



Budapesti Műszaki és Gazdaságtudományi Egyetem
Villamosmérnöki és Informatikai Kar
Irányítástechnika és Informatika Tanszék

3D modellezés keresztmetszeti kontúrok alapján

TDK dolgozat

Készítette:

Sánta Márton

Konzulens:

Dr. Csébfalvi Balázs

2020

Tartalomjegyzék

Kivonat	i
Abstract	ii
1. Bevezetés	1
1.1. A téma motivációja	2
1.2. A dolgozat további felépítése	3
2. Az áttekintett irodalom bemutatása	4
2.1. Alaktartó interpoláció	4
2.2. Implicit függvényekkel történő megközelítés	6
2.3. Szórt adat interpoláció	7
3. A fizikai modell elkészítésétől a fényképezésig	10
3.1. A fizikai modell elkészítése agyagból	10
3.2. A fizikai modell felszeletelése	12
3.3. A szeletek lefényképezése	12
4. A számítógépes feldolgozás lépései	15
4.1. A kontúrok detektálása	15
4.2. A távolság-transzformáció kiszámítása	18
4.3. A modell megjelenítése	20
5. A feldolgozás lépéseinek tesztelése	25
6. Az eredmények bemutatása valós szobrok esetén	27
7. Összefoglalás és jövőbeli célok	36

Kivonat

A 3D modellezés és rekonstrukció a számítógépes grafika egyik kiemelt és aktívan kutatott területe. Alkalmazása kiterjed az orvosi diagnosztikára, mint például volumetrikus adatok (CT, MRI stb.) megjelenítésére és vizualizációjára, de a játékfejlesztés és a 3D animáció területén szintén fontos.

A modellek elkészítése történhet 3D szerkesztőprogramok, vagy valós modellek felhasználása esetén különböző képalkotó berendezések segítségével (lézer szkennelés, tomográfia). Egy lehetséges módszer a modellezésre, a CT felvételek esetén is alkalmazható, keresztmetszeti kontúrok alapján történő alakrekonstrukció, amikor a modellezendő objektum valamilyen keresztmetszeti felvételei állnak rendelkezésre és az ezeken detektált kontúrok szolgálnak a feldolgozás alapjául.

Mivel az orvosi képalkotó berendezések használata meglehetősen drága és körülményes, egy praktikus és egyszerűbb alternatívát jelenthet a jelen dolgozat témájául is választott eljárás. A kiindulási alapot egy valós, agyagból megmintázott és digitalizálni kívánt objektum jelenti, amely felszeletelésével és a szeletek lefényképezésével nyerhetők a keresztmetszeti kontúrok. Ezután a kontúrok távolságtaszformációjával egy 3D skalármező számítható, amelynek zérus értékhez tartozó izofelülete szolgáltatja számunkra a digitalizált modellt.

A dolgozat egy konkrét eseten keresztül igyekszik bemutatni az eljárás és a feldolgozás lépéseit, elemezve azok előnyeit és hátrányait, valamint értékelve a módszer segítségével elérhető pontosságot és látványt, kitekintve a jövőbeli fejlesztési lehetőségekre is.

Abstract

3D modeling and reconstruction is one of the most actively researched areas of computer graphics. Its application extends to medical diagnostics such as the display and visualization of volumetric data (CT, MRI, etc.), but it is also important in the field of game development and animation.

Models can be created with 3D editing programs or in case of real models by the help of different imaging equipment (laser scanning, tomography). One method of modeling, which can also be used for CT images, is shape reconstruction based on cross-sectional contours, where cross-sectional images of the object to be modeled are available and the processing is based on the detected contours on them.

As the usage of medical imaging equipment is quite expensive and difficult, a procedure discussed in this paper may be a practical and simpler option. The starting point is a real object made from clay that is intended to be digitized, and by slicing and photographing the slices the cross-sectional contours can be obtained. Then, by a distance-transformation of the contours, a 3D scalar field can be computed and its zero value iso-surface provides us with the digitized model.

This paper presents the steps of the procedure through a specific case, analyzing their advantages and disadvantages, as well as evaluating the accuracy and visibility that can be achieved with the help of this method, also looking at future development opportunities.

1. fejezet

Bevezetés

A 3D rekonstrukció és modellezés kiemelten fontos és aktívan kutatott téma a számítógépes grafika témakörén belül. Alkalmazása kiterjed a mérnöki tervezésre (CAD), orvosi diagnosztikára, orvosi adatok 3D vizualizációjára, animációk készítésére és még sok egyéb más grafikai területre.

A modellek megalkotásának egyik lehetősége a számítógépes szerkesztő programok használata, amelyek ismerete és használata nagy jártasságot és sokszor komoly tapasztalatot igényel. A másik megközelítés konkrét, megvalósított fizikai objektumokon végzett mérések segítségével készíti el a 3D modelleket. Ez utóbbi esetben is több lehetséges választással élhetünk, attól függően, hogy a méréseket milyen rendszerrel végezzük el.

Orvosi területen gyakran alkalmazott eszközök például a CT és az MRI. Ezek a szkennelni kívánt objektumról különböző irányokból vetületi felvételek készítenek, majd egy feldolgozási láncon keresztül rekonstruálnak egy 3D volumetrikus adathalmazt, amely diszkrét térbeli pontokban, például az adott pontban jelen lévő anyag sűrűségértékét tartalmazza. Ezeknek a berendezéseknek a használata meglehetősen drága, nem alkalmazható tehát akármilyen területen.

Másik lehetőség a lézerszkennerek használata, amikor is az adott objektumról különböző irányokból készítve mélységképeket (a kamerától vett távolságok a lézer visszaverődési idejéből határozhatóak meg), az ezekből generált pontfelhők összeillesztése a feladat. A módszer ugyan olcsóbb, mint az orvosi berendezések használata, még mindig nehezen hozzáférhető egyszerűbb alkalmazásokhoz, illetve a mérési pontatlanságok is nagyok lehetnek, hiszen a fény nagy sebessége ($c \approx 3 * 10^8 \frac{m}{s}$) miatt

kis időtartamok pontos mérése a feladat (1mm-es pontosság esetén $\approx 3.33 * 10^{-12} s$), illetve a feldolgozás is számításigényes és bonyolult.

Az előbbieken említett eljárásoknak, viszont nagy előnye az érintésmentes mérés, vagyis a beszkenelt objektum nem sérül, nem károsodik az eljárás során.

1.1. A téma motivációja

A gyakorlatban a téma azért is fontos, mivel több olyan projekt is megvalósult mindezidáig, amik az ehhez hasonló feldolgozási folyamatok potenciális bemeneteit képezhetik. Az egyik legismertebb a *Visible Human Project*, aminek keretein belül eddig három donort is felszeleteltek haláluk után. A szeletelés előtt a testeket lefagyasztották, majd milliméter, illetve az alatti szeletvastagsággal vágták azokat fel. A szeleteket ezután analóg és digitális módon is lefényképezték, amiből aztán több gigabájtnyi adat keletkezett. Hasonló módszerrel manapság is terveznek nagy felbontással felválni testrészeket és azokat hasonló módon dokumentálni, illetve már fel is lelhetőek ilyen jellegű adathalmazok, például emberi fogról.

Animációk készítésével foglalkozó cégeknél előforduló példa, hogy a számítógépes grafikai szakemberek mellett képzőművészek is dolgoznak, akik például agyagból megmintázott plastikákat készítve biztosítanak jó kiindulási alapot a modellezéshez, amelyet a munka során számítógépekkel még szűrnék, szerkesztenek, finomítanak. Az elkészített szobrok digitalizálása olyan mérnöki feladat, amelynek egy lehetséges megoldásában alkalmazható például a fentebb is említett lézerszkennelési technika. A feldolgozás azonban más módon is elképzelhető és erre példa a keresztmetszeti kontúrok alapján történő 3D modellezés.

A dolgozat egyik fő motivációja tehát, hogy a keresztmetszeti kontúrok alapján számított rekonstrukció az egyéb más alternatívákkal szemben egy olcsóbb és skálázhatóbb megoldást kínáljon. A cél egy olyan 3D modellezési eljárás kidolgozása, amelynek eszközigénye könnyen hozzáférhető bárki számára, és jó minőségű, akár nagy felbontású, precíz rekonstrukciót tesz lehetővé. A módszer jól alkalmazható számítógépes animációk, játékok karaktereinek modellezésére, ugyanakkor az alkalmazott számítások és a feldolgozás menete ennél általánosabb és akár a korábban

említett orvosi területre is kiterjeszthető, például CT adatokból történő kiindulás esetén.

A dolgozat egy konkrét gyakorlati alkalmazást mutat be, amely a fizikai modell elkészítésétől a digitalizált reprezentáció előállításáig követi végig a folyamatot. Ehhez mindösszesen egy kamerával felszerelt állvány, egy agyagból elkészített plasztika és a programot futtató számítógép szükséges.

1.2. A dolgozat további felépítése

A továbbiakban a területhez tartozó elméleti háttérrel, illetve a kapcsolódó szakirodalmat ismertetem és azok kapcsolódását a jelen dolgozathoz. Ezt a 2. fejezetben teszem meg. Ezt követően bemutatom a feldolgozás lépéseit, szétválasztva a fizikai modellezés és előkészületek szakaszát (3. fejezet) a számítógépen implementált algoritmusoktól (4. fejezet). Ezt követően a teszteseteket (5. fejezet) és az eredményeket mutatom be (6. fejezet), majd a dolgozatot a jövőbeli célok megfogalmazásával és egy rövid összefoglalóval zárom (7. fejezet).

2. fejezet

Az áttekintett irodalom bemutatása

A témával, illetve az ehhez kapcsolódó elméleti háttérrel számos publikáció foglalkozik, amelyek közül a legfontosabbakat a jelen fejezetben tekintem át, bemutatva a különböző megközelítéseket és megoldásokat. Mindezek mellett azokat az elméleti alapokat is összefoglalom, amelyek a bemutatott módszer továbbfejlesztéséhez szükségesek.

2.1. Alaktartó interpoláció

Az alaktartó interpolációs (*shape-based interpolation*) megközelítéssel foglalkozó, egyik kiemelten fontos és sokszor hivatkozott munkát Bors és társai publikálták [1]. A módszerük általában bináris képekből indul ki és az ezeken számított távolságtaszformáció, illetve különböző morfológiai műveletek segítségével közelíti meg a feladatot.

Az említett cikkben morfológiai műveletek segítségével a rendelkezésre álló binarizált szeletfelvételek között további szeleteket generálnak és ezáltal egy sokkal nagyobb felbontású adathalmaz képezi a pontosabb modellalkotás alapját. A bemutatott konkrét esettanulmányban különböző típusú fogak rekonstruálását végezték el és ezek eredményein keresztül mutatták be az alkalmazott módszer működőképességét és hatékonyságát. Ehhez olyan adatbázisból indultak ki, amelyben egy-egy fog mechanikai felszeletelése utáni beszkenelt felvételek szerepelnek. A képek szegmen-

tálásával bináris képeket állítottak elő és így vált lehetővé az említett morfológiai műveletekkel való megközelítés.

Ha P és Q a két bináris objektum, amelyek között az átmenetet szeretnénk megvalósítani, akkor P -ből a Q -ba való átmenetet a következő művelet többszöri, egymásutáni végrehajtásával számítjuk ki, ahol a leállási feltétel az idempotencia:

$$f(P|Q, B) = [(P \ominus B) \cup ((P \cap Q) \oplus B)] \cap (P \cup Q) \quad (2.1)$$

Itt B a strukturáló elemet jelöli, a fenti kifejezés pedig azt a folytonos átmenetet fejezi ki, amely szerint P erodálása a B -vel éppen P fogyását eredményezi, míg P és Q metszetének dilattálása az átmenetet növeli B -től függő mértékben. Járulékos feltételként a műveleteknek a két bináris objektum unióján belül kell megvalósulnia.

Egy másik munkát, amely hasonlóan bináris képeket, illetve arra számított távolság-transzformációt alkalmaz, Alencar Lotufo és Xavier Falcao publikáltak [2]. Megközelítésükben a bináris szeletképekre előjeles távolság-transzformációt számítottak ki, amelyet kombináltak a szürkeárnyalatos képekből kinyerhető információval. A kettő segítségével ők is köztes szeleteket generáltak, pontosítva ezzel a modelljüket.

A két módszer egyik hátránya, hogy bináris képekkel operálnak, amelyek a véges felbontásból adódóan a rekonstruált felület egyenetlenségét fogják eredményezni. A rekonstrukciót a meglévő szeletek számának növelésével érik el, nem váltanak reprezentációt a megjelenítéshez. Ehhez hasonló eredmények érhetőek el, ha a dolgozatban tárgyalt módszert egyszerűen bináris képekre alkalmazzuk, hiszen ezekben az esetekben számított kontúrok éppen a háttér és objektum közötti, pixelhatárookra illeszkedő szegmensekből fognak állni. A felület megjelenítésénél jól megfigyelhetőek a pixeleségből adódó artifaktumok.

Egy másik hátrány, hogy a bemutatott módszerek a közbenső szeletek generálása után a modell előállításában nem vesznek figyelembe egyéb, például simaságra vonatkozó kritériumokat és megkötéseket, mint ahogyan teszi azt a következő részben ismertetett eljárás.

2.2. Implicit függvényekkel történő megközelítés

A másik megközelítés az implicit függvényeken alapuló interpolációs technika. Ebben az esetben a keresztmetszeti kontúrokat 2D implicit függvények adott értékhez tartozó szintvonalai reprezentálják. Turk és O'Brien is ezt a megközelítést használják [11] folytonos és megfelelő simaságú alaktartó interpolációra.

Az implicit függvény megválasztására kézenfekvő választás lehet az előjeles távolság-transzformáció, amely a zárt görbe belsejében pozitív, kívül negatív értékeket (ez akár meg is fordítható), a görbe mentén pedig pontosan zérus értéket vesz fel. Ezek az értékek a kontúrtól való távolságot reprezentálják. Ez a függvény azonban nem kellően sima és meglehetősen durva éleket tartalmaz, ami miatt a cikk szerzői egy hasonló módon definiált, de más módon előállított implicit függvényt javasolnak.

A kontúr mentén bizonyos mintapontok kijelölése után, meghatározták ezekben a pontokban a normális irányokat, amik a görbére merőlegesek az adott helyeken. A mintapontokban, valamint a normális egyenes mentén a görbe belsejének irányában és ezzel ellentétesen is egy kis lépésközzel kényszereket definiáltak. A kényszerek a fent leírtaknak megfelelőek, a görbe belsejében pozitív, kívül negatív, a görbén zérus értéket írnak elő, majd ezek alapján határozták meg az implicit függvényt szórt adat interpoláció (*scattered data interpolation*) segítségével, amelyről a következő részben lesz szó részletesebben.

Az implicit függvény zérus értékhez tartozó szintvonala definiálja a görbét, amely a megfelelő simasági kritériumoknak eleget tesz. Mivel azonban nem 2D-ben szeretnénk a kontúrt egy függvénnyel reprezentálni, hanem több keresztmetszeti kontúr alapján a 3D felületet szeretnénk visszaállítani, ezért a cikk szerzőinek ajánlása nyomán, célszerű nem külön kezelni az egyes keresztmetszeti kontúrokat, hanem egyetlen 3D implicit függvényt definiálni a fent leírtak megfelelő általánosításával. Tehát a kényszereket már nem pusztán a síkban, hanem a pontokat egy harmadik koordinátával kiegészítve a térben definiálták. Több keresztmetszeti kontúr esetén ezeket a kényszereket a 3D térbe ágyazva egy magasabb dimenziós implicit függvény állítható elő, amelynek zérus értékhez tartozó szintfelülete definiálja a végeredményt, azaz a simasági kritériumokat kielégítő, folytonos felületreprezentációt.

2.3. Szórt adat interpoláció

A szórt adat interpoláció (*scattered data interpolation*) [6] technika lényege, hogy rendelkezésre állnak elszórtan mintavételi pontok, ahol ismerjük egy interpolálni kívánt függvény által felvett értékeket. A cél egy olyan függvény definiálása, amely pontosan illeszkedik a mintavételi pontokban. A feladatra több megoldás is kínálkozik, mint például:

- Shepard-interpoláció: a köztes pontokban a függvény értékét a mintavételi pontokra vett értékek súlyozott összegéből határozza meg, ahol az egyes súlyok az adott mintavételi pontoktól vett távolságok reciprokai (lsd. [9])
- Kernel regresszió (approximál a kijelölt pontokban)
- *Moving Least Squares* módszer (approximál a kijelölt pontokban) [4]
- Lineáris interpoláció
- Radiális bázisfüggvények (*Radial Basis Function* - RBF) használata (lsd. [10])

A Radiális bázisfüggvények használata kiemelten fontos Turk és O'Brien tárgyalása szempontjából. Ebben az esetben a megoldást a következő alakban keressük:

$$\hat{f}(x) = \sum_{k=1}^N w_k \phi(\|\mathbf{x} - \mathbf{x}_k\|) \quad (2.2)$$

ahol a ϕ függvény az RBF és értéke csak az egyes \mathbf{x}_k mintavételi pontoktól vett távolságtól függ. A RBF megválasztásához számos szempontot lehet vizsgálni, például az interpolált függvény differenciálhatóságát, a folytonossági rendjét vagy az összegörbületének minimalizálását. Az RBF családból az úgynevezett *thin plate spline* kielégíti 2D-ben a görbület minimalizálására vonatkozó alábbi előírást.

$$\phi(r) = r^2 \log(r) \quad (2.3)$$

$$E = \int_{\Omega} f_{xx}^2(\mathbf{x}) + f_{yy}^2(\mathbf{x}) + 2f_{xy}^2(\mathbf{x}) \quad (2.4)$$

ahol E az összegörbület, amit minimalizálni szeretnénk.

Az egyes mintapontokban előírt értékekre vonatkozó kényszerek felírásából egy lineáris egyenletrendszer nyerhető, amely megoldásával a fenti kifejezésekben szereplő egyes feltételek kielégíthetők.

$$\begin{bmatrix} \phi_{11} & \phi_{12} & \phi_{13} & \cdots \\ \phi_{21} & \phi_{22} & \phi_{23} & \cdots \\ \phi_{31} & \phi_{32} & \phi_{33} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ \vdots \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ \vdots \end{bmatrix} \quad (2.5)$$

$$\phi_{ij} = \phi(\|\mathbf{x}_i - \mathbf{x}_j\|) \quad (2.6)$$

$$f_i = f(\mathbf{x}_i) \quad (2.7)$$

A dolgozat későbbi részeiben ismertetett implementációhoz, a fentiekhez hasonló, implicit függvénnyel történő megközelítést alkalmaztam. A dolgozat célja egy automata rendszer elkészítése volt, ami a paraméterek hangolásán túl nem igényel egyéb felhasználói beavatkozást. Az implicit függvény, amely a keresztmetszeti kontúrokat definiálja éppen ezért az RBF reprezentációval ellentétben az Euklideszi előjeles távolság-transzformált. Ennek oka, hogy a kontrollpontok automatikus kiválasztása nem triviális feladat, így az RBF reprezentáció kiszámítása sem egyszerű. További kihívást jelenthet, ha a kontrollpontok egymáshoz nagyon közel esnek és ezáltal a kényszereket definiáló egyenletrendszerben a mátrix rosszul kondicionált, ami numerikus instabilitáshoz vezethet.

A távolság-transzformáció alkalmazásával, viszont egy kényelmes reprezentációt kapunk. Ez alapján ugyanis a felületi normálisok jól számolhatóak, a jellemzés pedig robusztus, néhány kontúrponthiánya sem jelent gondot. Az egyik legfőbb előnye, hogy nem kell a detektált kontúrponthiányok közötti pontonkénti megfeleltetéssel foglalkozni. Bizonyos megközelítések ugyanis a modellt úgy próbálják megalkotni, hogy a szomszédos kontúrok adott pontjait egymással párba állítják, és ebből a párosításból definiálnak a megjelenítéshez szükséges háromszög elemeket. Ez a fajta megközelítés nem képes különböző topológiájú kontúrokat megfelelően kezelni, hiszen a modellezés szempontjából kritikus a megfeleltetés eredménye és így a folytonos átmenet nem minden esetben biztosított. Ezzel a kérdéssel foglalkozik Meyers és Skinner [8]

is. Az implicit reprezentáció éppen ezt küszöböli ki, hiszen a távolság-transzformáció ismeretében nem kell a különböző topológiájú eseteket külön kezelni, az ezek közötti folytonos és szép átmenet a reprezentációváltásból adódóan automatikus.

További előny, hogy amennyiben a felület túlságosan is egyenetlen, simább felület nyerhető ki, ha nem pontosan a zérus értékhez tartozó szintfelületet rendereljük, hanem egy attól távolabbi burkolót. Hibák forrása lehet azonban a hibásan kontúrként detektált és szegmentált zajos, zavaros régiók jelenléte. Ezek ugyanis a távolságmezőt teljesen el tudják torzítani, viszont előfeldolgozással és szűrők alkalmazásával ezek az esetek elkerülhetőek.

3. fejezet

A fizikai modell elkészítésétől a fényképezésig

A feldolgozási lánc bemenetét az egyes szeletekről készült képek adják, melyek előállításához az ebben a fejezetben tárgyalt lépések szükségesek. Ezeket a lépéseket a 3.1 ábra foglalja össze.

3.1. A fizikai modell elkészítése agyagból

Az agyagból történő mintázás az első lépése a feldolgozásnak, ahol mintázó eszközök segítségével egy plasztika formázása a cél. A mintázáshoz szükségesek bizonyos alapfokú ismeretek, azonban egyszerűbb alakzatok kialakítása kevesebb gyakorlattal is kivitelezhető, hiszen például elegendő (sőt kívánatos is) ha a modell tömör, nem tartalmaz üregeket, amelyek kerámia szobrok esetén szükségesek lennének.

A forma kialakításánál fontos szempont a leendő szeletek vastagsága, amelynél kisebb részletek kialakítása nem célszerű a szeletelésre merőleges irányban, hiszen a feldolgozás során azok elveszhetnek. Az agyag használatának nagy előnye a könnyű formázhatósága, a hibák kényelmes javíthatósága, megfelelően nedves környezet biztosításával a modellezési idő elnyújtása, nem utolsósorban pedig az agyag olcsósága és könnyű hozzáférhetősége.



3.1. ábra. A folyamat lépéseinek szemléltetése a fizikai modelltől a fényképezésig. Látható az elkészült modell, amit a későbbiekben rekonstruálni szeretnénk, az állvány építéséhez használt elemek, illetve a referencia lyukak kijelölése. Ezt követi a szeletelés, a szeletek rögzítése és lefényképezése, az eredmény pedig a kapott képek összessége.

3.2. A fizikai modell felszeletelése

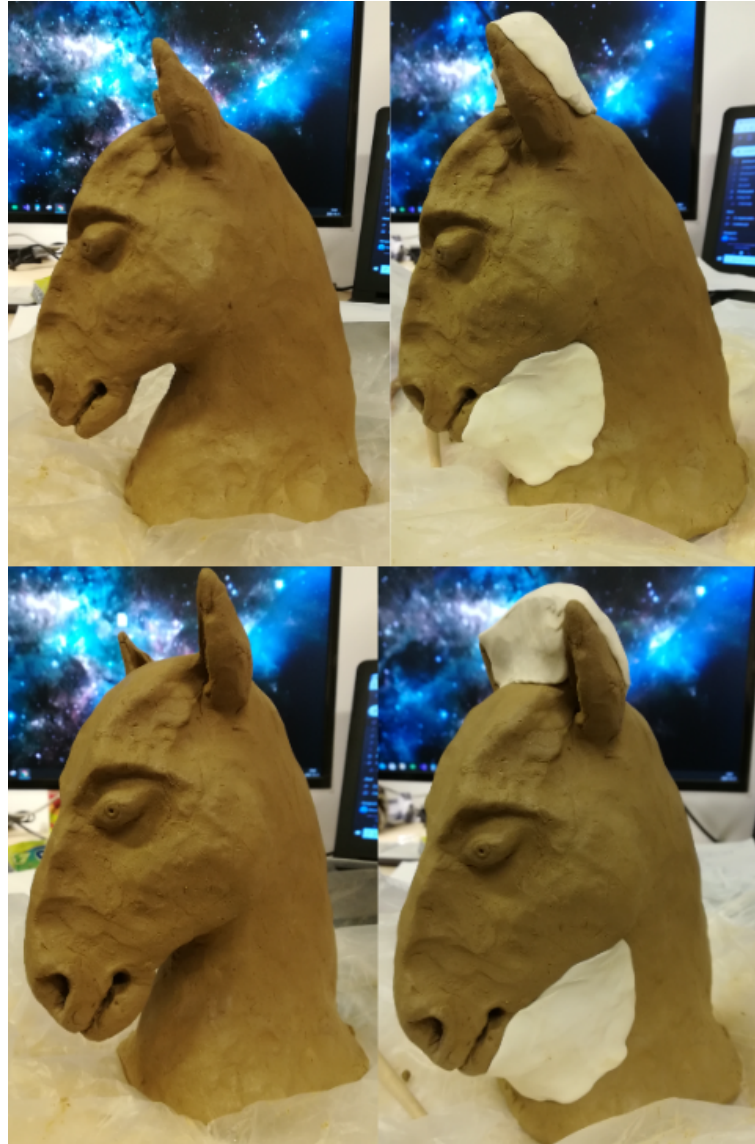
Az agyagból mintázott plasztikát kellő ideig szárítva, még könnyen szeletelhető marad, ugyanakkor a szeletek könnyebben kezelhetőek és nehezebben deformálódnak el a mozgatások során. A szeletvastagság megválasztásánál ügyelni kell a szeletelés irányában a részletezettségre, hiszen tulajdonképpen ezzel egy mintavételezés történik, amelyre vonatkozó mintavételi törvény betartása itt is szükséges.

A szeletelés egy vékony dróttal, húrral végezhető el, aminek alátámasztására meghatározott vastagságú falapok használhatóak, amelyek vastagsága mind a megkívánt szeletvastagságnak egész számú többszöröse (jelen esetben ez a vastagság 0.75 cm). A változó vastagságú fadarabokra azért van szükség, mert azok kialakítása során is számolni kell a gép, illetve a gépkezelő pontatlanságaival. Ha csak a szeletvastagságnak megfelelő vastagságú falapokkal dolgoznánk, azokat egymásra pakolva a hiba egyetlen gépbeállítás esetén kumulálódna, míg a többszöröseit is alkalmazva, az egyetlen darab esetén előforduló hiba megszlik, illetve több darab esetén a különböző gépbeállítások esetén az egyes hibák gyengíthetik egymás hatását a végeredményben.

A szeletelés esetén fontos kérdés, hogyan lehet bonyolultabb alakzatokat is úgy szeletelni, hogy azok ne essenek szét szeletenként több darabra, hiszen akkor azok egymáshoz viszonyított megfelelő elhelyezéséről is gondoskodnunk kell a továbbiakban. A probléma és megoldása analóg a 3D nyomtatásnál alkalmazott strukturáló elemek használatával. Erre egy lehetséges megoldás a modell kipótolása a kritikus helyeken megfelelően megválasztott színű agyaggal, vagy a forma körbeöntése egy befoglaló test segítségével szintén eltérő színű agyaggal. Az előbbire látható példa a 3.2 ábrán is.

3.3. A szeletek lefényképezése

A feldolgozáshoz szükség van az egyes szeletek digitalizálására. Itt fontos szempont a háttér megválasztása, illetve a szeletek megfelelő pozicionálása az egyes képeken. Szeretnénk ugyanis elkerülni a későbbiekben az egyes kontúrok közötti pontonkénti megfeleltetést.



3.2. ábra. Az egyik szobor kipótolásának bemutatása. A ló füleinél és a nyakánál azért volt fontos a fehér színű kipótolás, hogy a szeletelés közben a modellt ne essen szét több darabra és ne kelljen az egyes darabok egymáshoz viszonyított elhelyezésével foglalkozni a fényképezés előtt.

Az agyag többféle színben is beszerezhető, illetve színezhető, kellően világos (fehér) háttérrel azonban az alap barnás színe is tökéletesen megfelelő. Ennek jelentősége a szegmentálásnál van, ahol szeretnénk minél pontosabban megkülönböztetni egymástól a háttér és a szelet pontjait.

Ahhoz, hogy a képeken az egyes szeletek méreteinek aránya a valósággal megegyezzen, fontos, hogy azonos távolságból kell a képeket elkészíteni, amelyhez egy

állvány is szükséges. Ezzel a módszerrel akár még egy kalibrációt követően pontos, az eredeti méretekről is információt hordozó modell készíthető.

Ahhoz, hogy a szeleteken detektált kontúrok esetében ne kelljen foglalkozni a pontok egymásnak való megfeleltetésével a szomszédos szeleteken, fix kamera pozíció mellett elég azt garantálni, hogy minden szelet legalább két fix pontja minden felvételen ugyanott látszódjon, ha az elrendezés a korábbiakban vázoltaknak megfelelő. Egyik lehetőség, hogy a fizikai modellt átszúrjuk a képsíkra merőlegesen legalább két pálcával, amelyeket a szeletelés előtt eltávolítunk (a két lyukra az elfordulások kiküszöbölése miatt van szükség). Az állványzat alján elegendő azt a két pontot kijelölni, amihez minden szeletet igazítani kívánunk a fényképezés előtt.

4. fejezet

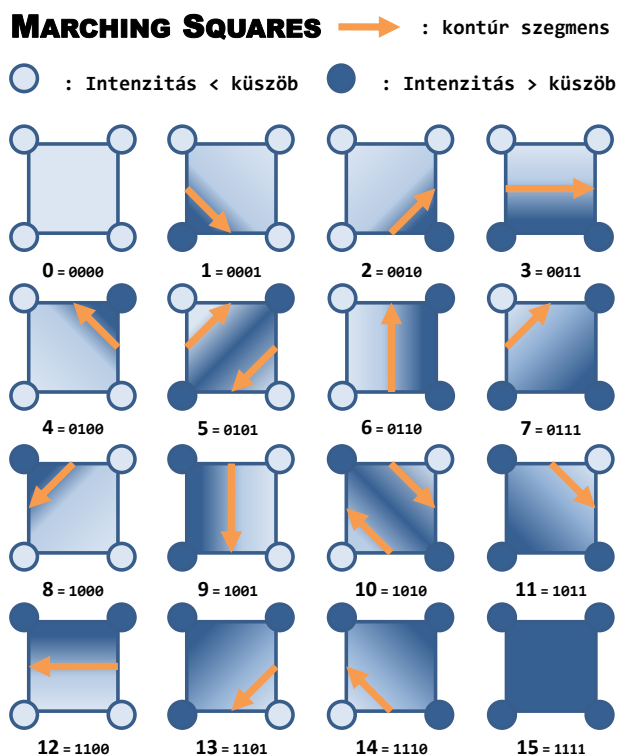
A számítógépes feldolgozás lépései

4.1. A kontúrok detektálása

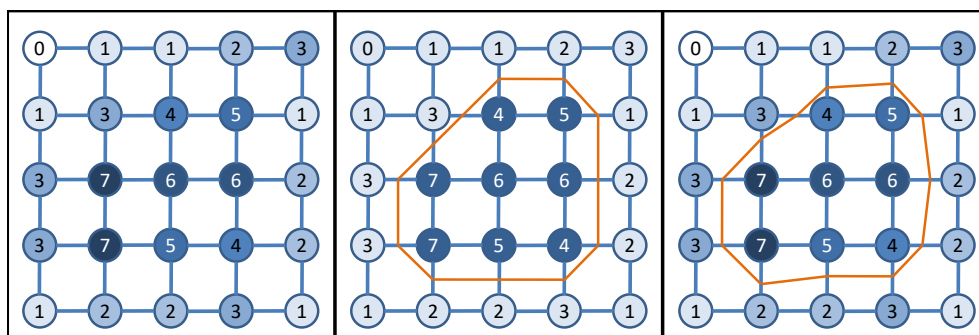
A fényképek elkészítése után következik a feldolgozás. Minden egyes képet egy-egy olyan objektumhoz rendeltem hozzá, amely a képet beolvassa, átkonvertálja azt szürkeárnyaltos képpé, amely a *Marching Squares* ([7]) algoritmus bementét fogja képezni. A Marching Squares algoritmushoz szükség van egyetlen darab egycsatornás képre, valamint egy paraméterként megadott küszöbértékre.

Az algoritmus a képet oly módon pásztázza végig, hogy mindig négy szomszédos (2×2) pixel értékét olvassa egyszerre, a 4.1 ábrának megfelelően, majd azt vizsgálja, hogy egyes intenzitás értékek a megadott küszöbérték alatt vagy felett helyezkednek el. Az osztályozás eredményeként generálható egy kód mind a 16 lehetséges esethez, amennyiben a vizsgált négyzet sarkait kettes számrendszerben négy különböző helyiértéknek feleltetjük meg.

A 16 különböző esetből topológiailag csak négy különbözik egymástól, implementáció során look-up tábla is alkalmazható. A módszer lényege, hogy a képeken úgy keresünk kontúrokat, hogy az egyes 2×2 -es cellákat vizsgálva, minden olyan élt vizsgálunk, amelyhez tartozó egyik csúcspontra igaz, hogy az ott található intenzitásérték a küszöb alatti, míg a másik a küszöb feletti. Lineáris interpoláció segítségével megkeressük az adott él mentén azt a pontot, ahol az intenzitás értéke éppen a küszöbértékkel egyezik meg. A metszéspontok kiszámításával beszűrhető egy-egy kontúrszegmens, ami a metszéspontokat köti össze. Ezt a folyamatot mutatja be a 4.2 ábra is.



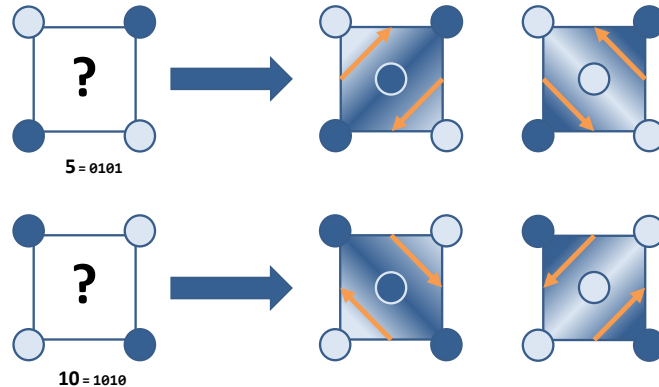
4.1. ábra. A Marching Squares algoritmus bemenetén elforduló 16 lehetséges eset. Az ábra a kontúrszegmensek irányítását is szemlélteti.



4.2. ábra. A Marching Squares algoritmus lépéseinek szemléltetése. A bemenet egy intenzitáskép, aminek minden 2×2 -es cellájához meghatározható a 16 lehetséges eset valamelyike. Csúcspontokat azokban a pozíciókban szúrunk be, ahol a lineáris interpoláció a küszöbértékkel azonos eredményt ad.

Érdekes és nem egyértelmű eseteket képeznek az ötös és tízes lehetőségek. Ilyenkor nem egyértelmű, hogy a két-két szegmens beszúrása milyen módon kell, hogy történjen, de ennek feloldására is van lehetőség. Az egyik, hogy vizsgáljuk a cella

közepére számolt intenzitásértéket (a cella sarkaiban vett értékek alapján ez a négy intenzitás átlagának fog megfelelni) és az alapján, hogy ez a küszöbérték alatt vagy felett található, oldható fel a probléma és határozható meg a konnektivitás a 4.3 ábrának megfelelően. Habár a konkrét gyakorlati alkalmazásban nem valószínű, hogy ezen két eset valamelyike is előfordulna, kezelésükre érdemes felkészülni.



4.3. ábra. A Marching Squares algoritmus ötös és tízes, nem egyértelmű eseteinek feloldása a középpontra számított intenzitás segítségével.

A Marching Squares minden esetben zárt görbét talál a képen (akár többet is), ha tudjuk azt garantálni, hogy a kép széleinél minden intenzitásérték a küszöbértéknél kisebb legyen. Ez könnyen megtehető például a képek peremezésével.

A fent ismertetett algoritmus nagy előnye, hogy jól párhuzamosítható és ezáltal a végrehajtás felgyorsítható, hiszen az egyes cellákra végzett műveletek egymástól függetlenek és nem tartalmazznak elágazást. Hátránya viszont, hogy nem egy egybefüggő görbét kapunk, azaz pontok egy adott sorrendű listáját, ahol a sorrend meghatározza az egyes szegmensek közötti konnektivitást, hanem csak önálló kontúrszegmensek sorozatát, amelyeket két végpontjuk definiál. Egy listában tárolva az éleket, redundáns lesz a végeredmény, hiszen minden pont kétszeresen lesz eltárolva. Ez a probléma egy utófeldolgozás során megoldható és a pontok kigyűjthetőek egy olyan listába, amely a fent említett tulajdonságokkal rendelkezik.

Mivel a következő fázisban előjeles távolság-transzformációt szeretnénk számítani, ezért fontos még az is, hogy az egyes pontokat milyen sorrendben illesztjük be a listába, azaz az egyes szegmenseknek milyen irányítást adunk. Ha ügyelünk arra, hogy a küszöbérték feletti értékű pixelek mindig a kontúr egy adott (mondjuk bal) oldalára essenek, akkor egy irányított görbét kapunk (jelen esetben negatív irányítá-

sút) eredményül, ahol jól meghatározható, hogy egy adott pont a görbe belsejében, vagy azon kívül található. Konkrét szeletkép estében a végeredményre mutat példát a 4.4 ábra.



4.4. ábra. A kontúr detektálása egy konkrét szeletfelvételen Marching Squares algoritmussal.

4.2. A távolság-transzformáció kiszámítása

Az előjeles távolság-transzformáció (SDT: *Signed Distance Transform*) számítása a következő lépése a feldolgozási folyamatnak. A feladat, hogy egy adott felbontású rácsháló (kép) esetében az egyes rácspontoknak (pixeleknak) a kontúrtól vett előjeles távolságát határozzuk meg. Bináris képek esetén sokszor alkalmazzák ezt a transzformációt, amire hatékony algoritmusok is születtek már. Példa erre Felzenszwalb és Huttenlocher lineáris futási idejű algoritmus [3]. Jelen esetben, mivel nem egy bináris képből indulunk ki, hanem a Marching Squares algoritmussal detektált kontúrszegmenseink vannak, így pontosabban is meghatározható az SDT. Minden pontban számítjuk az egyes szegmensektől, illetve azok végpontjaitól számított távolságok minimumát.

Felmerülő kérdés azonban, hogy mi a teendő abban az esetben, ha a SDT számításhoz használt kép felbontása nem egyezik meg a bementi kép felbontásával, amin a kontúrokat detektáltuk. Ez az eset azért fontos, mivel általában a bemeneti képek nagyobb felbontásúak, viszont nem szükséges az SDT-t ugyanekkora felbontásban megkapni, elegendő kisebb felbontással kiértékelni, ami jelentősen csökkenti a számítási költséget.

A Marching Squares algoritmus pixelkoordinátákban határozza meg a kontúr egyes pontjait. Az SDT számításánál viszont egy kisebb felbontású képre végezzük el a számításokat. Legelőször ennek a kisebb felbontású képnek a rácspontjait kell a $[0, 1]$ intervallumon keresztül áttranszformálni az eredeti kép koordinátarendszerébe, hiszen a kontúrponokat itt számítottuk. Ez egyszerűen megtehető a kiindulási kép felbontásának ismeretében.

$$x_{norm} = \frac{x_{px,1}}{width_{px,1}} \quad (4.1)$$

$$y_{norm} = \frac{y_{px,1}}{height_{px,1}} \quad (4.2)$$

$$x_{px,2} = x_{norm} * width_{px,2} \quad (4.3)$$

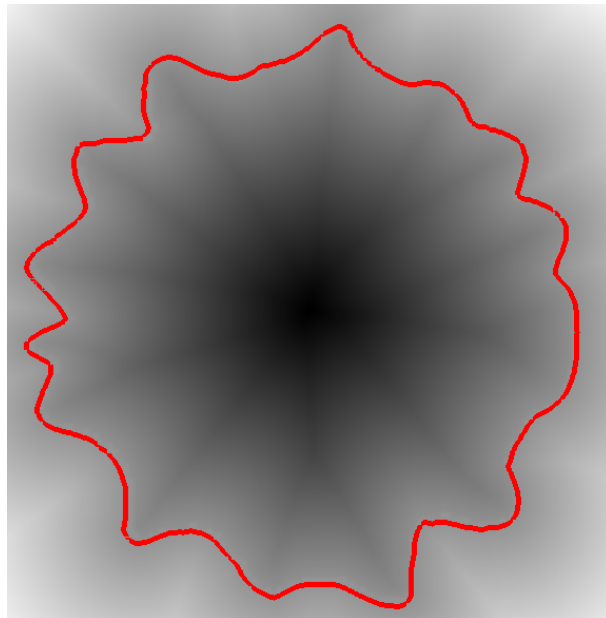
$$y_{px,2} = y_{norm} * height_{px,2} \quad (4.4)$$

Egy egyszerű és *brute force* implementáció lehet, hogy a célkép egyes pixeleit véve, számítjuk azok távolságát az egyes szakaszoktól, illetve a végpontoktól, és keressük a legkisebb abszolút értékű távolságot. Az előjelet pedig egyszerűen skaláris szorzással határozhatjuk meg, mivel az előző lépésben a görbét irányítással együtt definiáltuk. Az algoritmus lépésszáma nem túlságosan kedvező, hiszen $O(N \times m)$ -es, ahol N a rácsháló pontjainak számát, m pedig a szakaszok számát jelöli.

Ettől hatékonyabb implementáció is kidolgozható, illetve a meglévő is párhuzamosítható, hiszen itt sincs összefüggés az egyes rácspontokban végzett számítások között, de egyéb módszerekkel is gyorsítható még a jövőben. Egyik lehetőség például a kontúrponok számának csökkentése. Ez azért lehet fontos, mivel az egyes pontok maximum $\sqrt{2} * dist_{px}$ ($dist_{px}$ két szomszédos pixel közötti távolságot jelöli) távolságra helyezkednek el egymástól, ami egy meglehetősen sűrű mintavételezés.

Olyan szakaszokon, ahol görbe simább, tehát nincsenek nagyfrekvenciás komponensek, alulmintavételezés is elvégezhető. Ehhez nyújthat segítséget a görbület becslése lokálisan, aminek függvényében elhagyhatunk pontokat, ha azok egy kis görbületű szakaszon helyezkednek el.

Az SDT számításánál numerikus problémák léphetnek fel, ha a görbe közvetlen közelében végezzük a számításokat, így a kontúr zajosan rekonstruálható. Egy másik probléma lehet, hogy a görbén megfigyelhetőek a rácsháló alkalmazásából adódó artifaktumok, amik kiküszöbölésére lehet egy megoldás a görbe szűrése például Laplace-szűrővel. Ehhez azonban mindenképpen összefüggő görbereprezentáció szükséges. Konkrét szeletkép estében az SDT-re mutat példát a 4.5 ábra.



4.5. ábra. Előjeles távolság-transzformáció kiszámítása egy konkrét szeletfelvétel esetén, kiemelve a kontúrt.

4.3. A modell megjelenítése

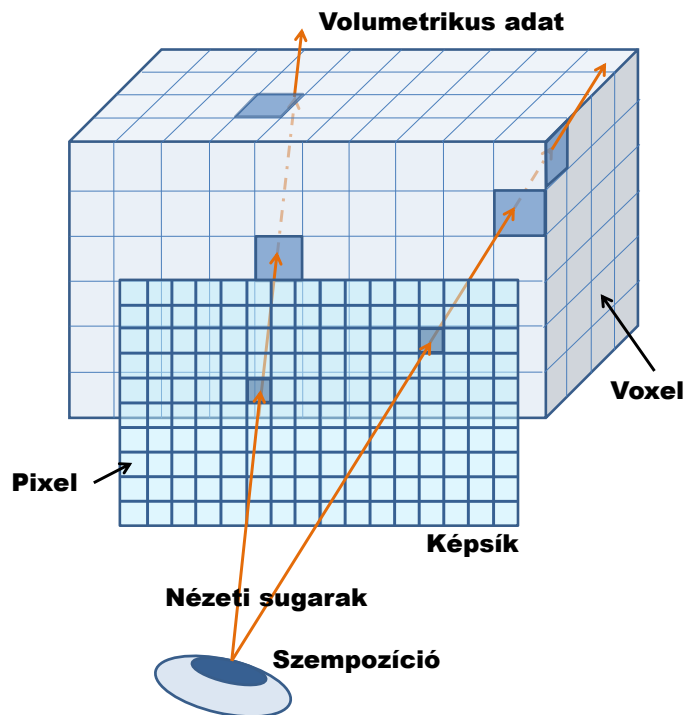
Azáltal, hogy a görbére egy SDT-t számítottunk ki, tulajdonképpen a görbe egy implicit reprezentációját határoztuk meg. Ezek alapján a görbe nem más, mint az SDT zérus értékhez tartozó pontjainak halmaza.

Ha minden szeletre meghatározzuk ezt az implicit reprezentációt, azokat egy 3D textúrába összegyűjtve kaphatunk egy volumetrikus adathalmazt. Ennek jellemzője, hogy az egyes szeletek felbontása nagyobb, mint a harmadik dimenzióban mért

felbontás, azaz a szeletek közötti távolság nagyobb, mint a szeletek egyes pixelei közötti távolság. Az így előállított implicit 3D textúra egyszerűen kapható meg a megelőző számítások elvégzése után. Ennél bonyolultabb és pontosabb reprezentációk előállítására is van példa a szakirodalomban (lsd. [11]).

Ezt követően a 3D felület megjelenítése a feladat, amihez ebben az esetben, legjobban a sugárkövetéses (*ray casting*) megközelítés alkalmazható, azon belül a *first hit ray casting* technika, hiszen minket csak a zérus értékhez tartozó szintfelület megjelenítése érdekel. A Marching Cubes algoritmus használata a modellezés egy későbbi szakaszában lehet érdekes, mikor nem csak a megjelenítés a cél, hanem egy háromszögháló generálása is fontos a további műveletekhez.

A sugárkövetés a renderelés egy képsorrendi fajtája, szemléltetése a 4.6 ábrán látható. A világkoordinátarendszerben adott egy szempozíció, valamint egy nézeti ablak, amin keresztül a virtuális világot szemléljük. A nézeti ablak egyes pontjain keresztül, amik megfelelnek a képernyőn is látott ablak egyes pixeleinek, nézeti sugarakat bocsátunk ki a szemből és e sugár mentén mintavételezzük a 3D világunkat. A mintavételezési feladat során azonban bizonyos tulajdonságok kihasználhatóak a gyorsítás érdekében.



4.6. ábra. A ray casting módszer szemléltetése.

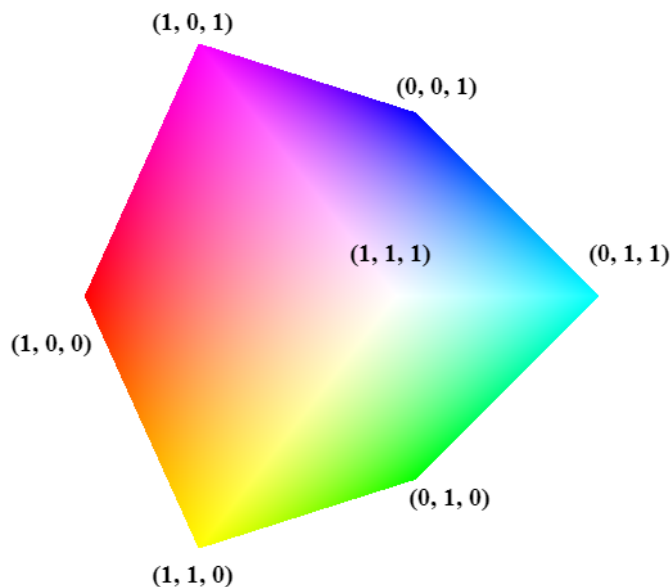
Ezek közül az egyik, hogy olyan térrészekben, amelyek nem tartalmaznak objektumot, nem kívánatos a mintavételezés (*empty space skipping*), hiszen teljesen felesleges a végeredmény szempontjából. Célszerű továbbá alkalmazni az *early ray termination* technikát is, ami szintén a felesleges mintavételektől óv meg azokban az esetekben, mikor a mintavételezés már nem járul hozzá érdemben a végeredményhez.

Ezekre kínál egy hatékony megoldást Kruger és Westermann ([5]) módszere. Ha ugyanis definiáljuk a 3D textúránknak (azaz az implicit reprezentációnknak) egy befoglaló dobozát, amelynek csúcspontjaihoz hozzá van rendelve színként az adott csúcsponthoz tartozó normalizált textúra koordináta is, akkor ezzel a befoglaló téglalattal kényelmesen alkalmazhatók a fenti gyorsítási lehetőségek. A szempozíciót és a nézeti ablakot a 3D térben elhelyezett kamera definiálja. Azt hogy a textúrát a világ mely részében kívánjuk elhelyezni, könnyen megadható homogén transzformációk segítségével, amit a kezdetben az origóban elhelyezett, befoglaló egységkockán alkalmazunk. Ezek lehetnek nyújtás, forgatás, eltolás. A sugárkövetéshez fontos információk kinyerhetők ennek a doboznak a renderelése során.

Ha a téglalattal egyes csúcsait a hozzájuk rendelt textúrákoordinátákkal, mint RGB színt komponensekkel jelenítjük meg, a grafikus hardver huzalozottan végzi el a lineáris interpolációt a színek között, ezáltal textúra koordinátákat rendelve a téglalattal lapjainak többi pontjához is. Ezek megadják a nézeti sugarak belépési pontjait a 3D textúrába, mint ahogyan azt a 4.7 ábra is szemlélteti.

Ha ezek után a téglalattal nem látható, hátsó lapjait is rendereljük (`GL_CULL_FACES` opció alkalmazásával), akkor hasonló logika alapján a sugarak textúrából való kilépési pontjait is megkapjuk a 4.8 ábrának megfelelően. Ahhoz, hogy ezeket a sugárkövetést implementáló fragmens shader programban fel lehessen használni, célszerű FrameBuffer-ekbe elvégezni a renderelést, és az ezekhez rendelt textúrát felhasználni a belépési és kilépési pontok meghatározására.

A sugárkövetéskor ezek után azt kell vizsgálnunk, hogy a sugár hossza a kezdő és végpont között egyáltalán nagyobb-e, mint zérus, hiszen ha nem, akkor ezekben a pontokban biztosan nem látunk objektumot. Ha viszont nagyobb, akkor megkezdjük a sugár mentén a mintavételezést. A sugár iránya a kilépési és belépési pontok különbségéből számítható, a mintavételezést csak a kezdő és végpontok között vé-

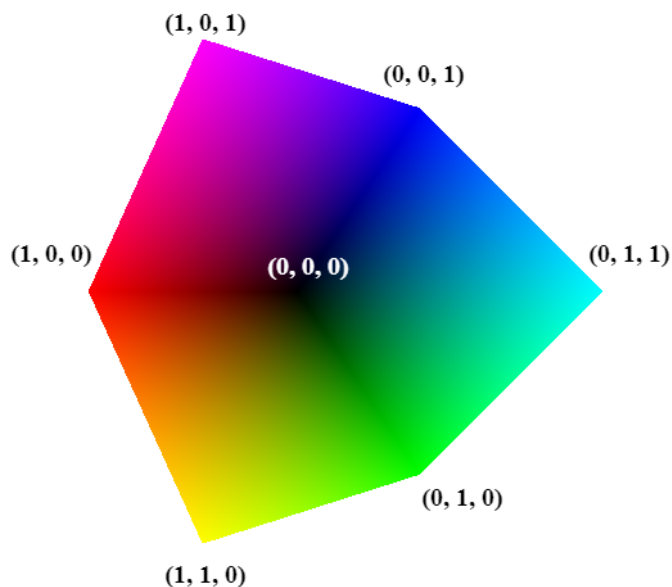


4.7. ábra. A befoglaló doboz előlapjainak megjelenítése az egyes csúcsokhoz rendelt textúra koordinátákkal (belépési pontokkal), mint színekkel. A színátmenetek a doboz lapjain a csúcsokhoz rendelt színekből lineáris interpolációval generálódtak, így rendelve textúra koordinátákat a lapok pontjaihoz is.

gezzük el ekvidisztánsan, egy előre meghatározott lépésköz segítségével az alábbi egyenletek szerint.

$$\begin{aligned} \mathbf{P}_{\text{start}} &= \text{EntryPoints}(x, y) \\ \mathbf{P}_{\text{end}} &= \text{ExitPoints}(x, y) \\ \frac{\mathbf{P}_{\text{end}} - \mathbf{P}_{\text{start}}}{\|\mathbf{P}_{\text{end}} - \mathbf{P}_{\text{start}}\|_2} &= \mathbf{e}_r \\ \mathbf{r}(t) &= \mathbf{P}_{\text{start}} + \mathbf{e}_r t \quad t \in [0, \|\mathbf{P}_{\text{end}} - \mathbf{P}_{\text{start}}\|_2] \end{aligned}$$

A fentebbiekben x, y jelöli az egyes pixelekhez rendelt 2D textúra koordinátákat, amikkel a belépési ($\mathbf{P}_{\text{start}}$) és kilépési (\mathbf{P}_{end}) pontok értékeit olvassuk ki a befoglaló doboz korábban említett renderelése után kapott textúrákból. \mathbf{e}_r az egységnyi hosszúságú irányvektort jelöli a sugár irányában. Az \mathbf{e}_r és a kezdőpont ismeretében $\mathbf{r}(t)$ a sugár egyenlete, aminek a kiértékelésével végezhető el a 3D textúra mintavételezése.



4.8. ábra. A befoglaló doboz hátulsó lapjainak megjelenítése az egyes csúcsokhoz rendelt textúra koordinátákkal (kilépési pontokkal), mint színekkel. A színátmenetek a doboz lapjain a csúcsokhoz rendelt színekből lineáris interpolációval generálódtak, így rendelve textúra koordinátákat a lapok pontjaihoz is.

Mivel jelen esetben csak a 3D felület megjelenítése az érdekes, ezért a zérus átmenet helyét keressük a sugár mentén, tehát minden lépésnél vizsgáljuk, hogy az előző és az aktuális mintavételi pontokban az intenzitás értéke ellenkező előjelű-e, mert ha igen, akkor a kettő között egyszerű lineáris interpoláció segítségével megkereshető a pontos metszéspont. A megjelenítéshez nem elegendő azonban ennyi, hiszen egyelőre csak azt tudjuk, hogy a nézeti ablak adott pontjában látszódik a felület, és azt, hogy ez a felületi pont milyen koordinátákkal rendelkezik.

Szükséges az árnyalások elvégzéséhez a felületi normális kiszámítása is, ami a normalizált gradiensvektorból számítható. Ismerve a 3D textúra felbontását, meghatározható az a lépésköz x , y és z irányban is (ez a lépésköz az egyes irányokban vett felbontás reciproka), amelyekkel pozitív és negatív irányba is ellépve, a centrális differenciák módszerét alkalmazva, számítható a gradiens. Ennek normalizálásával kapható a felületi normális, abban az esetben, ha korábban a távolság-transzformációt úgy számítottuk ki, hogy a görbe belsejében vett fel negatív értékeket, kívül pedig pozitívat. Elhelyezve egy fényforrást a térben és definiálva az anyag tulajdonságait, a felületi normális ismeretében értékelhető ki az árnyalási egyenlet.

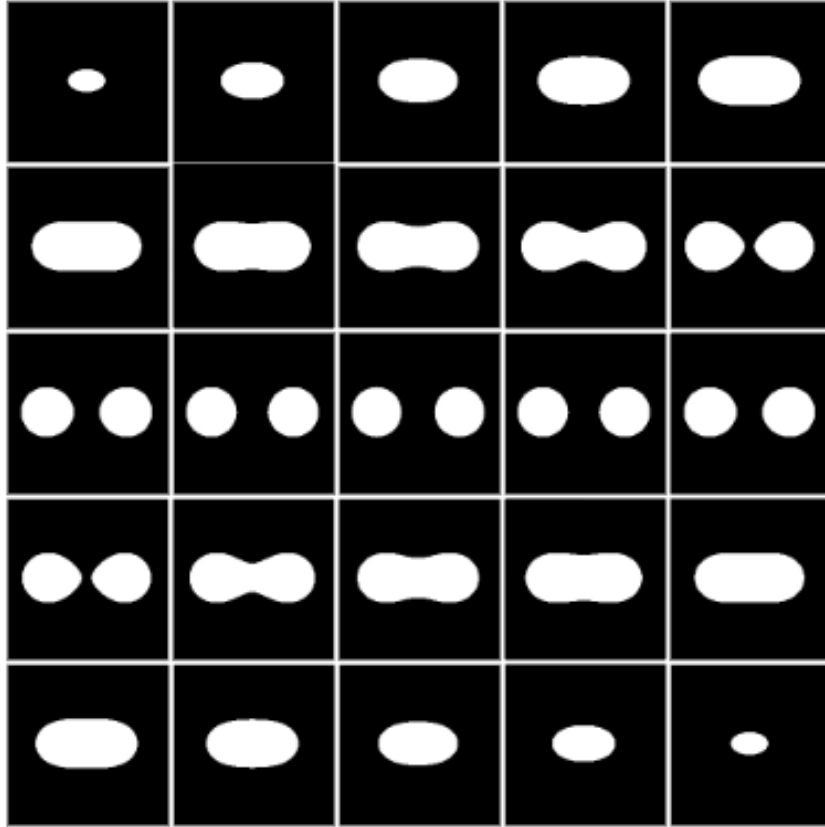
5. fejezet

A feldolgozás lépéseinek tesztelése

Az implementálás egyes fázisait külön-külön is teszteltem, mielőtt a konkrét agyag modell rekonstruálása megtörtént volna. Először is a befoglaló téglatest renderelését végeztem el. A kontúrdetektálás és a távolság-transzformáció számításának teszteléséhez Matlab környezetben generált bináris képeket használtam fel. Legegyszerűbb esetben egyetlen kör esetére vizsgáltam az algoritmus működését, ahol fehérek voltak a körlap pontjai és sötétek a háttér pontjai. A kontúr detektálását ezen teszteltem, majd ez alapján számítottam a távolság-transzformációt is.

A sugárkövetés és 3D textúra mintavételezésére gömbszimmetrikus skalármezőt definiáltam, ahol az egyes pontokban felvett értékek a középponttól vett távolságot tartalmazták. Ezen jól tesztelhető volt az izofelület megjelenítésének módja, valamint az árnyalás számítása.

Az egyes komponensek összeállása után következett egy szintén mesterségesen generált képsorozat alapján történő rekonstrukció. Ehhez a tórusz implicit egyenletét használtam fel, és generáltam a különböző szeletképeket, szintén bináris formátumban. A képek felbontása 600×600 és a szeletek száma egyenletes szelettávolsággal 50. Teszteltem azt az esetet is, amikor csak minden második szelet alapján történt a feldolgozás, illetve mikor a képek felbontása 200×200 -ra csökkent. Az eredmények az 5.1 és az 5.2 ábrákon láthatóak.



5.1. ábra. A tórusz implicit egyenletéből generált bináris képek. Az ábrán összesen 25 szeletfelvétel látható, amik a feldolgozás bemenetei voltak a tesztelési fázisban.



5.2. ábra. A tórusz renderelése bináris képekből kiindulva. A képen jól megfigyelhetők a bináris kiindulási képekből adódó artifaktumok, a felület egyenetlensége.

6. fejezet

Az eredmények bemutatása valós szobrok esetén

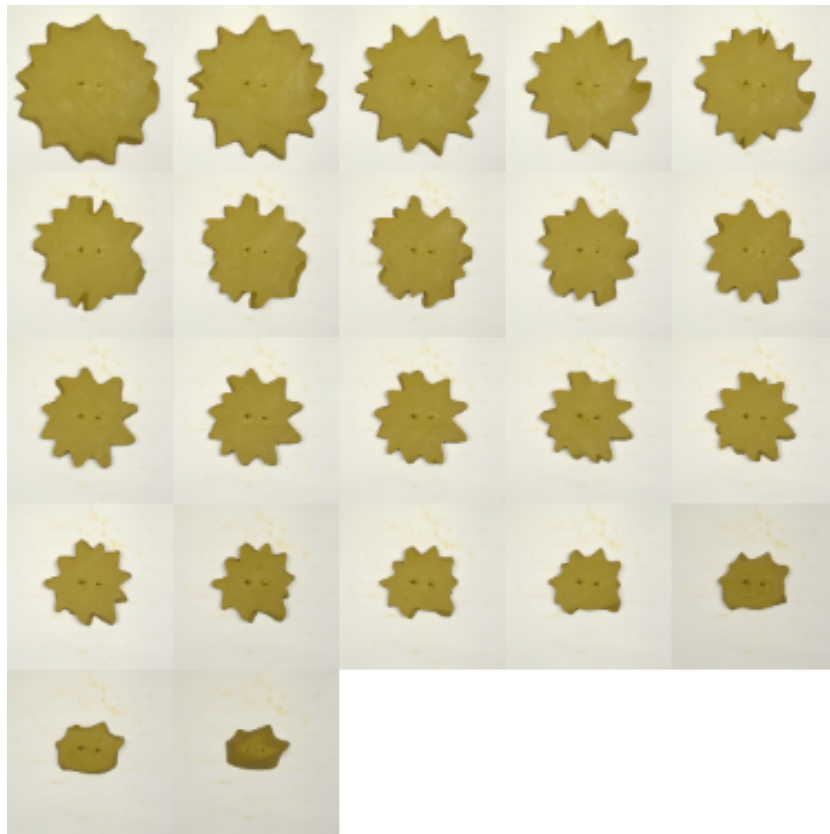
Ahogy arról korábban szó volt, a dolgozat célja az is, hogy konkrét fizikai modellek esetében megvizsgálja és bemutassa az ismertetett feldolgozás lehetőségeit és képességeit. Két agyagszobor készült erre a célra, amelyek közül az egyiket én készítettem el, a másik, lófejet ábrázoló szobrot pedig konzulensem, Dr. Csébfalvi Balázs. Ezek a szeletelés előtti állapotukban együtt a 6.1 ábrán láthatóak.



6.1. ábra. A feldolgozás tesztelésére készített két agyagszobor.

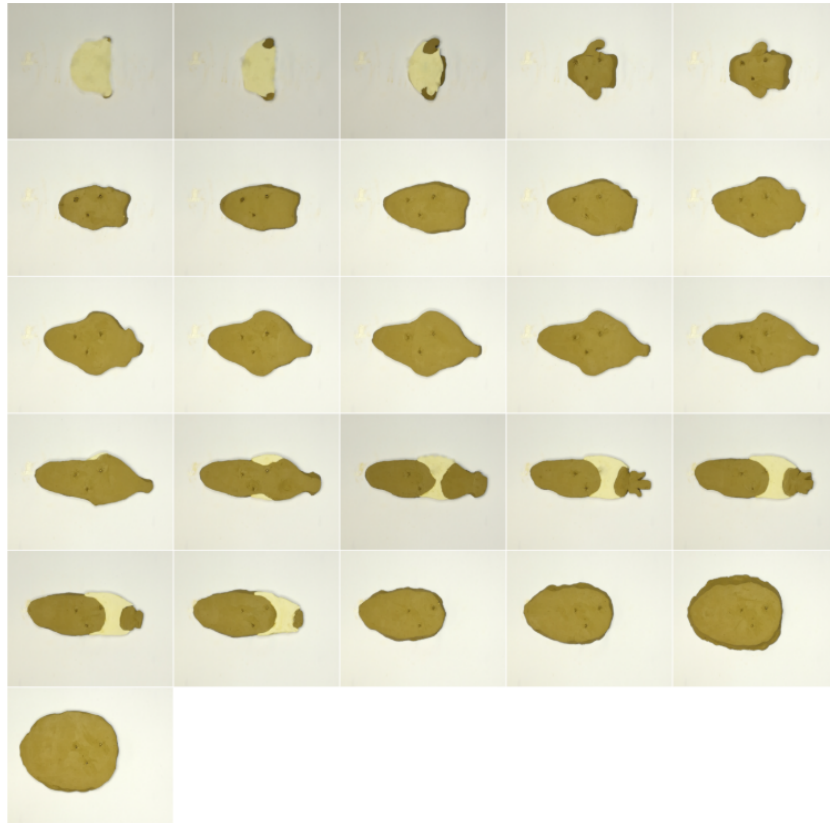
Mindkét szobrot dróttal szeleteltük fel 0.75 cm-es szeletvastagsággal. A szeleteket telefonnal fotóztam le és ezután végeztem rajtuk minimális előfeldolgozást. A

képeket ugyanazzal az ablak segítségével kivágtam, eltávolítva a felesleges részeket, majd az eredményül kapott képeket átméreteztem, hogy ezek után az algoritmus számára is könnyebben kezelhető legyen, a futási idő ne nyúljon túlságosan el. A Marching Squares algoritmus miatt egy Gauss szűrést is végeztem, hogy a néhány pixeles zavarok ne jelentsenek gondot a kontúrok detektálásánál. A feldolgozás be-
meneteként előálló szeletképeket a két szobor esetén a 6.2 és a 6.3 ábra mutatja. Első esetben 22, második esetben 26 szeletkép készült.



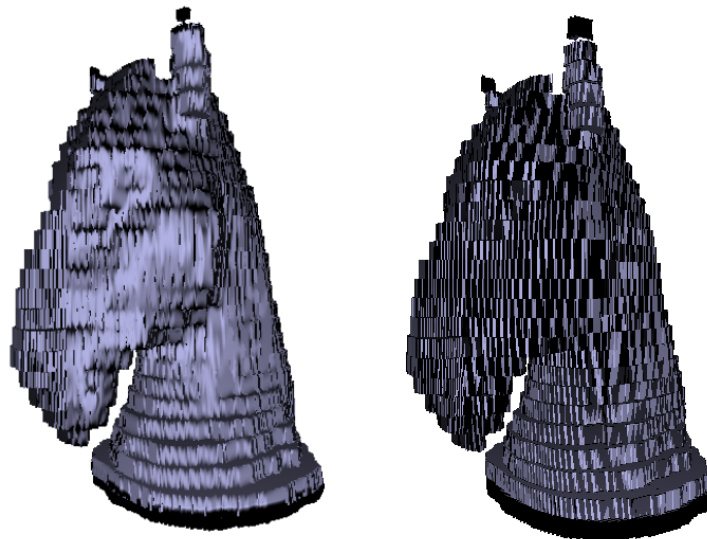
6.2. ábra. A saját készítésű szobor felszeletelésével kapott keresztmetszeti képek.

A képek alapján történő alakrekonstrukció a korábbiak alapján több módon is elvégezhető. A kapcsolódó szakirodalom ([1] [2]) áttekintésénél szó volt a bináris képek használatáról, illetve azok hátrányairól. A 6.4 ábrán látható két eset is, ami ezen a megközelítésen alapul. Az első kép azt a modellt mutatja, ami 256×256 -os bináris képek alapján készült, azokkal feltöltve egy 3D textúrát, majd megjelenítve a 0.5 értékhez tartozó szintfelületet. A második kép ugyanilyen módszerrel készült, csak a bináris képek felbontása változott 512×512 -re. Az ábra jól szemlélteti a módszer



6.3. ábra. A lófejet ábrázoló szobor felszeletelésével kapott keresztmetszeti képek.

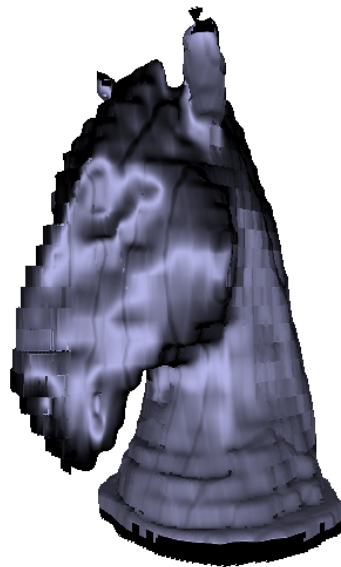
legfőbb hátrányát, a lépcsőzetes felületet, ami a bináris képek véges felbontásából adódik, és attól erősen függ.



6.4. ábra. A megjelenített modell 256×256 -os és 512×512 -es felbontású bináris képek esetén

A 6.5 ábrán egy másik módszer alkalmazásának eredménye látható. Ebben az esetben a szürkeárnyalatossá konvertált képek közvetlenül lettek a 3D textúrába töltve, mindenféle küszöbözés, kontúrdetektálás és reprezentációváltás nélkül. A sugárkövetésnél beállított küszöbérték, amelyhez tartozó szintfelületet megjelenítettem, az a küszöbérték volt, aminek a segítségével korábban a kontúrokat detektálta a program, illetve ami alapján előállította az előző bekezdésben említett bináris képeket.

A felület ebben az esetben már simább, köszönhetően a trilineáris interpolációnak, amit a grafikus hardver végzett el a textúra mintavételezése során, de még mindig jól látható rajta a lépcsőzetesség. Fontos azt is megjegyezni, hogy ebben az esetben a küszöbérték változtatásán túl, vagy a képek előfeldolgozásán kívül nincs túl sok lehetőség arra, hogy szebb felületet kapjunk ebben a fázisban.



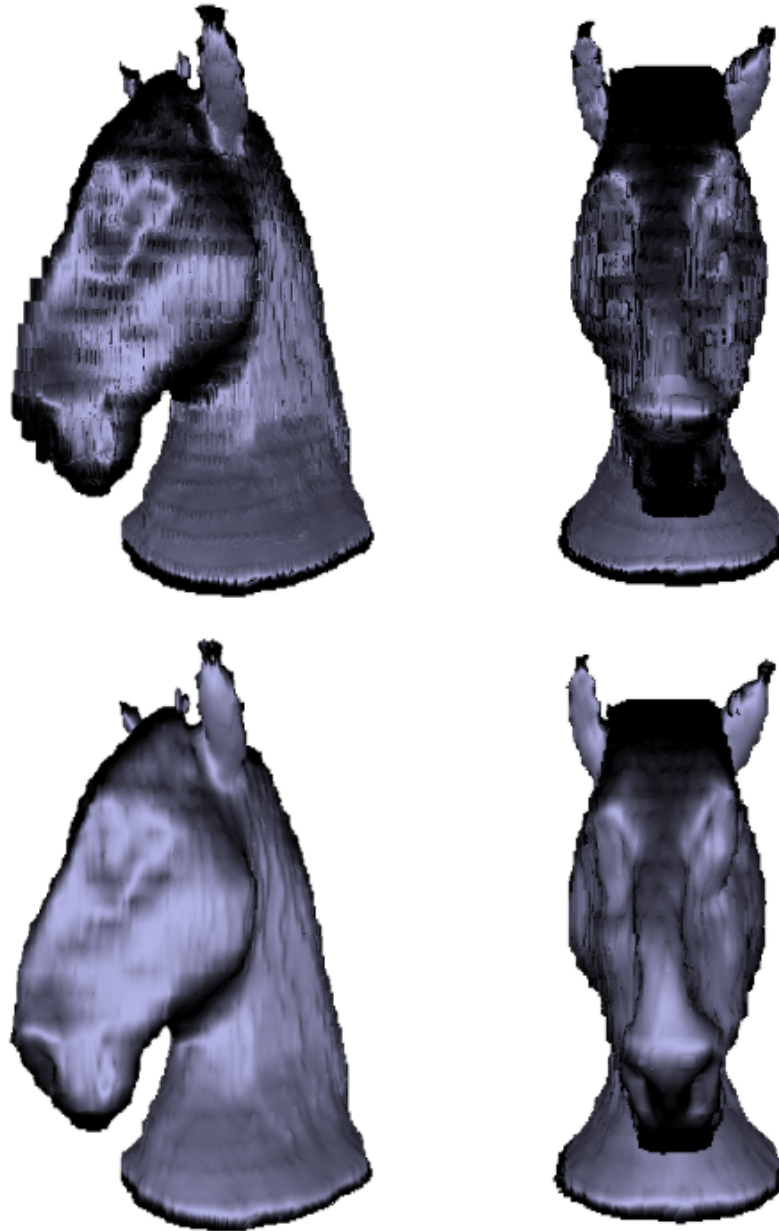
6.5. ábra. A szürkeárnyalatos képek egyszerű textúrába másolásával kapott eredmény.

A dolgozatban bemutatott módszer segítségével több esetet is kipróbáltam. Kísérleteztem azzal, hogy a távolság-transzformáció és a bemeneti szeletképek felbontásának kapcsolata milyen hatással van a kimenetre. A fő kérdés az volt, hogy milyen felbontásbeli arány megválasztása a legjobb a bemeneti képek és a távolság-transzformáció között. Néhány eredmény a 6.6 és a 6.7 ábrákon látható a lófej, illetve a 6.8 ábrán a másik szobor esetén. Ha a távolság-transzformáció felbontása a kiindulási képével közel azonos (256×256), akkor a felületen egyenetlenségek figyelhetők

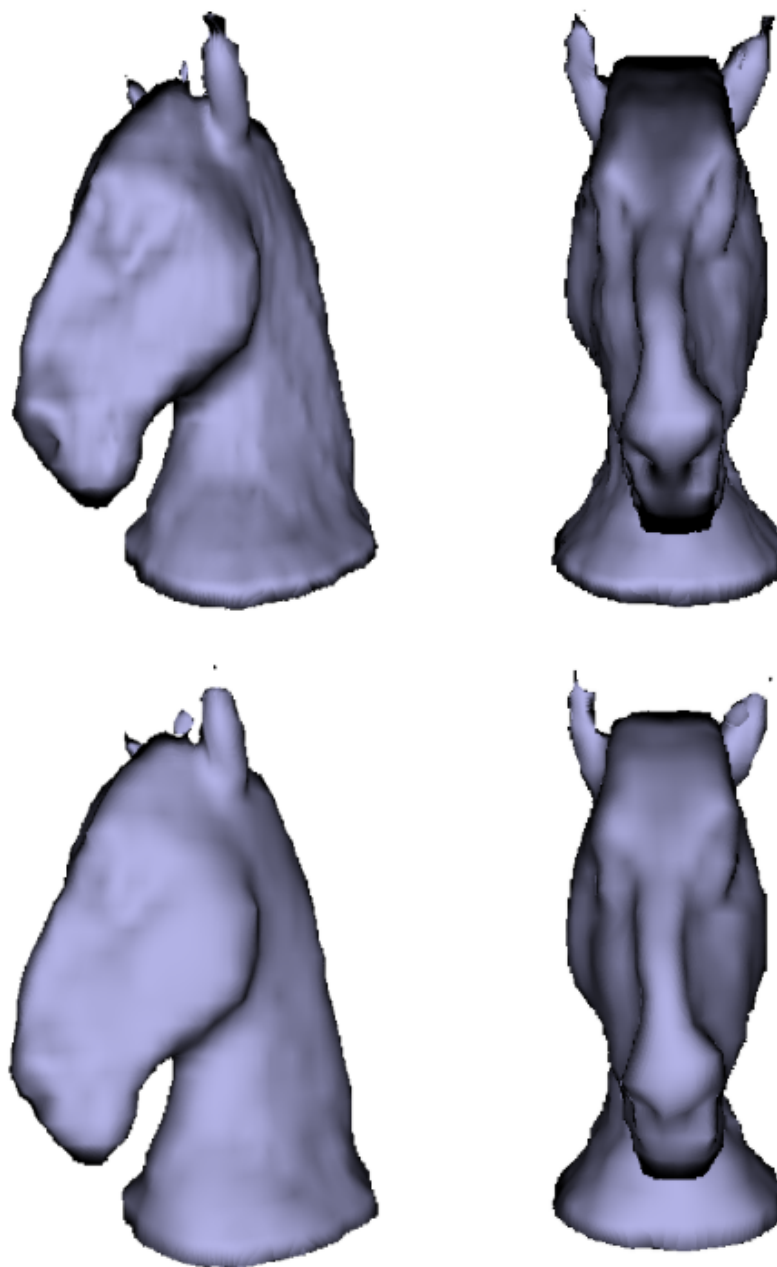
meg, amik azonban egyre inkább eltűnnek a DT felbontásának csökkentésével. Egyre simább felületet kapunk, ami azonban előbb-utóbb a részletek eltűnéséhez vezet, amely tendencia látható az alábbi képeken is. 32×32 -es felbontás esetén az eredmény egyre inkább elmosódott, míg 64×64 -es felbontás mellett már nem láthatóak zajok a felületen, mégis jól kivehetőek még a részletek, mint például az orrlukák.

A tapasztalat alapján egy négyszeres felbontásváltozás megfelelőnek bizonyult, azaz amíg a képek felbontása 256×256 körüli, addig a távolság-transzformáció számítását elegendő volt elvégezni 64×64 -es felbontású képre. Ez azért vezet pontosabb eredményekhez, mert így egyetlen pontban pontosabban meghatározható a görbétől vett távolság, aminek a szakaszossága egyre kevésbé van hatással a számított távolságmezőre.

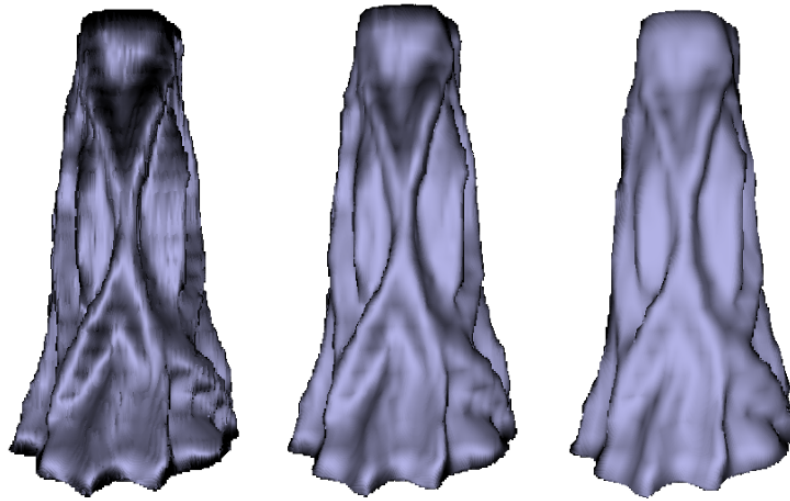
A másik szobor esetében a levonható következtetés hasonló. A szobor aljánál kismértékben észrevehető a fényképezésnél elkövetett pontatlanság, a szeletek bizonyos mértékű elcsúszása, elcsavarodása. Ez a bonyolultabb felület miatt is nehezebben észrevehető volt fényképezés közben, mint a lófej esetében, ahol támpontként szolgált a szeletek hossz tengelyének a iránya.



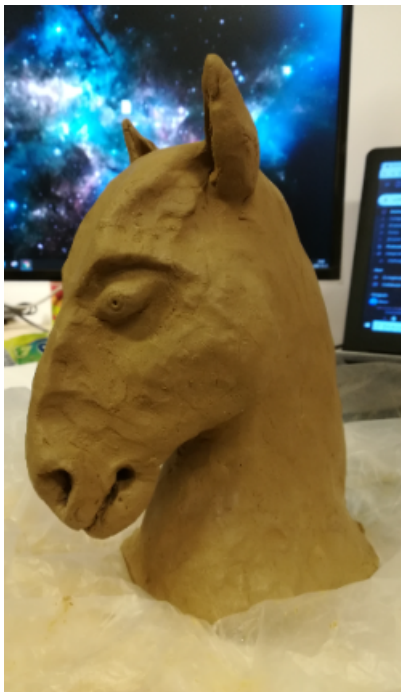
6.6. ábra. A lófejről készült szobor modellje oldalról és szemből, 256×256 -os, illetve 128×128 -as felbontású távolság-transzformáció esetén.



6.7. ábra. A lófejről készült szobor modellje oldalról és szemből, 64×64 -es, illetve 32×32 -es felbontású távolságtaszformáció esetén.



6.8. ábra. A másik szobor modellje, 128×128 -as, 64×64 -es és 32×32 -es felbontású távolság-transzformáció esetén.



6.9. ábra. A lófejről készült szobor agyagból mintázva, mellette a digitális modelljével.



6.10. ábra. A másik szobor agyagból mintázva, mellette a digitális modelljével.

7. fejezet

Összefoglalás és jövőbeli célok

Ahogy már a bevezetőben is említettem egy kezdeti modell kialakítása volt a cél, amire támaszkodva felgyorsítható a modellezési feladat, és aminek első része nem igényli semmilyen 3D szerkesztő program használatát és ismeretét. A dolgozatban vázolt módszer pontosan ezt teszi lehetővé.

A bemutatott számítógépes feldolgozáshoz közismert algoritmusokat implementáltam, hogy ezekkel egy olcsó és könnyen hozzáférhető modellezési eljárást dolgozzak ki, amelynek első, feldolgozást megelőző szakasza saját, egyedi ötleten alapul. Az eddigi eredmények a módszer lehetőségeit kívánták bemutatni. Az eredmények ígéretesek, érdemes tehát a továbbfejlesztés lehetőségeivel is foglalkozni.

Egy jövőbeli cél lehet a modell megjelenítésén túl, az eddigiekre építve, a későbbi szerkeszthetőséget is biztosító háromszög mesh generálása, amely az implicit függvénnyel történő modellezéshez képest egy tömörebb reprezentációnak tekinthető.

Irodalomjegyzék

- [1] Adrian Bors–Lefteris Kechagias–Ioannis Pitas: Binary morphological shape-based interpolation applied to 3d tooth reconstruction. *IEEE Transactions on Medical Imaging*, 21. évf. (2002. 03), 100–109. p.
- [2] Roberto de Alencar Lotufo–Alexandre Xavier Falcao: Shape-based interpolation methods applied to medical imaging. *Anais do SIBGRAPI VI*, 1993. október., 323–331. p. 9 p.
- [3] Pedro F. Felzenszwalb–Daniel P. Huttenlocher: Distance transforms of sampled functions. *Theory of Computing*, 8. évf. (2012) 19. sz., 415–428. p.
URL <http://www.theoryofcomputing.org/articles/v008a019>.
- [4] Rao Garimella–Rao Garimella: Title: A simple introduction to moving least squares and local regression estimation intended for: A simple introduction to moving least squares and local regression estimation, 2017. 06.
- [5] J. Kruger–R. Westermann: Acceleration techniques for gpu-based volume rendering. In *IEEE Visualization, 2003. VIS 2003*. (konferenciaanyag). 2003, 287–292. p.
- [6] J.P. Lewis–Frédéric Pighin–Ken Anjyo: Scattered data interpolation for computer graphics. *ACM SIGGRAPH 2014 Courses, SIGGRAPH 2014*, 2010. 01.
- [7] Marching cubes and variants. http://web.cse.ohio-state.edu/~wenger.4/publications/isosurface_book_preview.pdf, 2020.
- [8] David Meyers–Shelley Skinner–Kenneth Sloan: Surfaces from contours. *ACM Trans. Graph.*, 11. évf. (1992. július) 3. sz., 228–258. p. ISSN 0730-0301. URL <https://doi.org/10.1145/130881.131213>. 31 p.

- [9] Donald Shepard: A two-dimensional interpolation function for irregularly-spaced data. ACM '68 konferenciasorozat. New York, NY, USA, 1968, Association for Computing Machinery, 517–524. p. ISBN 9781450374866. URL <https://doi.org/10.1145/800186.810616>. 8 p.
- [10] Vaclav Skala: A practical use of radial basis functions interpolation and approximation. 37. évf. (2016. 01), 137–145. p.
- [11] Greg Turk–James F. O’Brien: Shape transformation using variational implicit functions. In *Proceedings of ACM SIGGRAPH 1999* (konferenciaanyag). 1999. augusztus, 335–342. p.
URL <http://graphics.cs.berkeley.edu/papers/Turk-STU-1999-08/>.