg a gain korrigált intenzitás értéket, mintha minden pixel érzékenysége, azaz intenzitás – virtuális expozíció függvénye egy origón átmenő egyenes lenne, mindenek azonos meredekséggel. A szóban forgó meredekséget pedig úgy határozzuk meg, hogy éppen annál az expozíciós értéknél telítődjön, amelynél a legérzékenyebb pixel már telítődik. Azaz meg kell vizsgálni, hogy az expozíciót növelte mikor kapunk olyan képet, amin már van szaturált pixel. Ezen expozíciót hívjuk szaturációs expozíciónak (E^{szat}). Ezen érték a kalibráló adatsorból szintén meghatározható paraméter. Ezen paraméter ismeretében a gain korrigált $I_i^{\text{gaintorrigált}}$ pixelintenzitás a (6) egyenlettel számolható, ahol $(2^{14} - 1)$ a detektor maximális intenzitásértéke.

$$I_i^{\text{gaintorrigált}} = E_i^{\text{virt}} \cdot \frac{2^{14} - 1}{E^{\text{szat}}} \quad (6)$$

A módszer hatékonyságát a kísérleti algoritmus létrehozása során Botond részletes tesztekkel vizsgálta, ezért én ettől eltekintek. A továbbiakban azt mutatom be, hogy ezeket az algoritmusokat hogyan építettem bele az elkészült szoftverbe, illetve milyen mérési utasítást ajánlok a korrekciók elkészítéséhez.

2.3. A gain kalibráció implementálása

Mint ahogy már említettem a kalibrációs szoftver létrehozásakor szempont volt, hogy annak sebessége gyors, kezelése egyszerű legyen, így létrehoztam egy grafikus felülettel rendelkező programot, amelyben a felhasználó nyomógombokkal tud választani az elérhető funkciók között. Az első ilyen megvalósított funkció az offset és gain hibák korrigálására szolgáló együtthatók meghatározása. Ezt a funkciót választva a szoftver egy új ablakban kéri a felhasználót, hogy válassza ki az elkészített kalibrációs mérések mappáját, valamint azt a mappát, ahova a kimeneti fájlokat menteni szeretné. Ez után a korrekciós faktorok számolása automatikusan megtörténik, illetve az esetleges hibákról a felhasználó értesítést kap a konzolon.

A bemeneti fájlok elkészítéséhez, azaz a mérési utasításhoz szükséges megjegyezem, hogy a mérések során észrevehető volt, hogy a detektor által készített képek

egy része valamiért jóval sötétebb, mint az azonos körülmények között készült egyéb képek. Ez a hiba a képek kb. 5%-át érinti, és okát nem sikerült felderíteni. Más felhasználók is panaszkodtak hasonló hibára ennél a detektortípusnál. A korrekciós faktorok számításánál azonban kimondottan fontos, hogy ilyen képeket ne használjunk fel a számításokhoz – továbbá az ilyen képek kezelése egyéb helyzetekben is szükségszerűnek tűnik. Ennek érdekében a gain kalibrációhoz szükséges méréseket úgy végeztük el, hogy egy adott beállítással több képet készítettünk (10–20 képet beállításoknál). Az azonos beállításokkal készült képeket külön mappákba mentettük, így a szoftver is ilyen formában várja. Ez a megoldás azért szerencsés, mert egy mappán belül átlagolva a képek intenzitásértékét, könnyen és egyértelműen kiszűrhető, ha valamelyik kép jóval sötétebb, a fent említett hiba miatt. Továbbá az azonos mappákban lévő, tehát azonos beállításokkal készült képeket átlagolva a képeket terhelő zaj is csökken. Így amikor a korrekciós faktorokat számolom, mappánként átlagolom azokat a képeket, amelyek átlagos intenzitása nem tér el 10%-nál jobban a mappában lévő képek átlagos intenzitásának átlagától. Az program robusztusságát tovább növelve, a képek beolvasásakor vizsgálom, hogy azok milyen expoziós beállításokkal készültek, és ez alapján is szelktálom a képeket, kiszűrve a mérés során elkövetett emberi hibákat. A mérésvezérlő szoftverről ugyanis tudni érdemes, hogy a felvételek készítésének elindítása a röntgen forrás bekapcsolásától függetlenül elindítható, és az indítás után sorozatosan készít képeket, amíg le nem állítják. Mérés közben a forrás feszültsége és áramerőssége szabadon változtatható, illetve a forrás ki- és bekapcsolható. A röntgenső független kezelése részben biztonságvédelmi célokra szolgál, azonban lehetőséget ad a felhasználónak, hogy a képek sorozatos kimentését elindítsa úgy, hogy a forrás nincs bekapcsolva, vagy a beállított paraméterei nem megfelelők. Ekkor gyakran előfordul, hogy a felhasználó a paramétereket menet közben állítja, és az addigi eredményeket nem törli ki. Ezt kiküszöbölgendő, a gain korrekcióhoz készített kalibrációs képek feldolgozása során a képek átlagos intenzitásán kívül a program vizsgálja az átlagos áramerősséget, feszültséget és exponálási idő paramétereket is, és a kilógó értékkel készült képeket figyelmen kívül hagyja. Továbbá figyelembe veszi azt is, hogy offset korrekcióhoz készült kalibráló képek között ne legyen olyan, ahol a forrás be volt kapcsolva – vagy ha van, akkor hagyja azt is figyelmen kívül. Hasonlóan a gain és flatness hibák korrigálásához készült képeknél ignorálja az algoritmus azokat a képeket, ahol a forrás ki volt kapcsolva. Amennyiben egy mappában kevés kép van, amely a fenti szűrknek megfelel (5nél nem több), akkor a mappa tartalmát a szoftver nem használja fel a korrekciós faktorok meghatározásához. Továbbá, amennyiben túl kevés mappát talál a szoftver, tehát túl kevés a különböző expoziós beállítások száma a korrekciós faktorok meghatározásához – akár összesen, akár túl kevés megy át a fenti szűrükön–, a kalibrációs faktorok nem kerülnek

meghatározásra. Így elkerülhető, hogy túl kevés pontra illesszünk egyenest, és ezért hibás eredményeket kapjunk, vagyis a képkészlet ellenőrzése szintén a szoftver robusz-tusságát növeli. A felvételek elkészítésekor alkalmazott feszültségértékeket az program szintén a képek beolvasásakor határozza meg, így azt külön megadni nem szükséges, mint ahogy a képeket feszültség szerint sem kell szétválogatni – ellentétben a kísérleti algoritmussal.

A képek beolvasása és mappánkénti átlagolása után a pixelenkénti egyenes illesztést a szokásos, egyenes illesztésére vonatkozó (7) képleteket felhasználva oldottam meg, ahol az x és y mérési pontokra illesztjük az α és β paraméterekkel jellemzett egyenest. A képletben a felülvonás az átlagolást jelenti.

$$\begin{aligned}\beta &= \frac{\bar{xy} - \bar{x} \cdot \bar{y}}{\bar{x^2} - \bar{x}^2} \\ \alpha &= \bar{y} - \beta \cdot \bar{x}\end{aligned}\tag{7}$$

Jelen esetben x értéke az expozíciós idő (offset hibáknál), vagy az expozíció (gain és flatness hibáknál), y pedig az adott pixel intenzitása. Az egyenes illesztésére egyszerű és gyors megoldásnak találtam, hogy az illeszteni kívánt képeket pixelemként összeadom, valamint ugyanezt megteszem az adott expozícióval – vagy expozíciós idővel – megszorozva is, majd ezeket a képeket pixelenként elosztom az összeadott képek számával. Így az \bar{x} és \bar{xy} értékek pixelenként előállnak, mint egy–egy kép. Ez után a kettő különbségét – mint képet – leosztva a $\bar{x^2} - \bar{x}^2$ skalárral, közvetlenül előáll a pixelekre illesztett egyenes meredeksége, és a tengelymetszet további képkivonással kapható. Mivel a szükséges képműveletek – összeadás, skalárral szorzás, kivonás – könnyen párhuzamosíthatóak, gyakorlatilag az egyenes illesztése is párhuzamosan történik, így jelentősen lerövidítve a számoláshoz szükséges időt. A módszer egyébként a memóriával is takarékosabban bánik, mint a kísérleti algoritmus, amely először beolvassa az összes képet a memóriába, majd pixelenként illeszt egyenest. Az ottani megoldással ellentétben, ahol a szükséges memória lineárisan nő a képek számával, nálam a szükséges memória a képek számától független és alacsony.

A fenti megoldáshoz a grafikus felületet és a mappakezelést a Qt biztosítja, míg a gyors képműveletek CUDA kód alkalmazásával valósulnak meg. A képfeldolgozáshoz létrehoztam olyan kép osztályt, amely képes magát fájlból a GPU memóriájába olvasni, és alkalmas a képeken végzett műveletek gyors végrehajtására a grafikus kártyán. A pixelenkénti összeadás, kivonás, skalárral szorzás megoldása viszonylag triviális, itt egy szál egy pixel értékét számolja. Ezzel ellenben a maximum, minimum, átlag, és szórás – utóbbi a gain korrekcióhoz nem használtam fel, de technikailag itt kell megemlítenem – számolása első ránézésre nem triviális, ezért néhány mondatban bemutatom.

Az előbb felsorolt műveletek, amelyek eredménye előáll a kép pixeleiből úgy, hogy minden egyes pixel értékét egyszer használjuk fel, egy kommutatív, asszociatív, két elem között ható – programozási értelemben vett – operátorral, az angol irodalom *reduce* műveletnek hívja. Ilyen az előbb említett összeadás, minimum és maximum számítás is. Például három elem, a , b és c összegét megkapjuk, ha a és b összegéhez hozzáadjuk c : $\text{sum} = (a+b)+c$. Az összeadást kicserélhetjük maximum számításra is, az algoritmus ekkor csak a számok közé rakott operátorban tér el: $\text{maximum} = \max(\max(a, b), c)$). Az ilyen, *reduce* típusú műveletek optimális használatát az összeadás példáján keresztül írom le, de a fentiek értelmében a minimum és a maximum számolás is hasonlóan kivitelezhető.

Az alkalmazásomban is használt reduce műveletet a CUDA dokumentációjához mellékelt példaaz [1] alapján állítottam össze. Az ilyen műveletek általános megoldása – az összeadás példáján –, hogy a képet két részre osztjuk, majd az egyik részt hozzáadjuk a másikhoz. A műveletet folytatjuk, mindig felezve a fennmaradó képrészletet, míg végül a kép egyik pixelében előáll az összes pixel összege. A már említett példa ezen túl olyan technikákat is ajánl, mint a *shared memory*, azaz az összes szál által hozzáférhető (de kívülről nem hozzáférhető), gyors elérésű memória használata – a lassú elérésű, globális memória helyett, ahol a képek eleve tárolva vannak. Továbbá figyelembe veszi, hogy a párhuzamos futtatás során a szálak 32 darabos „csomagokban”, úgynevezett *warpokban* futnak, ezzel tovább optimalizálva az algoritmust. Míg ha több, mint 32 szálat futtatunk egyszerre, semmi nem garantálja, hogy azok egyszerre fognak futni. Ilyenkor szakaszonként be kell várnia a szálaknak egymást. Például ha a kép egyik felét hozzáadjuk a másikhoz, csak akkor felezhetjük tovább az elkészült képet, ha minden összeadás megtörtént. Amikor már 32-nél kevesebb szálat használunk, akkor a szálaknak nem kell egymást bevárni, hiszen biztosan nem fut több. Továbbá a példa alapján az összeadáshoz vonatkozó kerneleket C++ template-ekkel írtam meg, amely segítségével már fordítási időben tudja a kernel, hogy adott méretű kép esetén hogyan kell felosztani a képet – nem pedig futási időben kell vizsgálnia, hogy a kép mekkora méretű.

2.4. A gain korrekció implementálása

A korrekciós faktorok meghatározása után, amelyek tehát szintén képként kerülnek mentésre, a felhasználónak lehetősége nyílik az elkészített projekciót offset és gain korrigálni, az erre vonatkozó (2) és (6) képleteket felhasználva. Ezek újabb képösszeadásokat és skalárral való szorzást, osztást jelentenek, így az eddig bemutatott függvényekkel gyorsan elvégezhetők. A felhasználónak ennél a funkcionál is grafikus felületen van lehetősége kijelölni a korrigálni kívánt mappát, amelyből a program az összes képet

gain korrigálja és elmenti egy kívánt, kijelölt mappába – akár ugyan abba.

3. A geometria kalibráció

A diplomaunkám során a másik tárgyalt kalibráció az úgynevezett *geometria kalibráció*, amely a CT-vel elkészített képek alapján állapítja meg, hogy milyen távol van egymástól a detektor, a forgatható asztal forgástengelye és a röntgenscső fókuszpontja. A geometriai adatokhoz igen pontosan szükség van, amikor az elkészült projekciók alapján meghatározzuk a felvett tárgy 3 dimenziós képét, vagyis a visszavetítéskor. Ugyanakkor a geometriai paramétereket kézzel nem tudjuk igen pontosan lemérni, már csak azért sem, mert a röntgenscső fókuszpontjának a helye nem hozzáférhető és nem ismert, továbbá a mérési pontatlanság is viszonylag nagy lenne. Ezért alkalmazzuk azt a módszert, hogy az elkészült képek alapján határozzuk meg a szóban forgó paramétereket.

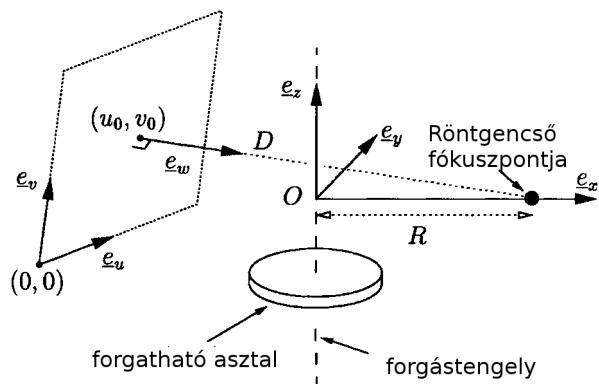
A kalibrációt a kísérleti algoritmus által is használt cikkek alapján kiviteleztem. Mind Noo cikke [2], mind Wu cikke [3] hasonló módszerekkel határozza meg a keresett paramétereket, azonban utóbbi egy egyszerűsített esete az előbbinek. A továbbiakban bemutatom a geometriai kalibráció meghatározásának elméletét a cikkek alapján, majd bemutatom, hogy a geometriai adatok meghatározásához milyen módszereket használt a kísérleti algoritmus, valamint az én algoritmusaim, továbbá ismeretem az elérte eredményemet.

3.1. A síkpaneles CBCT geometriája

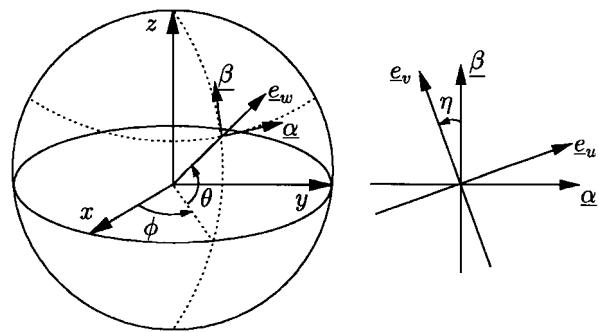
Hogy a geometriai kalibráció során milyen adatok kerülnek meghatározásra, és milyen adatok kerülnek elhanyagolásra, azt jól szemlélteti a (4). és az (5). ábra.

A 4. ábrán látható paraméterek jelentése a következő. A forgatható asztal forgástengelye és a röntgenscső fókuszpontjának távolsága R . Konvenció szerint a forgástengely a z tengellyel párhuzamos és az x tengely irányában található a forrás fókuszpontja. A detektor az (u_0, v_0) koordinátákkal jellemzett pixelétől mért távolsága a legkisebb a forrástól, és ez a távolság D .

Az 5. ábrán azt láthatjuk, hogy míg a forgástengely és a forrás iránya a modell szerint rögzített, a detektor irányát és állását meghatározó ϕ, θ, η szögeket az ábrák nem hanyagolják el. Ezek közül az η szöget, vagyis a detektornak a saját síkjában való elfordulását minden két cikk figyelembe veszi és számolja. A ϕ szöget, vagyis a detektor kifordulását, az asztal forgástengelyével párhuzamos tengely mentén, már csak Noo[2] cikke számolja ki. A θ szöggel jellemzett elfordulást pedig minden két cikk elhanyagolja.



4. ábra. A Cone-beam CT elrendezésének vázlata[2]. A z tengellyel párhuzamos forgástengely R távolságra van az x irányban lévő forrásponttól. A detektor és a forrás közötti legrövidebb távolság D . A forrás legközelebb eső pontja a detektornak az u_0, v_0 koordinátával jellemzett pixel.



5. ábra. A Cone-beam CT elrendezésében jellemző irányok[2]. e_ω a detektor síkjára merőleges irány, α és β a detektor síkjában fekvő, egymásra merőleges vektorok. A detektor dőlését a ϕ, θ, η szögek jellemzik.

Az elhanyagolásokra az ad lehetőséget, hogy az nem befolyásolja nagy mértékben a rekonstruált képeket, illetve a többi paraméter mérési pontosságát.

Összefoglalva tehát meghatározandó paraméter a detektor és a forrás távolsága (D), a forgástengely és a forrás távolsága (R), a detektor pixelei közül a forráshoz legközelebbi (u, v) , valamint a detektor elfordulása a saját síkjában (η). Ezen kívül Noo módszerével meghatározható a detektor ferdeségét jellemző ϕ szög is, de ez egyrést a kísérleti algoritmusban sem került meghatározásra, másrészt a rekonstrukciónak sem bemenete.

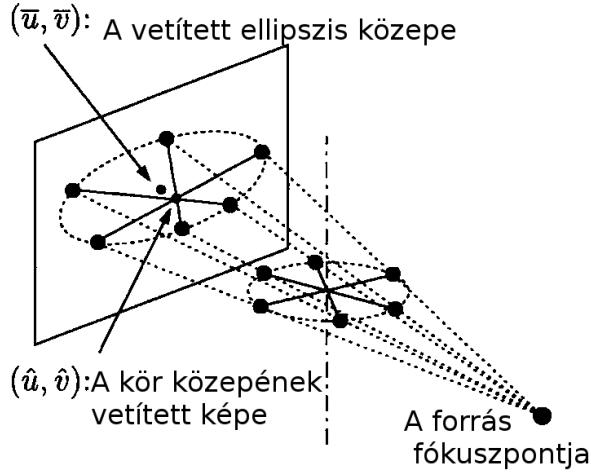
3.2. A geometriai paraméterek meghatározásának elmélete

A fent összegzett paraméterek az említett cikkek alapján meghatározhatóak úgy, hogy a forgatható asztalra egy plexi – vagy egyéb, alacsony sűrűségű – lapra fém – vagy egyéb, nagy sűrűségű – gömb alakú tárgyat helyezünk el, jól meghatározott távolságra, majd az asztalt körbeforgatva felvételeket készítünk. Belátható[2], hogy a forgatás során körbeforduló gömbök a detektoron ellipszis pályát írnak le. A geometriai paramétereket ezen ellipszis pályákból tudjuk meghatározni. Ennek módját először Noo cikke alapján mutatom be, majd ismertetem, hogy ahhoz képest Wu cikke milyen változásokat tartalmaz.

3.2.1. A paraméterek meghatározása Noo[2] cikke alapján

Az adatok feldolgozása minden esetben azzal kezdődik, hogy az egyes projekciókon meg kell keresni, hogy a detektoron hol jelenik meg a golyók képe. A sorrendben k . golyó helyzetét az i . projekción (u_k^i, v_k^i) -vel jelölöm. Noo cikkében két golyót tartalmazó fantomot használnak, így k értéke 1 vagy 2 lehet. i értéke 1 és N között változik, ahol N a teljes körbeforduláshoz alatt elkészített felvételek száma. Az egyes projekciók az említett koordináták megkeresése nem egyértelmű probléma, és az említett cikkek sem adnak részletes segítséget a feladat elvégzéséhez. Így a golyók középpontjának megkeresésére a megvalósítást tárgyaló a 3.3 fejezetben térek ki.

Miután a golyók középpontja az összes projekción rendelkezésre áll, elsőként a detektor saját síkjában való elfordulását kell korrigálni, azaz meghatározni az η szöget. Ehhez meg kell határozni a forgástengely vetített (\hat{u}_k, \hat{v}_k) képét a két ellipszis esetében. A 6. ábrán látható, ez a pont az épp ellentétes fázisban lévő pontok közé húzott szakaszok közös metszéspontjában van, amely nem feltétlenül esik egybe az ellipszis (\bar{u}_k, \bar{v}_k) középpontjával. A kereset (\hat{u}_k, \hat{v}_k) pont így a (8) egyenletrendszer alapján lineáris regresszióval kapható meg. A pontok ismeretében pedig a detektor elfordulása a (9) egyenlettel kapható.



6. ábra. A körbeforduló golyók vetített képe. A forgástengely vetített képe nem feltétlenül esik egybe a vetített ellipszis középpontjával.[2]

$$\left(u_k^j - u_k^i \right) \hat{v}_k - \left(v_k^j - v_k^i \right) \hat{u}_k = v_k^i u_k^j - v_k^j u_k^i, \quad j = i + \frac{N}{2}, \quad i = 1 \dots \frac{N}{2} \quad (8)$$

$$\eta = \arctan \left(\frac{\hat{u}_1 - \hat{u}_2}{\hat{v}_1 - \hat{v}_2} \right) \quad (9)$$

Az η szög meghatározása után az összes projekción a megtalált körök középpontját el kell forgatni ezzel a szöggel, és ez után a képek úgy kezelhetőek, mintha egy ferdeségtől mentes detektorral készültek volna. Vagyis egy ellipszis (u_k^i, v_k^i) pontja helyett a (10) egyenletekkel számolt (u_k^{i*}, v_k^{i*}) pontot vesszük figyelembe a számolás további részében. Csupán arra kell odafigyelni, hogy a detektoron keresett (u_0, v_0) koordinátákat végül vissza kell forgatni .

$$\begin{pmatrix} u_k^{i*} \\ v_k^{i*} \end{pmatrix} = \begin{pmatrix} \cos \eta & -\sin \eta \\ \sin \eta & \cos \eta \end{pmatrix} \cdot \begin{pmatrix} u_k^i \\ v_k^i \end{pmatrix} \quad (10)$$

Ez után az elforgatott pontokra ellipszist kell illeszteni, ahol az ellipszis pontjai a (11) egyenletet elégítik ki.

$$a \left(u^{i*} - \bar{u} \right)^2 + b \left(v^{i*} - \bar{v} \right)^2 + 2c \left(u^{i*} - \bar{u} \right) \left(v^{i*} - \bar{v} \right) = 1 \quad (11)$$

Az fenti egyenletből lineáris regresszióval meghatározhatóak az ellipszis (\bar{u}, \bar{v}) középpontjai, az a és b paraméterek, amelyek a tengelyek hosszával vannak kapcsolatban, valamint c , amely paraméter az ellipszis ferdeségére utal. Ezekből a paraméterekből először a forrás és a tengely távolsága (D) a meghatározható, a (12) egyenletek alapján. Az újonnan meghatározott D távolság ismeretében a további meghatározandó

paraméterek a (13) egyenletekkel számolhatóak. Az egyenletekben z_1 és z_2 a golyók z koordinátájára utal a fantomon. Amennyiben az adott golyó a fantomon a röntgenforrás vonala felett helyezkedik el, úgy előjele pozitív, amennyiben alatta, úgy előjele negatív. Ez az adat sem feltétlenül mérhető, így ezt is a képekből határozom meg. Továbbá a (12). egyenletből az elsőnél látható ϵ tag értéke ± 1 : pozitív előjellel, ha a két golyó a röntgenforrás két oldalán helyezkedik el, vagyis ha $z_1 z_2 < 0$. Amikor a golyók ugyan azon az oldalon vannak, vagyis $z_1 z_2 > 0$, akkor ϵ értéke lehet $+1$, valamint -1 is: ilyen esetben az előjel megállapításához szükséges a felhasználó döntése.

$$\left. \begin{aligned} D^2 &= \frac{(a_1 - 2n_0 n_1) - \epsilon \sqrt{a_1^2 + 4n_1^2 - 4n_0 n_1 a_1}}{2n_1^2} \\ n_0 &= \frac{1 - m_0^2 - m_1^2}{2m_0 m_1} \\ n_1 &= \frac{a_2 - a_1 m_1^2}{2m_0 m_1} \\ m_0 &= (\bar{v}_2 - \bar{v}_1) \sqrt{b_2 - \frac{c_2^2}{a_2}} \\ m_1 &= \sqrt{b_2 - \frac{c_2^2}{a_2}} \sqrt{b_1 - \frac{c_1^2}{a_1}} \end{aligned} \right\} \quad (12)$$

$$\left. \begin{aligned} v_0^* &= \bar{v}_1 - \text{sign}(z_1) \frac{\sqrt{a_1 + a_1^2 D^2}}{\sqrt{a_1 b_1 - c_1^2}} \\ u_0^* &= \frac{1}{2} \bar{u}_1 + \frac{1}{2} \bar{u}_2 + \frac{c_1}{2a_1} (\bar{v}_1 - v_0^*) + \frac{c_2}{2a_2} (\bar{v}_2 - v_0^*) \\ \rho_k &= \frac{\sqrt{a_k b_k - c_k^2}}{\sqrt{a_k b_k + a_k^2 b_k D^2 - c_k^2}} \quad k = 1, 2 \\ \zeta_k &= D \cdot \text{sign}(z_k) a_k \frac{\sqrt{a_k}}{\sqrt{a_k b_k + a_k^2 b_k D^2 - c_k^2}} \\ \sin \phi &= -\frac{c_1}{2a_1} \zeta_1 - \frac{c_2}{2a_2} \zeta_2 \end{aligned} \right\} \quad (13)$$

Végül már csak a forrás és forgástengely közötti R távolságot kell meghatároznunk. Ez a (14) egyenlet lapján történik, ahol d a két golyó távolsága.

$$\frac{d^2}{R^2} = \frac{1}{N} \sum_{i=1}^N \left\{ \left(\zeta_1 \frac{u_1^i - u_0^*}{v_1^i - v_0^*} - \zeta_2 \frac{u_2^i - u_0^*}{v_2^i - v_0^*} \right)^2 + \left(\frac{D \zeta_1}{v_1^i - v_0^*} - \frac{D \zeta_2}{v_2^i - v_0^*} \right)^2 + (\zeta_1 - \zeta_2)^2 \right\} \quad (14)$$

3.2.2. A paraméterek meghatározása Wu[3] cikke alapján

A fentiekhez képest a legnagyobb eltérést az jelenti Noo és Wu cikke között, hogy utóbbi esetben a fantomon nem két golyó van, hanem több, ezzel növelve a módszer

megbízhatóságát.

A paraméterek meghatározása ebben az esetben is az η szög meghatározásával és a vele való korrekcióval történik, ám ebben az esetben – mivel több golyó adata áll rendelkezésre – egyenes illesztéssel tudjuk meghatározni tan η értékét, a (15) egyenlet szerint.

$$\hat{u}_k = c + \tan \eta \cdot \hat{v}_k \quad (15)$$

A mérési pontokra ez után szintén ellipszist kell illeszteni, azonban Wu cikkében a (16) egyenlet írja le a ellipszist. el van hanyagolva tehát a (11) egyenletben még meglévő c paraméter, amely akkor lehető meg, ha az ellipszis nagytengelye éppen vízszintes állású. Ez jó közelítéssel igaz is, hiszen az ellipszisen el vannak forgatva az előzetesen meghatározott η szöggel, amely következtében a forgástengely képe a detektoron egy függőleges egyenes. Mivel az asztal körbeforgatása során a golyók merőlegesen mozognak a forgástengelyre, a vetített képükre is igaz lesz, hogy a pályájuk nagytengelye merőleges a forgástengely vetületére, azaz éppen vízszintes.

$$a \left(u^{i*} - \bar{u} \right)^2 + b \left(v^{i*} - \bar{v} \right)^2 = 1 \quad (16)$$

Ezzel az egyszerűsítéssel a keresett geometriai paramétereket leíró egyenletek is jócskán leegyszerűsödnek. A keresett v_0^* és D paraméterek a (17) egyenletrendszerből lineáris regresszióval megkaphatóak, amelyek a k és k' sorszámú ellipszisek között teremtenek összefüggést.

$$\frac{\bar{v}_k + \bar{v}_{k'}}{2} - \frac{1}{2(\bar{v}_k - \bar{v}_{k'})} \left(\frac{1}{b_k} - \frac{1}{b_{k'}} \right) = v_0^* + D^2 \frac{1}{2(\bar{v}_k - \bar{v}_{k'})} \left(\frac{a_k}{b_k} - \frac{a_{k'}}{b_{k'}} \right) \quad (17)$$

Vegyük észre, hogy Noo módszerével ellentétben itt nem szükséges a golyók függőleges pozíciójának (z_1, z_2) vizsgálata. Így ez a módszer minden esetben egyértelműen számolja a kérdéses paramétereket, míg Noo módszerénél az összes többi paraméter értékét meghatározó D paraméter bizonyos esetekben – amikor a kiválasztott golyók függőlegesen a forrás azonos oldalain voltak – nem volt egyértelmű. Továbbá ez a képlet jóval egyszerűbb, mint a (12) egyenletekben leírtak.

Ugyancsak egyszerű képletet kapunk u_0^* értékére: mivel a c paraméterek értékét elhanyagoljuk, a Noo által használt (13) képletek megfelelő sorára ránézve látszik, hogy u_0^* értéke az ellipszis illesztésnél kapott \bar{u}_k értékek átlagolásával kapható : 18, ahol E az alkalmazott golyók – és így az ellipszisek – száma.

$$u_0^* = \frac{1}{E} \sum_{k=1}^E \bar{u}_k \quad (18)$$

Ez után az egyetlen meghatározandó paraméter a forgástengely és a forrás R távolsága. Ezt a távolságot most a (19) egyenletekkel tudjuk meghatározni, amelyek egy k és egy k' sorszámú ellipszis között teremtenek kapcsolatot, és ahol $d^{kk'}$ a két ellipszishez tartozó golyók távolsága a fantomon.

$$\left(\frac{d^{kk'}}{R} \right)^2 = (\zeta_k - \zeta_{k'})^2 + (\rho_k)^2 + (\rho_{k'})^2 - 2\rho_k\rho_{k'} \cos(\alpha_k - \alpha_{k'}) \quad (19)$$

3.3. A golyók középpntjának meghatárzoása

Hivatkozások

- [1] M. Harris. Optimizing Parallel Reduction in CUDA. http://docs.nvidia.com/cuda/samples/6_Advanced/reduction/doc/reduction.pdf. [Online; utolsó hozzáférés: 2016.05.28. 14:00].
- [2] F. Noo, R. Clackdoyle, C. Mennessier, T. A. White, and T. J. Roney. Analytic method based on identification of ellipse parameters for scanner calibration in cone-beam tomography. *Physics in Medicine and Biology*, 45(11):3489, 2000.
- [3] D. Wu, L. Li, L. Zhang, Y. Xing, Z. Chen, and Y. Xiao. Geometric calibration of cone-beam ct with a flat-panel detector. In *Nuclear Science Symposium and Medical Imaging Conference (NSS/MIC), 2011 IEEE*, pages 2952–2955, Oct 2011.