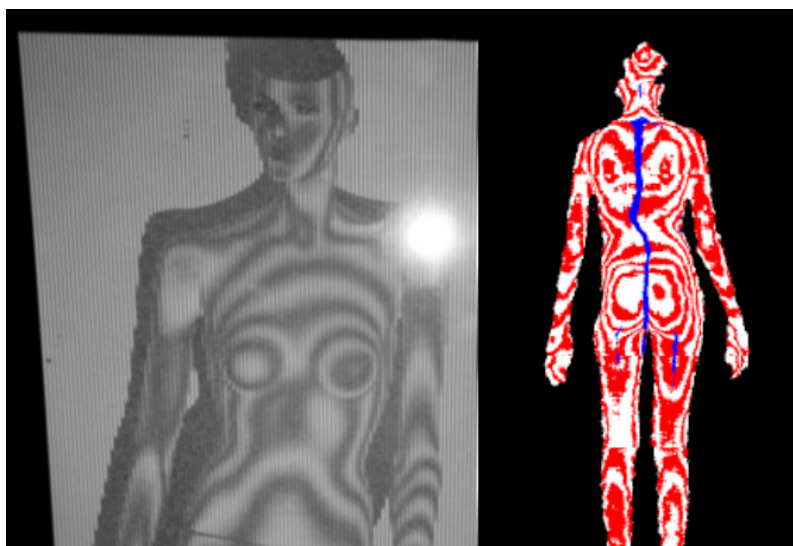




Ayhan Dániel II. évf. Mechatronika szak

Optikai gerincferdülés-diagnosztizáló szoftver fejlesztése



Konzulensek: Dr. Tamás Péter, egy. doc.
Molnár József, tansz. mérn.

Budapest, 2012. október

A dolgozat bemutatását a TÁMOP - 4.2.2.B-10/1--2010-0009 "Új tehetséggondozó programok és kutatások a Műegyetem tudományos műhelyeiben" című program támogatta.

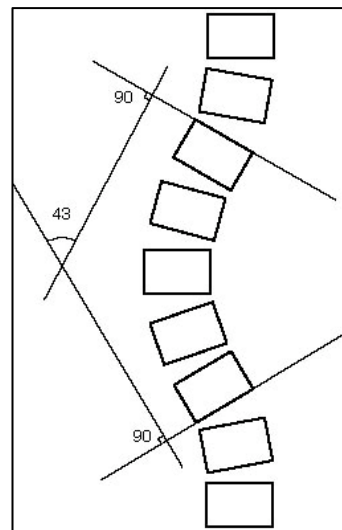
1. Bevezetés	3
2. A gerincferdülés (Scoliosis)	4
3. A gerincferdülés mérése	5
3.1. Lézeres szkennel	5
3.2. Fényminták vetítése.....	6
3.3. Moiré-csíkok	7
4. A mérőeszköz: Kinect.....	8
5. A Kinect felhasználása gerincferdülés mérésére	10
6. 3D-s megjelenítés	11
6.1. Háromszögháló, vertex-, indexpuffer	11
6.2. Koordinátarendszerek, transzformációs mátrixok és árnyalók	12
6.3. Hardver	12
6.4. XNA	13
7. A program működése	14
7.1. A bemenet feldolgozása	14
7.2. Szimmetriavizsgálat	16
7.3. Nyírás	17
7.4. Felhasználói felület, a szoftver kezelése, megjelenítés	19
7. Értékelés	21
8. Továbbifejlesztési tervek	22
9. Összefoglalás.....	23
10. Irodalom.....	24
11. Képgaléria	25

1. Bevezetés

Napjainkban az alternatív megoldások egyre nagyobb teret hódítanak, egyre több helyen váltják ki a hagyományos mérési módszereket új technikák. A gerincferdülés mérése is ilyen terület. Egyre több újszerű elképzelés van a gerincferdülés méréseire. Dolgozatomban szeretném bemutatni az eddig létező módszereket, majd ismertetni egy újat, valamint az ahhoz a módszerhez írt szoftver működését.

2. A gerincferdülés (Scoliosis)

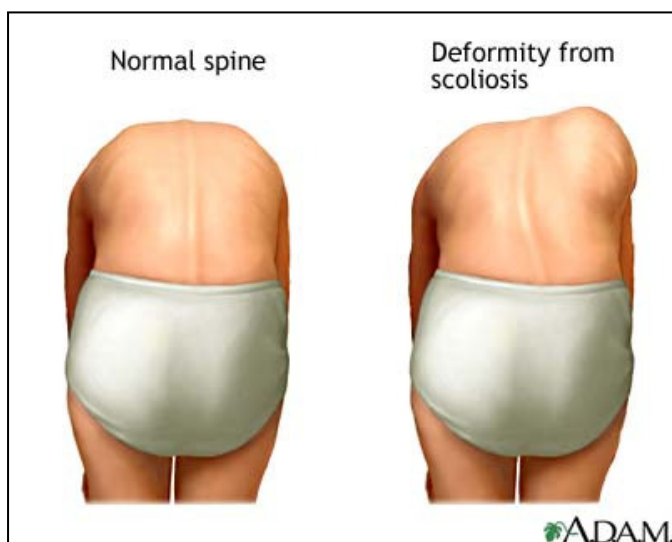
A gerincferdülés a gerinc oldalirányú ferdesége, az egyik leggyakoribb gerincbetegség. Az oldalirányú görbület mellett a gerinc torziója, a csigolyák egymáson való elfordulása is jelentkezik, így a hát egyik oldalán bordapúp keletkezik. A gerincferdülés kiváltó oka az esetek többségében ismeretlen, általában növekedési rendellenességnek tudható be. Legtöbbször már fiatal korban kialakul, lányok esetében nagyobb a kialakulás valószínűsége. A gerincferdülés mértékét Cobb-fokkal mérjük, amely a gerincgörbe frontális síkvetületének inflexiók pontjaiban állított érintőinek az egymással bezárt szöge (2.1. ábra). A gerincferdülés kezelése történhet:



2.1. ábra
Ferdült gerinc

- **Gyógytornával:** A gerincferdülés egyik kiváltó oka lehet az izomzat féloldalassága, többek között ezt, valamint a tüdő benyomódását (Schroth módszer) próbálják kompenzálni a tornával, légzésgyakorlatokkal.
- **Korzettel:** Többféle korzett (2.2. ábra) létezik, többféle korrekciós hatással. A megfelelő korzett kiválasztását a gerincferdülés típusának helyes diagnosztizálásával érhetjük el, bár esetenként mások a korzetre való reakciók.
- **Műtéttel:** 50°-nál nagyobb görbületű gerincferdülésnél jön szóba a műtét. A betegség súlyosságától függően távolítanak el porckorongokat a gerincből és rögzítik azt csavarok és sínek segítségével.

Források: [1], [2]



2.1. ábra

*A gerinc torziója miatt a lapockák magasságában
kiemelkedik a hát*



2.2. ábra
Korzett

3. A gerincferdülés mérése

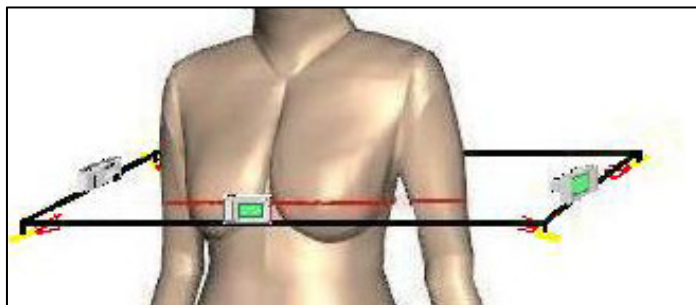
A gerincferdülés kezelése – mint a legtöbb betegségnél – annál könnyebb, minél hamarabb felismerik azt. Mivel általában gyermekkorban alakul ki, ezért a hirtelen növekedés gyors romláshoz vezethet. Bár enyhe görbületnél 10-30%-os javulást is el lehet érni, mégis a legtöbb esetben a betegség kezelésekor a fő cél az, hogy annak továbbhaladását, romlását megakadályozzuk, annál is inkább, mert – szintén, mint a legtöbb betegségnél – a gerincferdülés romlása annál gyorsabb, minél nagyobb a görbület mértéke. A betegség előrehaladtának megállapítása érdekében a betegeknek rendszeresen kontrollra kell járniuk, és minimum évente kétirányú röntgen is kell hogy készüljön a gerincükről a további kezelés mikéntjének megállapításának érdekében. Ezen kívül röntgenfelvételt készítenek a betegről korzettben is, megállapítandó, hogy a kívánt korrigáló hatást el sikerült-e érni.

Mint arra már följebb utaltam, a gerincferdülés mérésére egészségügyi intézményekben szinte kizárólag a röntgent használják. A röntgen elvitathatatlan előnye, hogy igen kicsi a mérési hiba lehetősége, mivel nincs szükség a mérés elvében különböző egyszerűsítésekre, modellalkotásra, hiszen a gerinc méretei láthatóvá válnak gyakorlatilag torzítás nélkül. Ebből kifolyólag a röntgenes vizsgálat nagyon pontos mérést tesz lehetővé. A módszer számos előnye mellett nagy hátránya, hogy nagy sugárdózist terhel a páciensre közvetlenül, közvetve pedig az orvosra is, ezen kívül a berendezés nagy méretei folytán költséges is.

Mivel a gerinc deformálódása szükségszerűen a hát alakjának változását vonja maga után, ezért logikus volna, hogy a gerincgörbe alakját a hát formájából visszakövetkeztetve keressük. Ehhez „csak” a hát geometriáját kell megmérni, nincs szükség röntgensugarakra, bár a berendezés nem lesz feltétlenül sem kicsi, sem olcsó, mint pl. egy jó lézeres szkennerek esetében. Az alábbiakban tekintsük át, hogy milyen módszerek állnak rendelkezésünkre a hát geometriájának mérésére.

3.1. Lézeres szkennerek

A lézeres szkennerek a geometriát metszetenként képesek letapogatni. Ezek a metszetek a könnyű feldolgozás és nagy pontosság érdekében legtöbbször vízszintesek. Az adott metszet síkjában egy lézervonalsugárzó működik, ez világítja meg a vizsgált tárgyat. A lézervonalsugárzó egy hagyományos pointerből és egy optikai elemből áll,



3.1. ábra

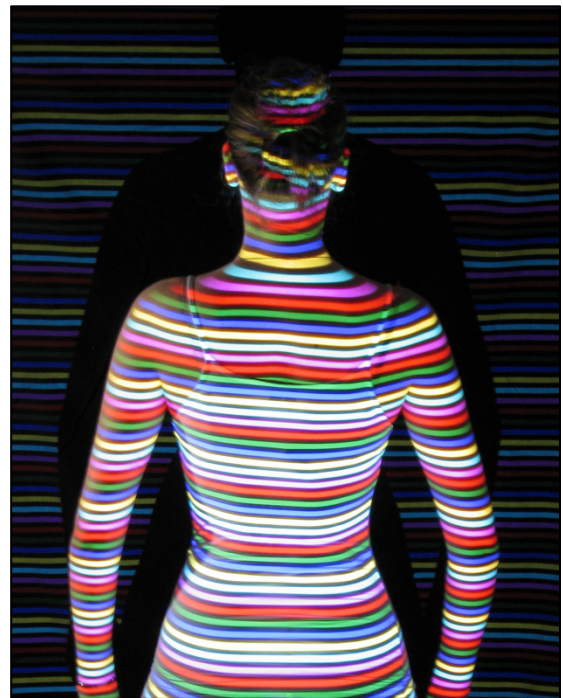
Lézeres szkennerek működése

amely optikai elem a pointer által kibocsátott lézernyalábot síkban „elnyújtja”. Így a vonalsugárzó egy meghatározott síkban, adott látószögben képes sugározni. A lézeres

szkenner esetén a lézer feladata csupán megvilágítás, így nincs szükség nagy teljesítményű lézerre, ami az emberi szemre is káros. A vonalsugárzó síkjához képest szöget bezáróan egy kamera helyezkedik el, ami egy monokromatikus szűrő segítségével csak a lézercsíkot látja. A lézercsík alakja a mérendő tárgy geometriájának függvényében változik, ebből számítható a tárgy pontjainak helyzete, ugyanis a kamera képén minden ponthoz (pixelhez) tartozik egy térrész, amelyből jövő fénysugarak színét a kamera az adott pontba (pixelre) rögzíti. A ma kereskedelemben kapható kamerák felbontása mellett ez a térrész közelíthető egy iránnyal, melynek alig van kiterjedése. Ennek az iránynak és a kamerának a síkhoz viszonyított helyzetének ismeretében az egyenes dőléspontja számolható, ez lesz a pont, amit a vonalsugárzó megvilágított, feltéve, hogy minden pixelhez tudjuk a térbeli irányt, amit képvisel, ami szintén számolható lenne a kamera látószögéből (és képarányából), de ezen a ponton érdemes az eszköz kalibrálására bízni a dolgot. A berendezés megvalósítása egy kerettel történik, melyre rá van rögzítve mind a lézersugárzó, mind a kamera, így azok relatív helyzete nem változik. A keret függőleges mozgásával változtatható a metszet síkja, a mozgást sínekkel biztosítjuk.

3.2. Fényminták vetítése

Fényminták vetítésénél hasonló az elv, mint a lézeres mérésnél, ám ekkor nem feltétlenül tudjuk a sík egyenletét minden vetített csíkhoz, Bár kiszámolható lenne a projektor által vetített képből, itt sokkal érdekesebb a kalibrálásra támaszkodni, hiszen egy jó mérés mindig pontosabb, mint egy hosszú számolás. Ebben az esetben nincs szükség szűrőre a kamera előtt, minthogy a kibocsájtott és ezáltal a visszavert fény is a látható tartományba esik. Pont ebből adódik a módszer nehézkessége, ugyanis a számítógépes feldolgozáskor nagy nehézséget tud okozni a mérendő tárgy színe, vagy ha nem egyszínű, hanem foltos, stb. Volt próbálkozás színes csíkok vetítésével is.



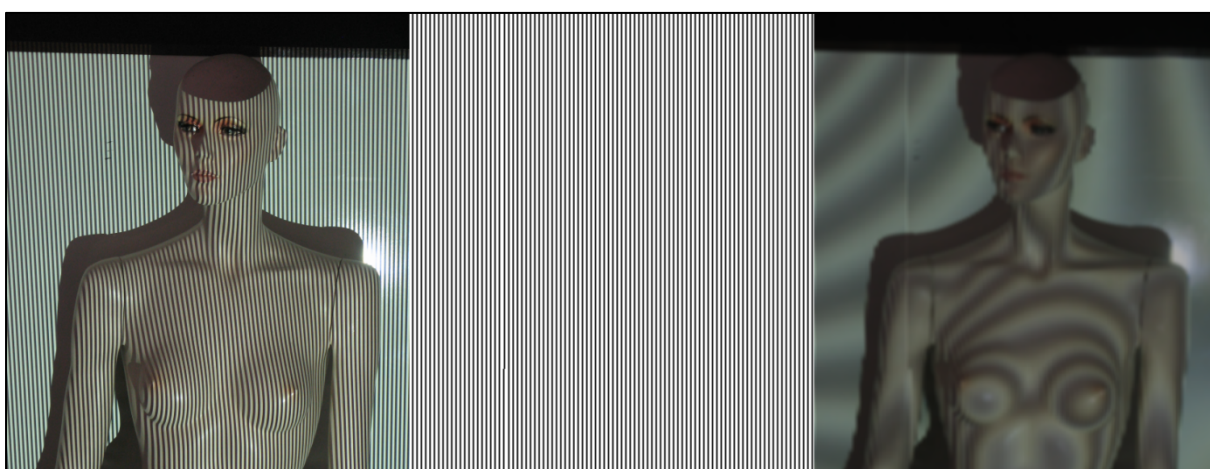
3.2. ábra

„Rainbow” vetítés

3.3. Moiré-csíkok

A Moiré jelenség akkor jön létre, amikor egy adott tárgyra egy rácson keresztül fényt vetítünk, majd a kialakult képet egy más rácssűrűségű rácson keresztül, vagy azonos rácssűrűségű rácson keresztül más szögből nézzük. Ekkor a két minta egymásra hatásából ahol a két rács fedésben van, világosabb, ahol pedig az egyik rács „rését” a másik rács pont kitakarja, sötétebb lesz a kép. Ebből megfelelő elrendezés esetén olyan csíkozat alakul ki, mint a térképek szintvonalai, ebből pedig következtetni lehet a geometriára. A fő probléma itt is a szoftveres feldolgozásban van, nehéz pontos mélység értékeket kinyerni az így kapott képből.

Források: [3], [4], [5], [6]



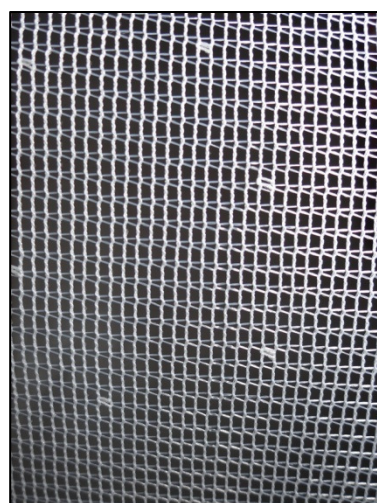
3.3. ábra

A Moiré effektus: Baloldalon az eredeti kép, középen a számítógépesen ráhelyezett rács, jobboldalon a keletkező kép a vonalak sűrűségével közel egyező Gauss-torzítással.



3.4. ábra

A szobám függönyén életközelié váló Moiré jelenség. A képen két réteg függönyön nézünk keresztül.



3.5. ábra

A függöny képe közlről: a rácsot a textil laza szövése adja.

4. A mérőeszköz: Kinect

A Kinect a Microsoft által 2010 novemberében piacra dobott mélységszenzor, amely a PrimeSense nevű cég technológiáját használja. A Kinectet eredetileg az Xbox 360 játékkonzolhoz tervezték bemeneti eszköz gyanánt. A termék nagyfokú sikerét illusztrálja, hogy a forgalomba kerülése utáni első fél évben 10 millió darabot adtak el belőle. Nem csak szórakoztatási célokra használható a Kinect, világszerte ezerféle feladathoz alkalmazzák. Népszerűsége az ár/érték arányával magyarázható, kis méretéhez és alacsony árához képest az általa szolgáltatott mélységadatok felveszik a versenyt például a time-of-flight módszerrel működő (tehát időmérésen alapuló) érzékelőkkel.

A Kinect mélységérzékeléshez egy infravörös lézerrel pseudorandom mintát vetít a környező térrészre. Ezt a mintát egy infravörös szűrővel ellátott kamera érzékeli. Az eszköz a vetített és a kapott kép eltéréséből számítja mélységadatot.

Az eszköz kimenete egy mélységtérkép, amely a hagyományos fényképezőgéppel készített képtől annyiban különbözik, hogy pixelenként nem színinformációt tartalmaz, hanem mélységinformációt, így a kimenetből lehet következtetni a geometriára.

A Kinect névleges paramétereit az alábbi táblázat tartalmazza:

4.1.1. táblázat: A Kinect névleges paramétereit

Látószög	vízszintesen 57°, függőlegesen 43°
Mintavételezési frekvencia	kb. 30 Hz
Mélységkép felbontása	640 x 480
Névleges méréshatár	0,8 m – 3,5 m
Névleges mérési pontosság (2m távolságban)	1 cm



4.1. ábra

Kinect

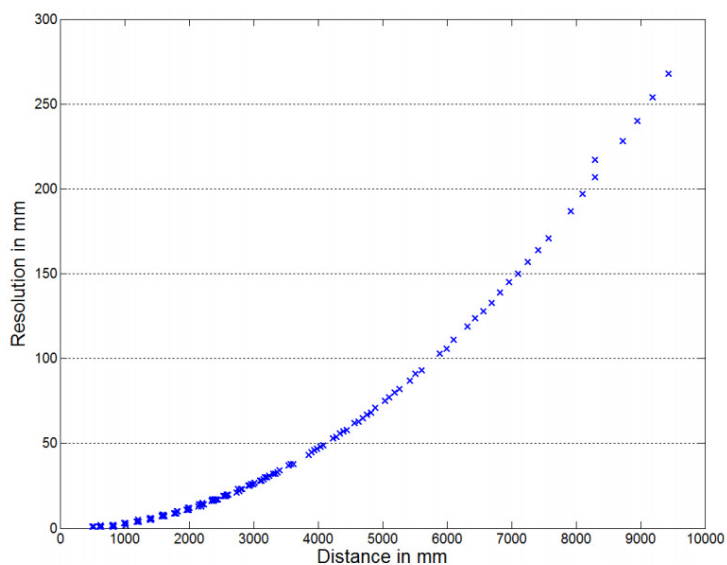


4.2. ábra

A Kinect által vetített minta részlete külső infrakamerával rögzítve.

A tényleges felbontás a 4.3. ábra alapján változik, így ha közelebb állunk az eszközhöz, nagyobb pontosságot is el tudunk érni. A névleges méréshatár sem mérvadó, bár a kívánt pontosságot csak itt lehet elérni, a méréshatár tágítható 0,5 m – 9 m-re, ha nem a Microsoft által biztosított szoftverfejlesztői eszköztárat használjuk, habár a felső véglet esetén a mérési pontosság akár 25 cm-re is kitolódhat, valószínűleg ez az oka a Microsoft SDK-ban a méréshatár korlátozásának.

Források: [7]



4.3. ábra

A Kinect felbontása a távolság függvényében

5. A Kinect felhasználása gerincferdülés mérésére

Mint azt említettem, a Kinectet rengeteg olyan dologra használják világszerte, amihez 3D-érzékelés szükséges. A kutatásunk célja az, hogy megállapítsuk, vajon a gerincferdülés diagnosztizálására is használható-e. Az alapfelgondolás tehát, hogy a Kinect segítségével le tudjuk mérni a páciens hátának geometriáját, ebből a geometriából pedig valamilyen módon meghatározni a gerincgörbe alakját, ami gyakorlatilag a röntgenes vizsgálatot válthatná ki. A Kinect által küldött adatok feldolgozása számítógéppel történik. A feldolgozó szoftver működését tárgyalja az alábbi két fejezet.

A módszer újdonsága abban áll, hogy használható adatokat szolgáltató 3D-s leképezésre eddig szinte csak a lézeres szkennerek voltak képesek, amelyek piaci áruk miatt nem voltak széles körben hozzáférhetőek. A Kinect ezzel szemben egy sokkal alacsonyabb árkategóriát képvisel, így lehetőség nyílik a hát geometriája és a gerinc deformitása közötti kapcsolat vizsgálatára.

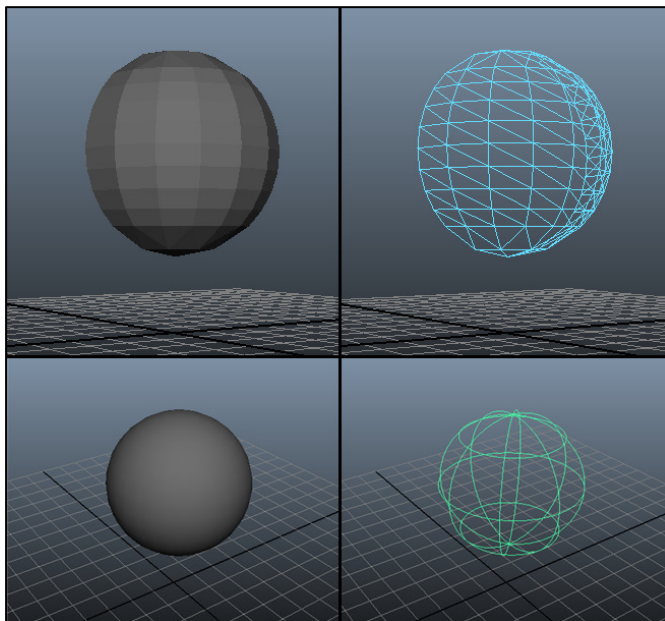
Az alapfeltevés az, hogy a gerinc környékén az ember háta szimmetrikus. A betegség kialakulása során a feltevés szerint ez a szimmetria lokálisan megmarad, „csupán” a globális szimmetria torzul a gerinc ferdülése mentén. Erre épít a 3D-s adatokat feldolgozó program, a szimmetriát keresve próbálja megtalálni a gerinc vonalát.

6. 3D-s megjelenítés

A programban többek között a kapott kép egyszerű 3D-s megjelenítése történik, így röviden ismertetném a térbeli számítógépes megjelenítés működését.

6.1. Háromszögháló, vertex-, indexpuffer

A számítógépes háromdimenziós térbeli megjelenítés legelterjedtebb, és jelenleg leggyorsabb módja a háromszöghálóra való megjelenítés. E módszerrel olyan térbeli felületeket lehet megjeleníteni, amelyek kizárólag háromszögekből állnak. Mivel a legtöbb modellezőprogram (SolidWorks, Inventor, Maya stb.) parametrikus felületekkel modellezi le a létrehozott 3D-s objektumot, ezért ezeket megjelenítés előtt tesszelálni kell, vagyis háromszögesíteni, melynek során az eredeti alakját elveszti a test, de megjeleníthető állapotba jut. Az így kapott pontokat, melyek a háromszögek csúcspontjai, vertexeknek nevezzük. A vertexek nem csupán geometriai információt tartalmazhatnak, tartozhat hozzájuk texel érték és/vagy



6.1. ábra

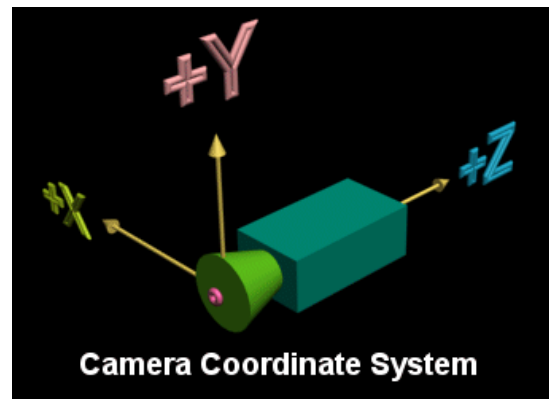
Felül: Egy háromszöghálóra leírt félgömb árnyalt és drótvázis képe. Alul egy parametrikusan leírt gömb két hasonló képe. A felső félgömb az alsó gömb tesszelálásával készült.

normálvektor is. A texel a háromszöghálóra ráfeszített kép (textúra) egy pontjának koordinátája, amely a textúra bal felső sarkához rögzített, vízszintesen jobbra mutató x tengely és függőlegesen lefele mutató y tengely által alkotott koordináarendszerben van megadva úgy, hogy a textúra jobb alsó sarka az (1,1) koordinátát kapja, így a texelben megadott értékek mindig 0 és 1 közé esnek. A vertexeket az úgynevezett vertexpufferben tároljuk, ami a belőlük épített tömb a memóriában. Mivel a legtöbb megjeleníteni kívánt objektum térben folytonos, ezért egy vertexhez több háromszög is tartozik (a 6.1. ábrán például a félgömbben egy-egy vertex 6 háromszögben is csúcspont). Éppen ezért a háromszögek megadására az indexpuffert használjuk. Az indexpuffer sorban az összes háromszög csúcspontjainak vertexpuffer-beli sorszámát adja meg, vagyis az indexpufferben hármassával vannak a megjelenítendő háromszögek csúcspontjainak vertexpuffer-beli sorszáma. Ezen felül minden háromszögnek van normálvektora is, a vertexekben tárolt normálvektortól függetlenül. Ennek a normálvektornak az irányától függ, hogy látjuk-e a háromszöget, ugyanis a normálvektor irányából nézve a háromszög megjelenítésre kerül, ellenkező

esetben nem. A normálvektor is az indexpuffer segítségével van megadva úgy, hogy a normálvektor irányából nézve a háromszöget, az indexpufferben beállítástól függően óramutató járásával megegyező vagy ellentétes irányban vannak feltüntetve a vertexek sorszámai.

6.2. Koordinátarendszerek, transzformációs mátrixok és árnyalók

A vertexek koordinátái modell-koordinátarendszerben vannak megadva, ami egy, a modellre jellemző koordinátarendszer. Mivel egyszerre legtöbbször több modellt szeretnénk megjeleníteni, amelyek ráadásul még egymáshoz képest mozoghatnak is, ezt transzformálni kell világ-koordinátarendszerbe, amely a megjelenítendő térre jellemző. Ezt egy modellenként különböző mátrixszal való beszorzással érjük el. Ez a mátrix 4x4-es, ugyanis tartalmazza az eltolási koordinátákat is. A koordinátákat azért tároljuk modell-koordinátarendszerben, mert így a modellek mozgását könnyebb létrehozni, nem kell az összes vertexet újraszámolni, csupán a transzformációs mátrixot. Ezeket a világ-koordinátarendszerbeli koordinátákat át kell számolni kamera-koordinátákba, amely egy olyan koordinátarendszer, amelynek általában adott egy nézési iránya és egy felfelé mutató irány. A harmadik irány ezek keresztszorzatából számolható (6.2. ábra). Itt is a fent leírtakhoz hasonló megfontolásból van szükség egy új koordinátarendszerre, a kamera is mozoghat. Ezek számítása után a vertexeket egy utolsó mátrixszal a képernyő síkjába kell transzformálni a projekciós mátrixszal, így 2D-s képet kapunk az objektumból, bár minden így kapott vertexhez tartozik egy z koordináta is, ami a mélységet adja meg. Ezen z koordináták alapján történik az egyes háromszögek kirajzolása, és végül a legfőbb háromszögek kerülnek megjelenítésre.



6.2. ábra

Kamera-koordinátarendszer

Ezeket a transzformációkat az úgynevezett vertexárnyaló végzi, míg a pixelárnyaló általában a megjelenített objektum különböző pontjainak színével kapcsolatos számításokat végez (pl árnyékszámítás, Lambert-féle, Phong-féle reflektációs modell alapján a színek számítása). Az árnyalók alatt az őket megvalósító algoritmust értjük, amely az újabb hardvereken egyre inkább programozható és nem csak a hardverhez készült szoftverfejlesztői eszköztárban (SDK, Software Development Kit) található függvényekből rakható össze, hanem külön árnyaló nyelven (HLSL, GLSL) írt program kerül fordításra, amit aztán a grafikus kártyán fut.

6.3. Hardver

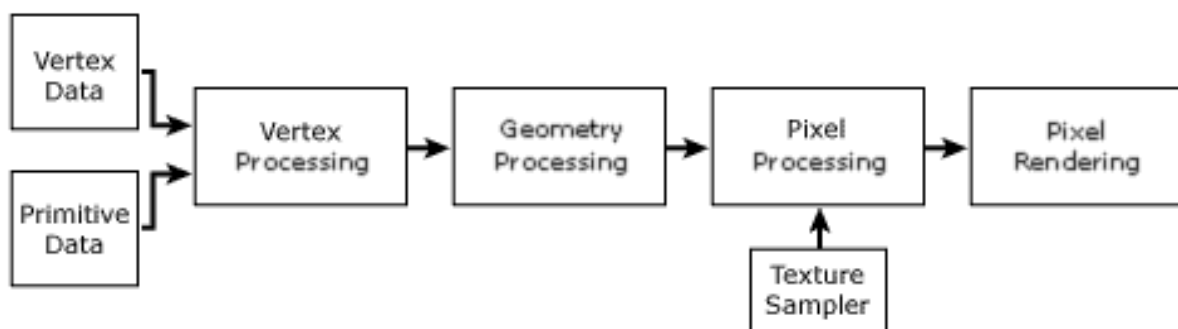
A mai PC-k háromdimenziós megjelenítési funkcióit a grafikus kártya végzi. A grafikus kártyán található a GPU (Graphics Processing Unit), ez felel a szükséges számítások elvégzéséért. Míg a CPU-ban csak pár processzormag található, a GPU több ezer maggal számol egyszerre,

amelyek sokkal kisebb teljesítményűek, használatukban kevésbé rugalmasak, csak egyszerűbb utasításokat tudnak elvégezni, ám pont ezért hatékonyabbak, hiszen célirányosabban tervezhetők. A GPU magjai a processzoréval szemben sokkal egyszerűbb számításokat végeznek, ezek végzik a vertexárnyaló és a pixelárnyaló számolását. Mivel egy megjelenítendő objektum általában sok vertexet tartalmaz, ez lehetőséget ad a feladat egyszerű és nagyfokú párhuzamosítására, ugyanis ekkor egy-egy mag foglalkozhat egy-egy vertexszel, innen a GPU magjainak nagy száma. Ez egy CPU esetén nem feltétlen megoldható az elvégzendő feladatok sokfélesége miatt. Ennek a párhuzamosításnak az egyik legnagyobb hátránya, hogy ha esetleg egy magnak többet kell számolnia, mint a többinek (pl. bonyolultabb, számításigényesebb árnyalási modellt számol egy pontban), az egyszerűbb számítások eredményei nem hagyják el előbb a GPU-t mint a legbonyolultabbaké, hogy a párhuzamosítás szinkronban maradjon.

6.4. XNA

A program a Microsoft XNA nevű keretrendszert használja megjelenítéshez. Ez egy viszonylag új, a számítógépes grafikát igencsak „programozóbaráttá” tévő keretrendszer, amely támogatja a HLSL nyelven írt árnyalók futtatását és számos, a 3D grafikus programok fejlesztését nagymértékben megkönnyítő függvénykönyvtárat tartalmaz, ami nagy súlyt vesz le a szoftverfejlesztő válláról. Az XNA-t jelenleg csak C# nyelven adta ki a Microsoft. Egyébként a keretrendszer kifejezetten számítógépes játékok programozására készült, természetesen ettől függetlenül általános grafikus szoftverfejlesztésre is használható.

Források: [8]



6.3. ábra

A DirectX grafikus csővezetéke (Graphics Pipeline). A vertex- és indexpuffer adatai ezen mennek keresztül feldolgozás során. A vertex- és pixelárnyalók a Vertex Processing és Pixel Processing részben futnak. A „csővezeték” elnevezést onnan kapta, hogy az egyik algoritmus bemenete a másik kimenete, így mint egy csövön megy át rajta az adat.

7. A program működése

A program C# nyelven van írva, a Microsoft XNA keretrendszert használja, és egy hagyományos Windows ablakos alkalmazás keretein belül fut.

Kötelességem megemlíteni, hogy a program grafikus megjelenítési alapfunkcióinak megírását nem én végeztem el. Az alapprogramot az „*Objektum-orientált technika C++ és C# nyelven*” című, BMEGERIA4C3 Neptun-kódú tantárgy keretein belül kaptam, és Major Péter, az órákat tartó hallgató munkája. Forrás: [9]

A program blokkdiagramja a 11.1. ábrán látható, a szövegben a diagramon jelölt részfolyamatokra azok sorszámaival így hivatkozom: (11.1./##)

7.1. A bemenet feldolgozása

A szoftver bemenete egy szövegfájl, amely a mélység adatokat tartalmazza sor-oszlop szerint sorrendben, 480*640 db számot. Lehetőség van korábban megnyitott és feldolgozott fájlok megnyitására is, ezek *.dat formátumban vannak, és a felépítésük kicsiben különbözik csak a feldolgozatlan fájloktól. A mélységtérképet egy osztályban tároljuk, ez az osztály végzi később az adatokon a különböző műveleteket. A megnyitott fájlból a program visszaépíti a mélységtérképet, és tömbben tárolja (11.1./1). A koordináta-rendszert a 7.1. ábra alapján vesszük föl (bár nem egyértelmű, a képen látható hölgy – minthogy a gerincferdülését vizsgáltuk – nekünk háttal áll). Ha a fájl nem előfeldolgozott fájl, akkor a beolvasott adatok nem kívánatos információt is tartalmaznak (pl. a páciens mögötti térrész képét, valamint azokat a pontokat, ahol nem talált mélységadatot a Kinect, mert nem érzékelte az ott lévő tárgyat annak távolsága miatt), ezért szűrni kell azokat. Először a szoftver megkeresi a minimális mélység adatot (ez a távolság miatt nem érzékelt adat), az annál nagyobb legkisebb mélységet (igazi minimum), valamint a legnagyobbat (maximum). Ezek után a nem érzékelt adat mélységét 0-ra állítja (a 0 a mélységtérképben a NaN (Not a Number, nem szám) érték szerepét tölti be (11.1./2), majd lineárisan interpolálja az adatokat úgy, hogy a minimum és a maximum egy meghatározott értéket vegyen föl a későbbi könnyebb kezelhetőség miatt (11.1./3). Ezután egy szűrést végez, ahol a kis területű „szigeteket”, vagyis a többi felülettől elkülönülő kis foltokat eltávolítja (11.1./4). Ezt úgy teszi, hogy veszi az adott pont körüli négyzet alapú egyenes hasábot, megadott oldalhosszal és magassággal. A hasáb alapjának az oldalai a z és x tengelyekkel, az alkotók pedig az y tengellyel párhuzamosak. A program megszámlolja, hogy a

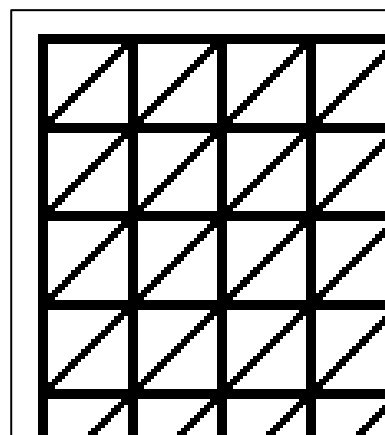


7.1. ábra

A használt koordináta-rendszer

hasábbba hány pont esik bele. Ha ez egy adott minimum alatt van, akkor az adott pontot mérési hibának minősíti, és a nem mért adat értékével, azaz nullával helyettesíti. Ez az algoritmus egyébként nem túl hatékony, de működik, és mélységtérképenként egyszer kell végrehajtani (hiszen utána a feldolgozott adathalmazt menteni lehet), úgyhogy a hatékonyabbra való cserélése jelenleg nem sürgős. Mindezek után egy meghatározott érték mögötti mélység adatokat szintén 0-ra állítja („hátsó vágósíkot” hoz létre), ezzel eltüntethető a páciens mögötti térrész képe (11.1./5).

Mivel a program képes 3D-s megjelenítésre is, a feldolgozás következő lépésében vertex- és indexpuffert épít (11.1./6) oly módon, hogy a tömb koordinátarendszeréhez (x koordináta az oszlop száma, y a sor száma) hozzáveszi a tárolt mélység adatot, mint z koordinátát (ez a koordinátarendszer a 7.1. ábrán bemutatottól eltér, és csak a vertexek használják, mint modell-koordinátarendszert). Ezek után a háromszögeket úgy építi föl, hogy ha a tömbben tárolt adatokat mint egy négyzetháló csúcspontjaiban értelmezett függvényt tekintjük, a négyzetháló élei mentén szomszédos pontok, valamint a négyzetháló négyzeteibe berajzolt, a bal alsó és jobb felső sarkot összekötő átló alkotnak háromszög-éleket (7.2. ábra). Ezek után a háromszögek közül eltávolítja azokat, amelyek túl nagy területűek (11.1./7) pl. abból kifolyólag, hogy a három vertex nem ugyanahhoz a térbeli tárgyhoz tartozik (ez az objektumok szélén jelentkezik), vagy kiugró mérési hibák esetén fordul elő (ha azokat valami miatt nem vágta volna le egy előző algoritmus). Ezek után eltávolítja azokat a háromszögeket is (11.1./7), amelyekben található olyan vertex, aminek a mélység adata 0.



7.2. ábra

A tömb elemeinek élekkel való összekötése az indexpufferben

Ezzel olyan adathalmazhoz jutottunk, ahol 0-tól különböző értéket (szinte) csak a páciens testének pontjaiban találunk. Jelenleg ezeket a feldolgozáshoz használt értékeket (pl. hátsó vágósík mélysége) kézzel, kódból kell beállítani, viszont utána menthetők az adatok.

Ezekből a program előfeldolgozott fájl esetén csak a 3D-s megjelenítéshez szükséges műveleteket végzi el.

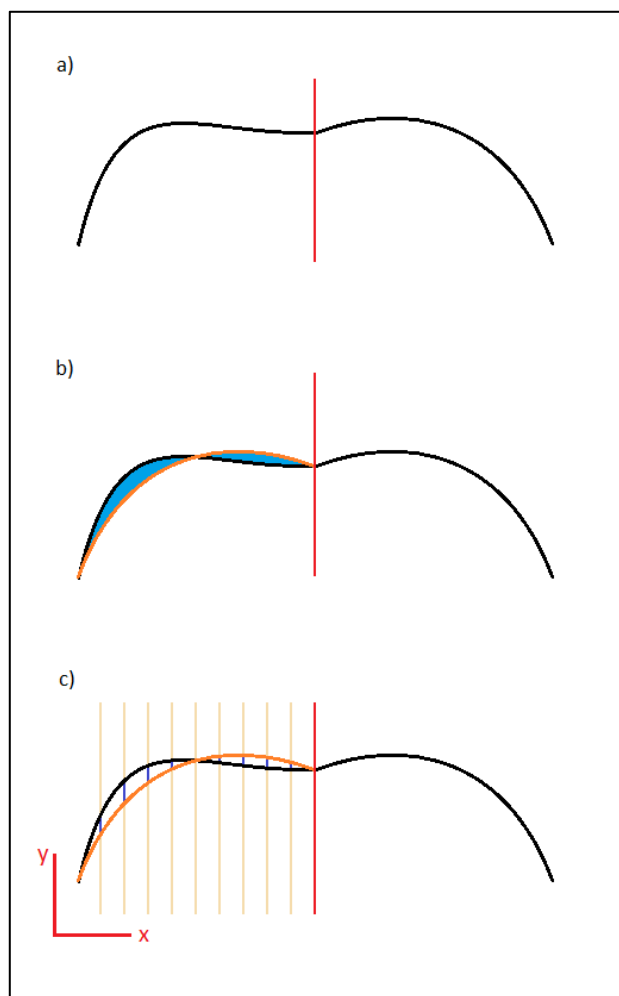
7.2. Szimmetriavizsgálat

Miután a program elvégezte az adatok beolvasását, nekiállhatunk vizsgálni a szimmetriát. A szimmetria keresésének általános elve a programban az, hogy 2D-ben valami (görbe, síkidom, pontthalmaz, stb) akkor szimmetrikus egy egyenesre, mint tengelyre, ha tükrözve az egyik felét az egyenesre visszakapom a másik felét. Attól függően, hogy az egyik fele mennyire különböző alakú, mint a másik, tekinthető valami kevésbé vagy jobban szimmetrikusnak. Ezt a kvalitatív megállapítást a következőképpen számszerűsítettem: Legyen adott a 7.3./a) ábrán látható görbe, amelynek szimmetrikusságát szeretnénk vizsgálni a jelölt tengelyre. Tükrözzük az egyik felét a tengelyre, így kapjuk a b) ábrán látható narancssárga görbét. Ekkor a kék terület jól jellemzi az eredeti görbe aszimmetrikusságát, hiszen ha az szimmetrikus volt, akkor ez a terület nulla,

ha pedig nagyon aszimmetrikus, vagyis a két fél nagyon eltér egymástól, akkor nagy ez a terület. Mivel azonban nekünk

pontsorunk van, ezért a c) ábrán látható módon y -koordinátabeli különbségeket kapunk az eltérés vizsgálatakor. Ezek négyzetének összegét kiszámoljuk bizonyos határokon belül (pl. a vizsgált tengelytől ± 50 pontnyira), átlagoljuk a mérhető pontok között (pl. nem mérhető az a pontpáros, amelynek egyik tagja 0 mélységgel szerepel a mélységtérképben), valamint tetszés szerint a tengelytől távoli pontokat kisebb vagy azonos súllyal számolhatjuk bele az összegbe, ha a szimmetrikusságot inkább lokálisan vizsgálnánk. Így kapunk egy mennyiséget, amely jellemzi az adott görbének egy tengelyét (a)szimmetrikusság szempontjából. Átlagolni azért kell az összeget, mert nagyobb x -kiterjedésű görbe és kb. ugyanakkora szimmetrikusság esetén nagyobb eltérést kapunk ebben az összegben, tehát egy ugyanilyen alakú, kétszer nagyobb görbénél átlagolás nélkül kétszer nagyobb értéket kapunk, pedig ugyanannyira tekinthető szimmetrikusnak, mint a feleakkora párja.

Mivel a fenti módszer csak két dimenzióban működik, ezért a hátnak csak metszeteire alkalmazható. Jelenleg a program a modell vízszintes metszeteit képes vizsgálni. Ezekben a metszetekben a fenti módszerrel meghatározza az aszimmetrikusságot minden pontra úgy,



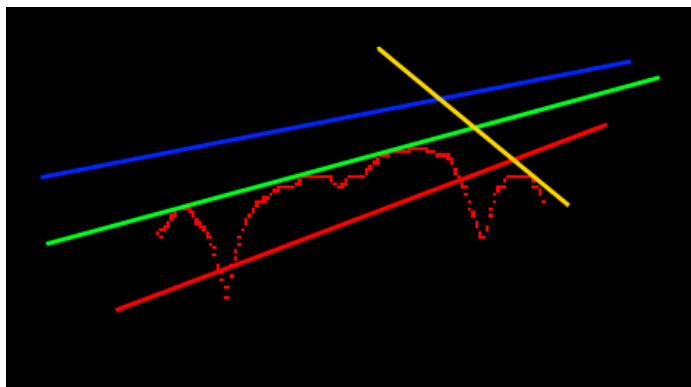
7.3. ábra

A szimmetriakeresés alapötlete

hogy veszi a ponton átmenő, y tengellyel párhuzamos egyenest és arra számolja a leírt módon az aszimmetrikusságot (11.1./11). Így tud egy bizonyos határ alatti aszimmetrikusságú pontokat megjeleníteni (11.1./12), mint „szimmetrikus pontokat”. Lehetőség van az alacsony „megbízhatóságú” mérési adatok eltüntetésére (megbízhatóságnak nevezem azt, hogy az aszimmetrikusságot hány pontpárból lehet meghatározni, pl. a modell szélén, a kontúrok mellett akadhat alacsony aszimmetrikusságú, ellenben egy-két pontpárból számolt adat, ami nyilván irreleváns).

7.3. Nyírás

Mivel a Kinectre pontosan merőlegesen állni (vagyis úgy, hogy a Kinect nézési iránya beleilleszkedjen a páciens szimmetriasíkjába, tehát pontosan „hátról” nézzen) szinte lehetetlen, ezért a modellek is ferdek voltak. Ebből kifolyólag a szimmetria-kereső algoritmus nem a hát közepén végighúzódo szimmetriát találta meg, hanem a ferdeség miatt a Kinecthez közelebb eső lapocka felületét, illetve egyéb, a gerinctől eltérő részeit a modellnek, ezért valahogy be kellett forgatni a modellt a kép síkjába. Mivel egy forgató algoritmust bonyolultabb megírni, és lassabb, valamint pontatlanabb is, mivel újra kell utána mintavételezni, ezért nyírást végeztem inkább az adatokon, ez kis szögnél egyébként sem torzít jelentősen a geometrián. Ehhez meg kell határozni egy nyírási szöget. Ezt érintők segítségével kerestem meg, annak az érintőnek az x tengellyel bezárt szögével kell nyírni az adatokat, amely a két lapockát köti össze, minthogy ez ideális esetben közel egyenlő a modell elfordulási szögével. Minden keresztmetszethez, mint pontsorhoz keresünk érintőt. Egy egyenes akkor tekinthető megfelelő érintőnek, ha minden pont „alatta van”, vagyis minden pont kisebb vagy egyenlő y koordinátával rendelkezik, mint az egyenes az adott x koordinátánál, továbbá legalább két pontot tartalmaz a pontsorból (7.4. ábra).



7.4. ábra

Jó és rossz érintők: A piros egyenes az első, a kék és a sárga egyenes a második feltételnek nem tesz eleget, míg a zöld egyenes mindkettőt kielégíti

Természetesen ilyen érintő többféle lehet. Az algoritmus, amellyel a megfelelőt keresem, a lokális maximumokból indul ki. Először megkeresi a maximumokat, vagyis azokat a pontokat, amelyek egy környezetükben a legnagyobb y értékkel rendelkeznek. Ezek közül kiválasztja a két legszélsőt, ami legtöbbször vagy a lapockák pontjai, vagy a páciens karjainak tetőpontjai lesznek, szerencsétlenebb esetben egy bennmaradt mérési hiba. Ezeket a program mint kiindulópontokat tekinti. Az algoritmus rekurzív módon fut. Minden iterációban van egy baloldali és egy jobboldali pont. Ezt a két pontot összeköti egy egyenessel, majd megnézi, hogy az egyenes fölött melyik az a pont, amelyik a leginkább az egyenes fölött van, vagyis

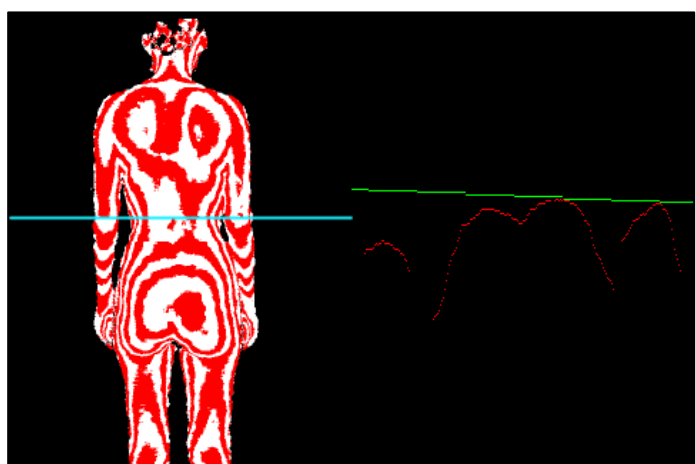
ami a legnagyobb y eltéréssel rendelkezik. Ezt a pontot a bal illetve jobboldali pontok közül azzal helyettesíti, amelyikhez közelebb van az x koordinátát tekintve. A következő iteráció alapjául ez a két pont szolgál, az új, valamint a régi, amelyik megmaradt az előző lépésből. Ezt az iterációt addig végzi a program, amíg meg nem találja az érintőt. Az első két kiindulópont a két legszélső lokális maximum. Az algoritmus helyességének bizonyítását nem végeztem el, mivel azoknál a metszeteknél, ahol szükség volt rá, 1-3 lépés alatt megtalálta az érintőt. A legtöbb esetben az iterációk száma 6 alatt volt. Egyébként, ha a rekurzió végtelen ciklusba futna, 10 iterációs lépés után megáll, de erre kevés példát találni. A kérdéses metszetekben a megtalált érintők láthatóan a két lapockát kötik össze, a hát azon metszetein pedig, ahol már nincs lapocka, vagy nem egyértelmű, nem érdekes, hogy mit ad az algoritmus, csak a lapocka részén vizsgáljuk az eredményt.

A helyes érintők keresése félig manuálisan történik. Mivel a felhasználó változtathatja, hogy melyik z értékhez tartozó síkmetszetet akarja látni, az ő feladata behatárolni, hogy mely érintők alapján érdemes szöveget számolni. A nyírási szöveget úgy számoljuk, hogy a felhasználó kiválaszt két z értéket (11.1./8), majd a program veszi a két érték közti összes metszősíkot, és mindegyikhez kiszámolja az adott metszetben a háthoz húzott érintő x tengellyel bezárt szögét (11.1./9-10). Az így kapott szögértékeket átlagolja, majd ezzel a szöggel a megfelelő irányba nyírást alkalmaz a modellen, így egyenesbe állítva azt (11.1./10). Természetesen a felhasználónak érdemes két olyan értéket megadni, amelyek minél távolabb vannak egymástól, hogy sok adatból lehessen átlagot számolni. Fontos továbbá, hogy a közbeeső metszetek mind értelmesek legyenek, így legtöbbször a felső érték az a metszet, ahol még nem kezdődik el a páciens nyaka (7.5. ábra) az alsó érték pedig ahol már a karjaiból adódóan rosszul behúzott érintőket is kapunk (7.6. ábra).



7.5. ábra

A páciens nyakánál már nem értelmezhető a hát elfordulásával kapcsolatban lévő érintő



7.6. ábra

A hát közepe felé a húzott érintő néhol már a hátat és a kart köti össze.

A szoftver végső kimenete jelenleg azon pontok halmaza, amelyet a nyírással beforgatott modellre a szimmetriakereső algoritmus meghatároz.

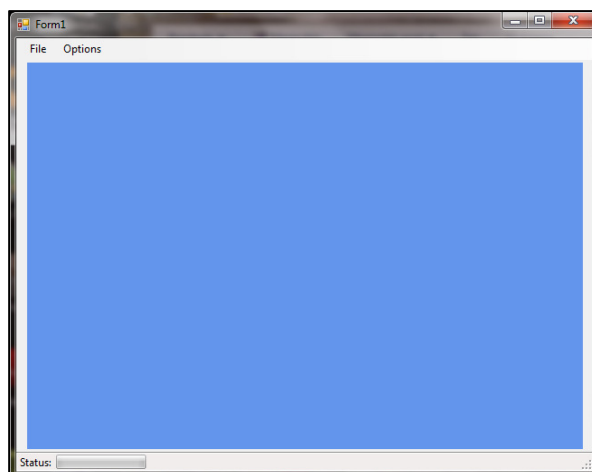
7.4. Felhasználói felület, a szoftver kezelése, megjelenítés

Mivel a szoftver jelenleg elég kezdetleges stádiumban van, ezért a kezelőfelülete sem a kényelemre fektet hangsúlyt. Tekintve azonban, hogy a hibakeresésben és a kapott eredmények értékelésében nagy szerepet játszik a könnyű és gyors kezelhetőség, pár fejlesztést kénytelen voltam alkalmazni a program megjelenésére.

A szoftver megnyitásakor megjelenik a főablak (7.7. ábra), amely egy menüsorból, a grafikus felületből és egy státuszszorból áll. Az utóbbi még nem lát el funkciókat, ez majd később kerül implementálásra. A grafikus felületnek két módja van, 3D-s illetve 2D-s üzemmód, amelyek képesek egyszerre is működni.

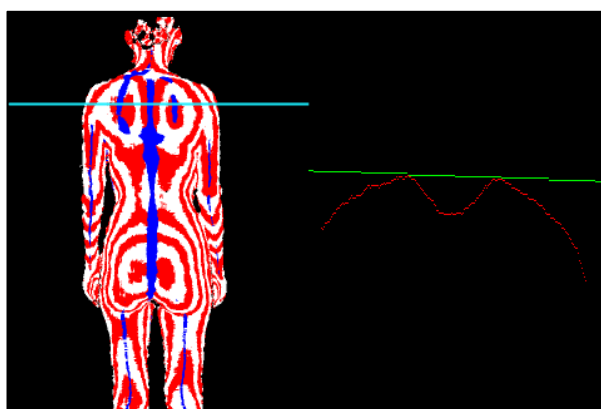
A 2D megjelenítés funkció (7.5-6, 7.8. ábra) két képet jelenít meg, a baloldalin a páciens látható hátulról, az x-z síkba vetítve, tetszés szerint csak fehérén, a körvonalat mutatva, vagy szintvonalakkal a geometriát is szemléltetve. Ezen a képen mutatja a program a megtalált „szimmetrikus pontokat”. A jobboldali képen a hát egy adott z koordináta magasságában lévő sík metszetét láthatjuk, valamint a hozzá húzott érintőt. A metszet vonalát a baloldali képen látható világoskék vonal jelzi. Ennek a metszetnek a mélységét, vagyis a metszősík z koordinátáját szabályozhatjuk a nyílombokkal. A képekre rá lehet közelíteni, bal egérgombbal kattintva a program nagyítja azokat, jobbal kicsinyíti, valamint az egérmutató vonszolásával mozgatni is lehet a felnagyított képet.

A 3D-s megjelenítési funkció (7.9. ábra) elég kezdetleges, az árnyalás nem tökéletes, mivel ezt a program írása közben egyre kevesebbet használtam, a javítása elmaradt, és csak



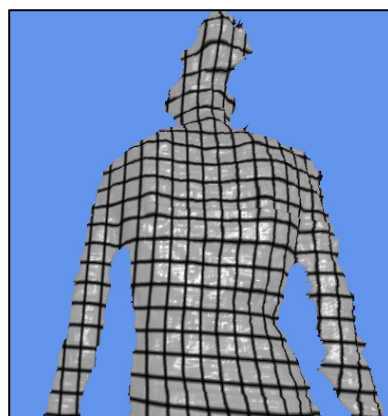
7.7. ábra

A program főablaka



7.8. ábra

2D-s megjelenítés

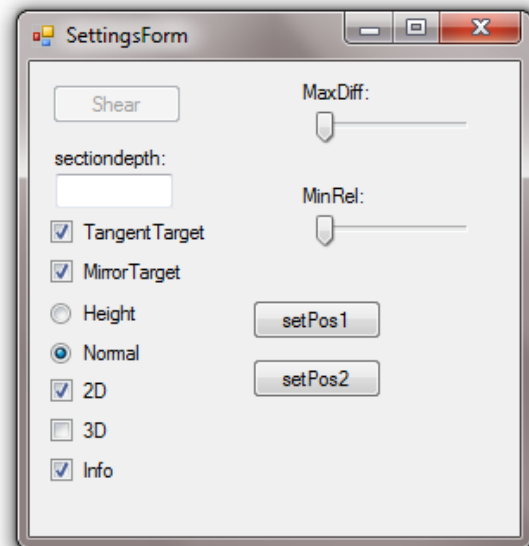


7.9. ábra

3D-s megjelenítés

említés szintjén érdekes. A 3D funkció navigálása csak akkor működik, ha a 2D funkció ki van kapcsolva, különben a lenyomott billentyűket az eseménykezelő 2D-s része dolgozza fel. A kamera gömbi koordinátarendszerben mozgatható egy adott pont körül. A pont is mozgatható külön-külön a tengelyek mentén.

A menüből a megfelelő menüpontot kiválasztva megjeleníthetők a beállítások egy külön ablakban (7.10. ábra). Ezek a beállítások a következőket tartalmazzák:



7.10. ábra
Beállítások

- Shear gomb: A nyírást végzi el az adatokon. Akkor használható, ha már mindkét értéket megadtuk a nyírási szög meghatározásához
- sectiondepth: A metszősík z koordinátájának egy adott értékre való gyors beállítását végzi.
- TangentTarget, MirrorTarget checkboxok: Az érintő, illetve a szimmetriapontok megjelenítését kapcsolják be vagy ki.
- Height/Normal rádiógomb: A modell szintvonalakkal vagy körvonalakkal való megjelenítését szabályozza.
- 2D, 3D, Info checkboxok: a 2D-s 3D-s módot, illetve az információ megjelenítését kapcsolják be vagy ki.
- MaxDiff, MinRel csúszkák: A kirajzolandó szimmetriaképben a pontok maximális eltérési értékét (aszimmetrikusságát), illetve a minimális megbízhatósági értékét adja meg (ld. 7.2. szakasz utolsó bekezdés)
- SetPos1, SetPos2 gombok: A nyírási szögének meghatározásához határként megadott két z koordináta beállítását végzi.

7. Értékelés

Bár a fejlesztés folyamatában még nem jutottunk el odáig, hogy a kapott eredményeket összevessük a páciensek röntgenjeivel, biztató eredményeket kaptam a páciensek mélységtérképeinek feldolgozásával (11. fejezet: Képgaléria). Az egészséges modelleknél a gerinc alakja egyenes volt, a betegeknél pedig attól lényegesen eltért. Némi fejlesztéssel a program nem kizárt, hogy képes legyen meghatározni a gerinc vonalát kellő pontossággal a Kinectből nyert adatok alapján. Érdemes megjegyezni, hogy a program független a Kinecttől, minthogy a bemenete egy mélységtérkép, így lehetséges más hardverrel együtt való használata is.

8. Továbbfejlesztési tervek

A program jelenleg kevés funkcióval rendelkezik, ezért azokat tovább bővítendő számos fejlesztési ötlet van a jövőre nézve:

- **A program felhasználóbaráttá tétele:** Jelenleg a program fejlesztési fázisban van. A kezelői felülete is ennek megfelelő, a fejlesztés igényeihez van szabva, nem a kényelem vagy könnyű használhatóság jellemzi. Amennyiben a további kutatás eredményei azt mutatják, hogy a szoftvert kívülállók is használni fogják, a kezelőfelület átdolgozása szükséges. Ezen felül fontos, hogy a szoftver minden bemenetre és felhasználói lépésre fel legyen készítve, ne okozzon kivételt, hanem legrosszabb esetben is hibaüzenettel reagáljon, ne pedig kezeletlen kivétellel. Ez jelenleg nincs megoldva, hiszen ez csak az elegancia szempontjából érdekes.
- **Karok levágása (képletesen):** Ha a mélységtérképből egy algoritmussal eltávolítanánk a páciens karjait, úgy minden metszethez lehetővé tesszük egy érintő húzását. Feltételezésem szerint ez sokat könnyítene a szimmetria megkeresésén, mivel a gerincferdülés torzióval is jár, a csigolyák egymáson elfordulnak, ezért a hát alsóbb részein más irányba áll a test szimmetriasíkja, mint a lapockáknál. A karok eltávolítása lehetővé tenné minden metszet külön szöggel való nyírását, így megkönnyítve a szimmetriakeresést.
- **Röntgen:** Bár a mérések egészséges emberek esetén egyenest adnak, természetesen a gerincferdüléses betegek esetén össze kellene hasonlítani a kapott képeket a páciensek röntgenjeivel.
- **Gerincgörbe:** Amennyiben használhatónak bizonyul a módszer, érdemes lenne parametrikusan is megkeresni egy közelítő térgörbét a gerinchez. Ebből akár 3D-ben a csigolyaoszlop is kirajzolható, így egy vizuális képet adva a páciens gerincéről.
- **Cobb-fok:** A térgörbe meghatározása egy, a homloksíkban felírt 2D-s görbe számítása után történne. Ennek a 2D-s közelítő görbének az ismeretében kiszámolható a Cobb-fok is az inflexiós pontokba állított érintők szögéből.
- **Cobb-fok és torzió kapcsolata:** Mindezek után a gerinc torziójának és a Cobb-foknak a kapcsolatát is vizsgálni lehetne, hátha érdekes összefüggést találunk a kettő között.
- **3D-modell építése:** Mivel a Kinect csak egy irányból tud nézni, a kép, amit kapunk, valójában nem teljes 3D-s kép, csak „2.5D-s”. Az igazi 3D-s modell építése nagyban elősegítené a korzett-tervezést, minthogy ezeket egyedileg gyártják. Ehhez több irányból készült 2.5D-s képeket kellene összeilleszteni, amire volt próbálkozás, de a Kinectek zavarták egymást. A legújabb Microsoft SDK lehetővé teszi a Kinectek gyors ki-be kapcsolását, amivel ez a probléma kiküszöbölhetővé válik. Így a négy irányból készült felvételekből 3D-s modellt lehetne építeni.

9. Összefoglalás

Dolgozatomban egy gerincferdülés diagnosztizálására használható program fejlesztését, a hozzá tartozó algoritmusok kidolgozását mutattam be. A program újszerű, mert 3D-s adatokból, térbeli geometriából számolja a gerinc vonalát, amire eddig berendezés hiányában nem volt lehetőség. Erre szolgáltatott megoldást a Microsoft új mélységszenzora, a Kinect, ami az árához képest meglehetősen nagy pontossággal bír. A program alapvetően a 2D-s szimmetria vizsgálatával próbálja megkeresni a „hát közepét”, a gerinc pontjait. Jelenleg kevés mintát sikerült feldolgoznunk, ám ezekre biztató eredményeket kaptunk, és nem kizárt, hogy a program képes legyen a módszer segítségével használható adatokat szolgáltatni a páciens egészségi állapotára nézve.

Az ilyen irányú kutatások ötlete Dr. Wenzel Klárának, a BME MOGI tanszék munkatársának köszönhető.

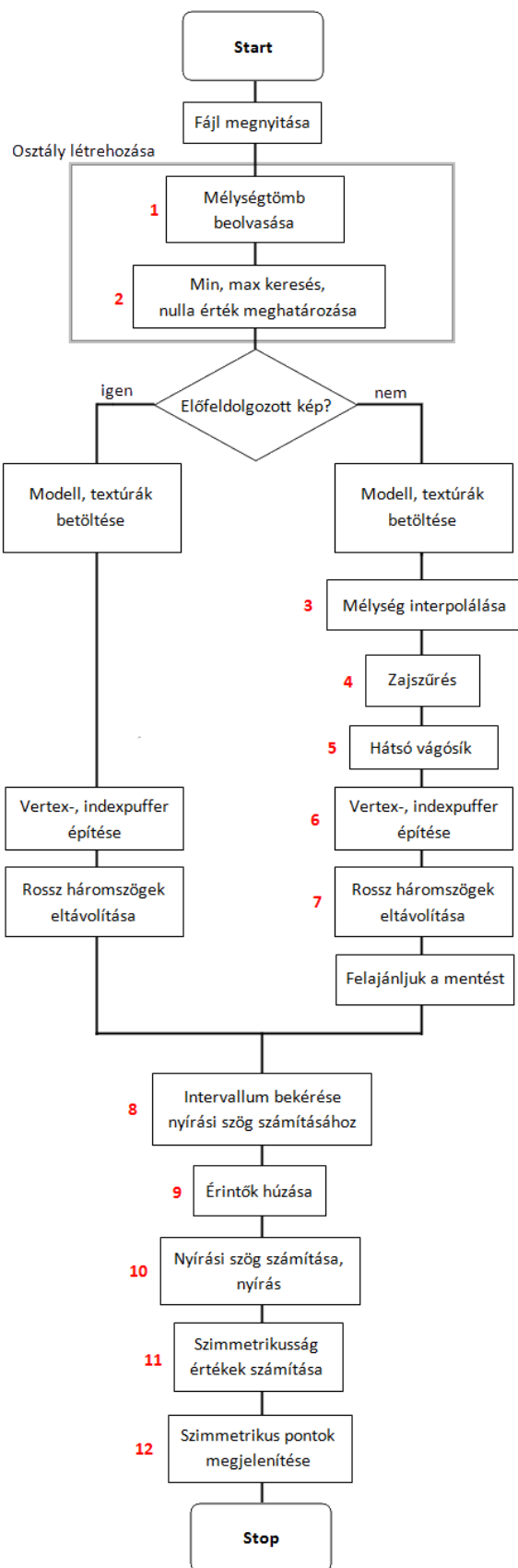
A munka a Nemzeti Innovációs Hivatal a GERINCO2 TECH_08-A1/2-2008-0121 kutatási projekt keretében nyújtott támogatása nélkül nem jöhetett volna létre.



10. Irodalom

- [1] dr. Csernátorny Zoltán: A scoliosis-csapda – tévedések, tévhitek és tévutak.
Biomechanika, terápia, prognosztika és egy kis pszichológia
Gyermekegyógyászat, 56. évf. 6. szám
- [2] Grósz Gábor: Optikai elven működő orthopéd diagnosztikai eszköz fejlesztése,
Szakdolgozat, 2012
- [3] Ádám J Kocsis, Attila Drajkó, Péter Tamás, Ákos Antal:
Moiré Image Based Three-Dimensional Surface Mapping for Spinal Diagnostics.
In: 8th International Conference on Mechanical Engineering 2012. Budapest,
Magyarország, 2012.05.24-2012.05.25. pp. 97-103.
- [4] Attila Drajkó, Ádám J Kocsis, Péter Tamás, Antal Ákos
Computer-Aided Moiré Image Processing to Determine the Angle of Spinal
Curvature.
In: Proceedings of the 8th International Conference on Mechanical Engineering
2012. Budapest, Magyarország, 2012.05.24-2012.05.25. Budapest: pp. 90-96.
- [5] dr Wenzel Klára, dr Antal Ákos, Szakály Norbert, dr Tamás Péter, Tóth Bertalan:
Scoliosis moiré vizsgálata.
In: OGÉT 2012. Kolozsvár, Románia, 2012.04.19-2012.04.22. Kolozsvár: pp. 503-
506.(ISBN: ISSN 2068-1267)
- [6] Wenzel Klára, Antal Ákos, Molnár József, Tóth Bertalan, Tamás Péter
New Optical Equipment in 3D Surface Measuring.
JOURNAL OF AUTOMATION MOBILE ROBOTICS & INTELLIGENT SYSTEMS 3:(4) pp.
29-32. (2009)
- [7] M.R. Andersen, T. Jensen, P. Lisouski, A.K. Mortensen, M.K. Hansen, T. Gregersen,
P. Ahrendt: KINECT DEPTH SENSOR EVALUATION FOR COMPUTER VISION
APPLICATIONS, Aarhus University, 2012 február
- [8] <http://msdn.microsoft.com>
- [9] Szirmay-Kalos László: Számítógépes Grafika, ComputerBooks, Budapest, 1999
- [10] <ftp://www.inflab.bme.hu/pub/oktatas/XNA%20programozas/>

11. Képgaléria



11.1. ábra

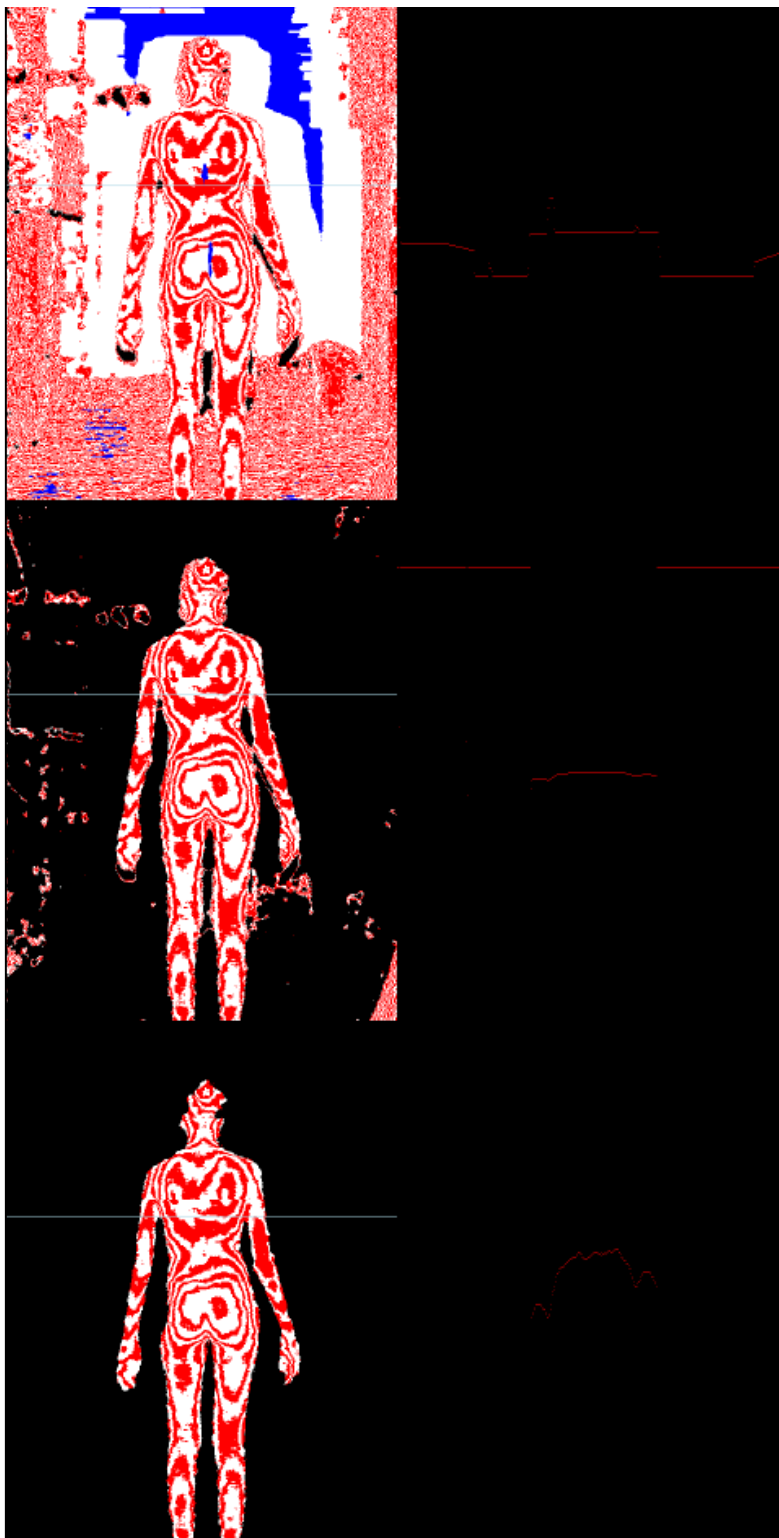
A program blokkdiagramja

A következő képek a feldolgozás menetét hivatottak szemléltetni. A szemléletesség kedvéért a 4-es és 5-ös lépéseket (ld. 11.1. ábra) felcseréltem. Ennek csak demonstrációs okai vannak, a szoftver kimenetére nincs hatással.

11.2. ábra

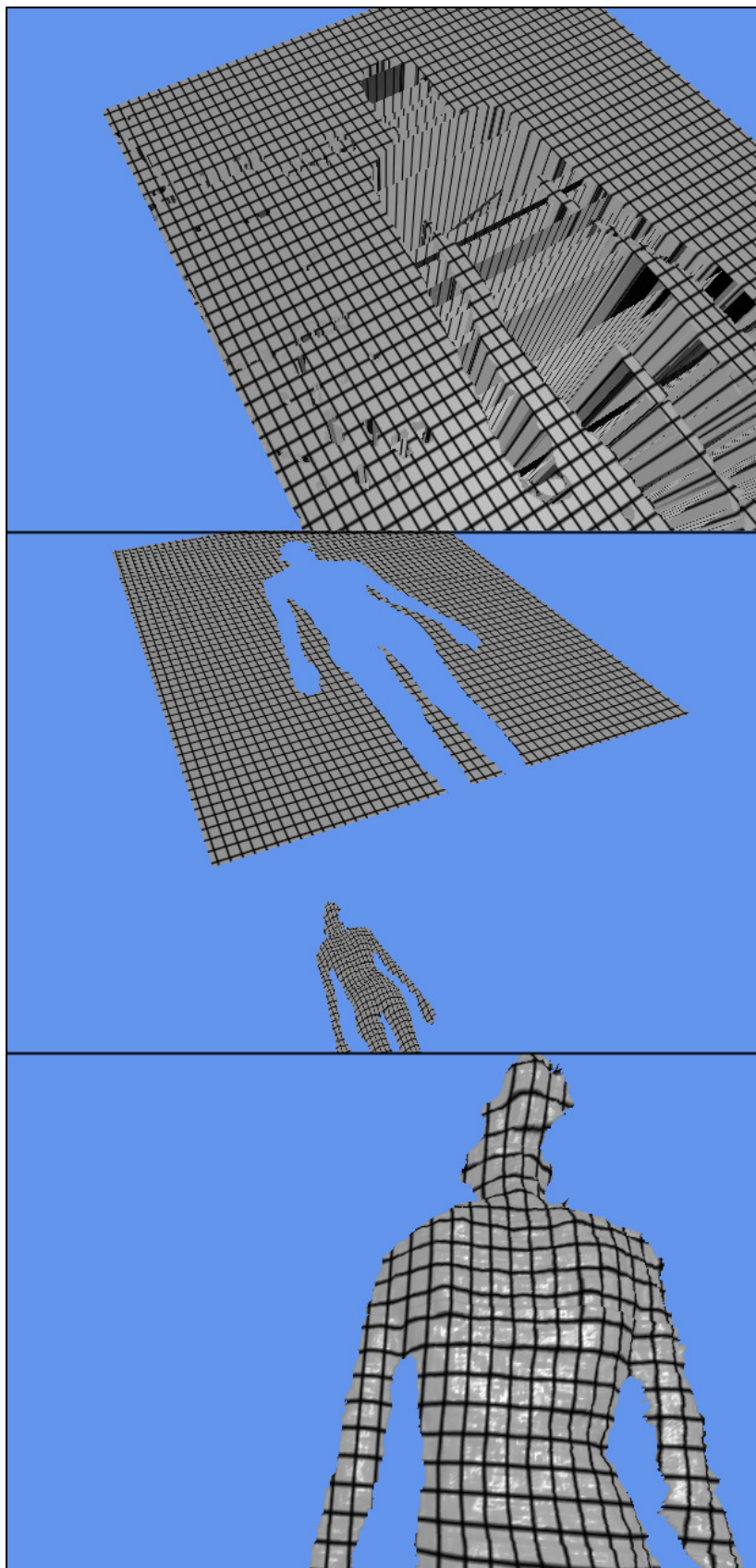
Felül: A feldolgozatlan kép. Középen: A kép a hátsó vágósíkkal való vágás (10.1./5) után.

Alul: A feldolgozott kép, zajszűrés (10.1./4) után.



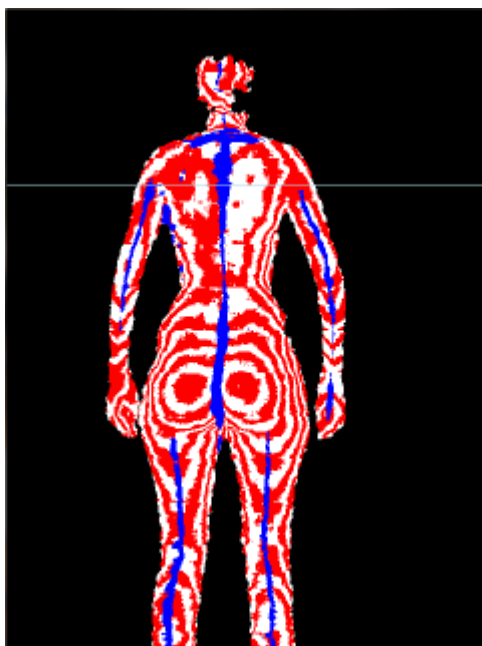
11.3. ábra

*Felül: A feldolgozatlan kép. Középen: A kép a nagy területű háromszögek eltávolítása után.
Alul: A nulla adatot tartalmazó háromszögek eltávolítása után a feldolgozott kép.*

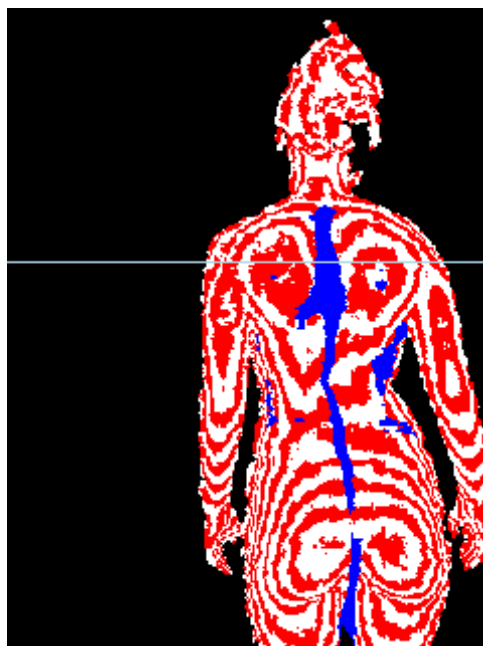


A mérési eredmények: A baloldali (11.4, 11.6.) ábrákon egészséges pácienseket láthatunk, a másik kettőn gerincferdüléseket. A 11.5. ábrán nem látszik annyira a páciens gerincferdülése, ez adódhat pl. a hölgy tartásából is, vagy egyszerűen abból, hogy neki kisebb mértékű gerincferdülése van. Ez a minta egyébként is kavarzóknak számít, mert ez az egyetlen, ami – valami különös oknál fogva – 320*240-es felbontású.

11.4. ábra



11.5. ábra



11.6. ábra



11.7. ábra

