



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	<input type="button" value="▼"/>
A diplomamunkát gondozó (a záróvizsgát szervező) tanszék:		Nukleáris Technika Tanszék <input type="button" value="▼"/>

A diplomamunka készítésének helye: Budapesti M^o szaki és Gazdaságtudományi Egyetem	
A témavezető neve: Dr. Légrády Dávid	A konzulens neve:
– munkahelye: BME NTI	(külső témavezető esetén kijelölt tanszéki munkatárs)
– beosztása: docens	– beosztása:
– e-mail címe: legrady@reak.bme.hu	– e-mail címe:

A diplomamunka címe:	Mikro-CT készülék rekonstrukciós és kalibrációs környezetének létrehozása	azonosítója: DM-2016-45
<p>A téma rövid leírása, a megoldandó legfontosabb feladatok felsorolása:</p> <p>A BME Nukleáris Technikai Intézetében rendelkezésre álló mikro-CT készülék kalibrációs és rekonstrukciós szoftverrendszere alkalmas tomográfiás felvételek készítésére és a képrekonstrukció elvégzésére. A vonatkozó algoritmusok érettségi foka azonban még alacsony szint^o, az eddig megvalósított eljárások a lehe^ti legegyszer^bb és olykor instabil eljárások. A hallgató feladata egységes, grafikus kártyát (GPU) is igénybe vevő kalibrációs és rekonstrukciós szoftver- és algoritmuskönyezet létrehozása. A hallgató feladatai:</p> <ul style="list-style-type: none"> -A geometriai kalibráció képfelismerési algoritmusának kidolgozása, robosztus algoritmus létrehozása és implementálása, melynél a geometriai kalibráció felhasználói beavatkozás nélkül, automatikusan elvégezhető. 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 miniségi elemzése, a készülék felbontásának vizsgálata. -A determinisztikus rekonstrukció felkeményedés-korrekciójának implementálása és összehasonlítása Monte Carlo alapú rekonstrukcióval. 		

A feladat kiadásának időpontja: 2016.02.19.
--

Témavezető vagy tanszéki konzulens aláírása:	A diplomamunka témakiírását 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 vettem, 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
Hivatkozások	14

1. Bevezetés

2. Gain korrekció

A gain korrekció az elkészült felvételeken elsődlegesen elvégzendő korrekció, amely felőlős 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.

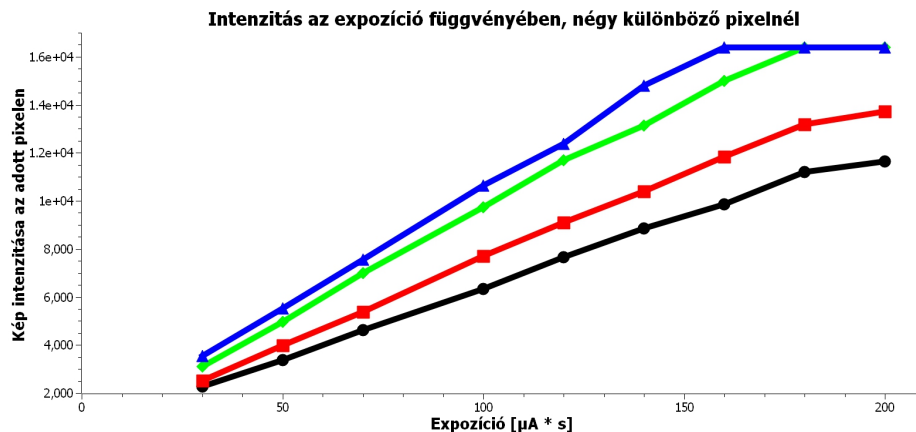
2.1. A gain korrekció által javított artefaktumok bemutatása

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 forrásponttól távolabb vannak, hiszen a mérhető intenzitás a fókuszponttól számított távolság négyzetének inverzével arányos. A jelenség következtében tehát a detektorpixelek intenzitása a szélek felé csökken, és a csökkenés mértéke az expozíciós beállításoktól – úgy mint az expozíciós idő, valamint a röntgenső feszültsége és árama – független. A kialakult képtorzulást *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, mígnem az intenzitás egy adott értéket elérve nem nő tovább. Utóbbi esetet, azaz, amikor az intenzitásérték tovább nem nő, szaturációnak hívjuk. Az előbbihez hasonlóan, ha állandó expozíciós idő mellett a röntgenső áramerősségét változtatjuk, és így nem a felvétel idejét növeljük, hanem a detektorra eső nyalábinszenzitást, szintén kezdetben lineáris, majd telítődő összefüggést kapunk a pixel által mért intenzitás – alkalmazott áramerősség 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.

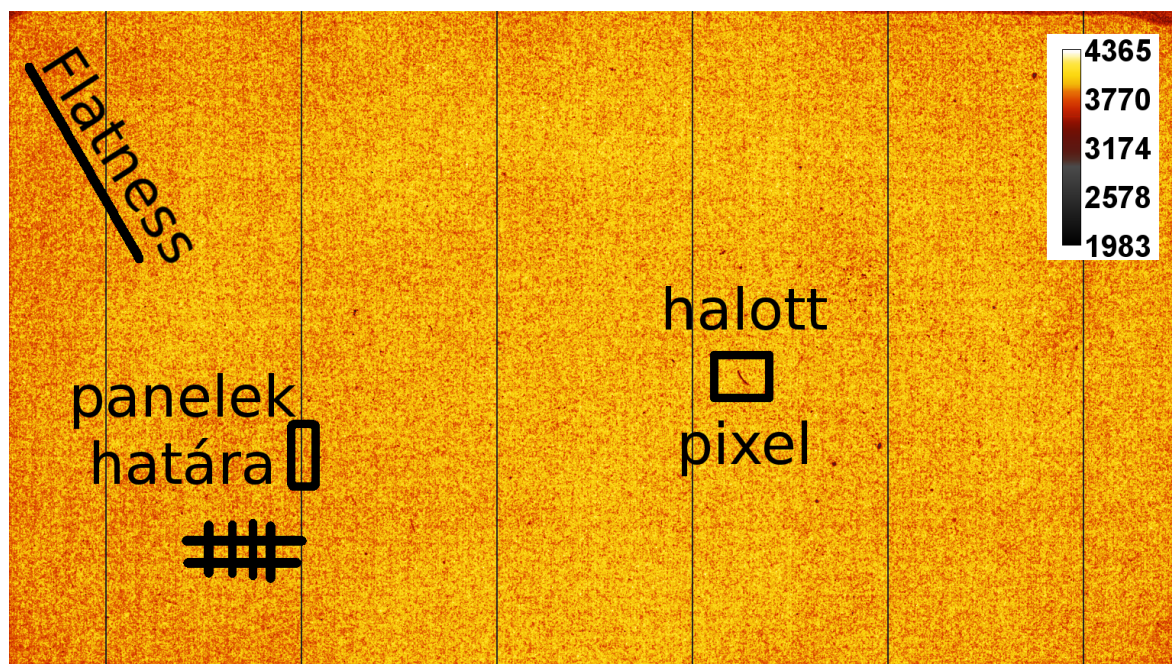
Ez az érzékenység azonban pixelről pixelre változik. Különböző pixelek érzékenységét megvizsgálva észrevehetjük, hogy az egyes érzékenységi görbék lineáris szakaszának meredeksége és tengelymetszete is változik. Változik továbbá a szaturációs expozíció is, azaz az az expozíció érték, amelynél az adott 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 jellemezhető, 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 tengelymetszetéből 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.

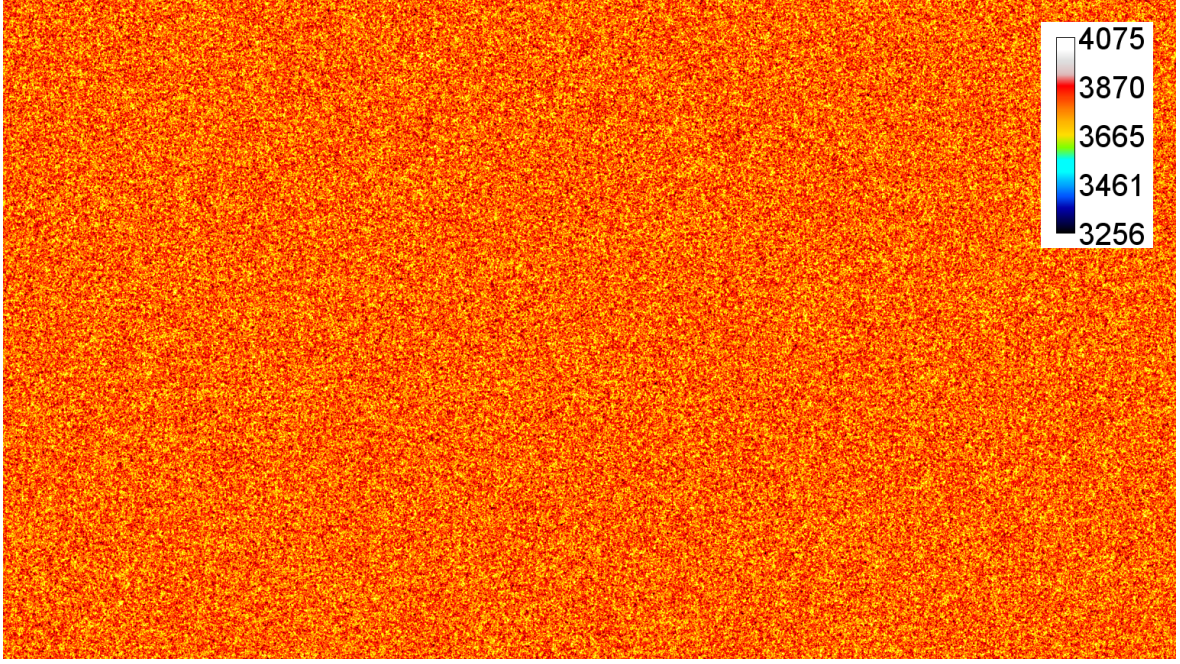


2. ábra. Tárgy nélkül készített felvétel, gain korrigálás nélkül.

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 mindenképp 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 mellett 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 $I_i^{\text{mért}}$ az i . pixel eredeti, korrekció nélküli intenzitása, $I_i^{\text{offset korrigált}}$ pedig az offset hibától mentes érték.

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

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 faktork 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 röntgenső 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 térni 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, mindnek 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 növelve 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 $I_i^{\text{gainkorrigált}}$ pixelintenzitás a (6) egyenlettel számolható, ahol $(2^{14} - 1)$ a detektor maximális intenzitásértéke.

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

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 együtthetők 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 megjegyzennem, 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 beállításonként). 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 expozíciós beállításokkal készültek, és ez alapján is szelektálom 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 röntgencső független kezelése részben biztonságvédelmi célokat 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 kiküszöbölendő, a gain korrekcióhoz készített kalibrációs képek feldolgozása során a képek átlagos intenzitásán kívül a program vizsgálja az átlagos áramerősség, feszültség 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 szűrőknek megfelel (5-nél 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 expozíciós beállítások száma a korrekciós faktorok meghatározásához – akár összesen, akár túl kevés megy át a fenti szűrőkön –, 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 robusztusságát 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.

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

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 \overline{xy} értékek pixelenként előállnak, mint egy–egy kép. Ez után a kettő különbségét – mint képet – leosztva a $\overline{x^2} - \bar{x}^2$ 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 bánt, 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 – utóbbit 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 -t: $\text{sum} = (a+b)+c$. 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 mellékelte 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 szál 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 szál fut, 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 elvégezhetőek. A felhasználónak ennél a funkciónál 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 röntgenső 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 röntgenső 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.

Hivatkozások

- [1] Mark Harris, *Optimizing Parallel Reduction in CUDA*, online:
http://docs.nvidia.com/cuda/samples/6_Advanced/reduction/doc/reduction.pdf,
Utolsó hozzáférés: 2016.05.28. 14:00