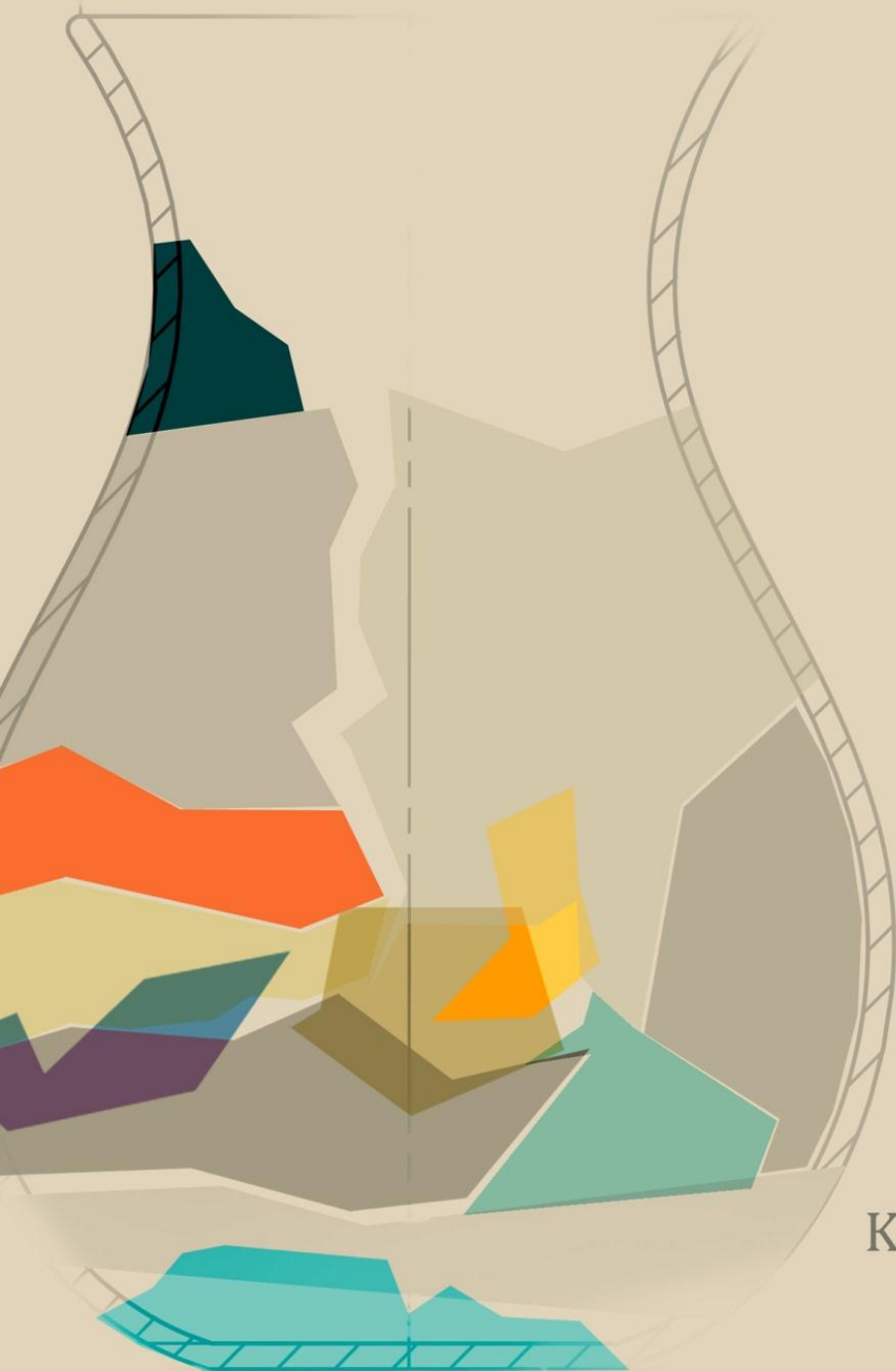


Virtuális műtárgy rekonstrukció

TDK 2012/13



Készítették:
Majoros Melinda
Varga Eszter

Konzulensek: Piros Attila
Hajdú Gyula

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.



BME GÉP- ÉS TERMÉKTERVEZÉS TANSZÉK
BME DEPARTMENT OF MACHINE AND PRODUCT DESIGN



Köszönetnyilvánítás

Szeretnénk köszönetet mondani konzulensünknek, Piros Attilának, a biztatásért, a módszer kidolgozását támogató javaslataiért és azért, hogy végig segítette munkánkat az eljárás kidolgozásában. Külön köszönjük, hogy biztosította a C3D kft. támogatását a módszer kivitelezése során.

Külön köszönjük Hajdú Gyulának a sok hasznos tanácsot a műtermi fényképek elkészítésénél és azt, hogy megfelelő műtermet biztosított számunkra. A fotórealisztikus vizualizáció elkészítését előkészítő eljárások kidolgozásában tett javaslatai is sokat segítettek munkánk során.

Továbbá szeretnénk megköszönni Körtélyesi Gábornak a 3D szkennelési eljárás lebonyolítása során tett segítségét és azt, hogy kapcsolatot biztosított az Óbudai Egyetemmel.

Itt szeretnénk kiemelni Horváth Richárdot, az Óbudai Egyetem munkatársát, aki a 3D szkennelési eljárást elvégezte nekünk, és gyakorlatban is megmutatta és elmagyarázta számunkra az eljárás működését.

Az eljárás során használt forgásszimmetrikus műtárgyat köszönjük az ELTE BTK Régészeti Intézet, Magyar Középkori és Kora Újkori Régészeti Tanszéknek, és Herbst Annának, aki a kapcsolatot biztosította, és tanárának Mordovin Maximnak, aki a megfelelő leletet rendelkezésünkre bocsájtotta.

Végül szeretnénk köszönetet mondani a családunknak, a tanárainknak, a hallgatótársainknak és a barátainknak, akik támogatásukkal és biztatásukkal segítették tudományos munkánk elvégzését.

Tartalomjegyzék

JELÖLÉSJEGYZÉK	5
RÖVIDÍTÉSEK ÉS SZAKKIFEJEZÉSEK JEGYZÉKE	7
BEVEZETÉS	9
1. A SZAKIRODALOM ÁTTEKINTÉSE	10
2. A KUTATÁS CÉLKITŰZÉSEI	12
3. A MÓDSZER BEMUTATÁSA	13
3.1 A MÓDSZER ELŐKÉSZÍTŐ ELJÁRÁSOK	15
3.1.1. 3D körszkennelési eljárás	15
3.1.2. A műtárgy fotózása műtermi környezetben	15
3.2. FOTÓREALISZTIKUS TEXTÚRA KÉSZÍTÉS	16
3.2.1. Az elkészült fényképek adott méretre vágása	18
3.2.2. A fényképek torzítása előzetes számítások alapján	19
3.2.3. A textúra elkészítése	19
3.3. FELÜLETILLESZTÉS MATLAB PROGRAMBAN	20
3.3.2. Koordináta transzformációk	20
3.4. A 3D-S TESTMODELL LÉTREHOZÁSA PRO/ENGINEER-BEN	30
3.4.1. A pontsorok behívása, a hengeres felület létrehozása	30
3.4.2. A textúra illesztése	31
3.4.3. A módosított felület, majd a testmodell elkészítése	31
3.5. FOTOREALISZTIKUS MEGJELENÍTÉS A PRO/ENGINEER PHOTOLUX MODULJÁBAN	32
4. A MÓDSZER ALKALMAZÁSA A KIVÁLASZTOTT MŰTÁRGYON	33
4.1. A TECHNOLÓGIÁT ELŐKÉSZÍTŐ ELJÁRÁSOK	33
4.1.1. A műtárgy szkennelése	33
4.1.2. A műtárgy fotózása műtermi környezetben	34
4.2. FOTÓREALISZTIKUS TEXTÚRA KÉSZÍTÉS	36
4.2.1. Az elkészült fényképek adott méretre vágása	36
4.2.2. A fényképek torzítása előzetes számítások alapján	38
4.2.3. A textúra elkészítése	40
4.3. A MATLAB PROGRAMBAN A PONTSOROK GENERÁLÁSA	44
4.4. A MŰTÁRGY 3D-S REKONSTRUKCIÓJA A PRO/ENGINEER-BEN	45
4.5. A MŰTÁRGY VIZUALIZÁCIÓJA	49
ÖSSZEFOGLALÁS	53
A KUTATÁS EREDMÉNYEI	53
AZ EREDMÉNYEK HASZNOSÍTÁSA	54
TOVÁBBI CÉLKITŰZÉSEK	55
ÁBRAJEGYZÉK	56
IRODALOMJEGYZÉK	58
MELLÉKLET	59
A MŰTÁRGY ADATAI	59
ÁSATÁS JELENTÉSE	59
3D SZKENNER ADATAI	61

MŰTEREM ADATAI	62
ALKALMAZOTT SCRIPTEK.....	64
TORZÍTÁSI TÁBLÁZAT	66
TEXTÚRA VÁLTOZATOK.....	67
MATLAB PROGRAMKÓD	69

Jelölésjegyzék

gDens - a pontok egymástól mért függőleges és vízszintes távolsága

g - a szintek száma, *gDens* -ből és a tárgy paramétereiből számolja a program

zLevel - *z* koordináta értéke az adott szinten

zMin - a legkisebb *z* szint, az első szint

zMax - a legmagasabb *z* szint, utolsó szint, *zMin* és *zMax* között helyezkednek el egyenletesen elosztva a szintek.

grid - pontsorok

homogenCoords - a szkennelt pontokhoz hozzáadott homogén koordináta

Points - a behívott szkennelt pontok

idxMax - az adott *zLevel* plusz a *gDens* fele(távolságok)

idxMin - az adott *zLevel* mínusz a *gDens* fele(távolságok)

nCS - a megfigyelő koordináta rendszerek változója

theta - a megfigyelő koordináta rendszer forgatási szöge

dist - a megfigyelő koordináta rendszer távolsága az eredeti koordináta rendszer origójától

CoordTrans1 - a megfigyelő koordináta rendszerből az eredeti koordináta rendszerbe transzformáló mátrix

CoordRot - az eredeti koordináta rendszerből a megfigyelő koordináta rendszerbe transzformáló mátrix forgató mátrixa

CoordMove - az eredeti koordináta rendszerből a megfigyelő koordináta rendszerbe transzformáló mátrix eltolási mátrixa

CoordTrans2 - az eredeti koordináta rendszerből a megfigyelő koordináta rendszerbe transzformáló mátrix

P2 - a transzformált szkennelt pontok a megfigyelő koordináta rendszerbe

P2Ysorted - az összes szkennelt pont „y” koordináta szerint sorba rendezve

P2Yselected - az adott *z* szinten kiválasztott „y” koordinátájú pontok

P2Xsorted - a kiválasztott „y” koordinátájú pontok sorba rendezve „x” koordináta szerint

idxP2Xsorted-a legnagyobb „x” koordinátájú kiválasztott pontok

P2XYselected-az „x” és az „y” koordináta alapján kiválasztott pont

P2near-a legközebb eső kiválasztott pont az adott megfigyelő koordináta rendszer x tengelyéhez

P2selected-a kiválasztott pont visszatranszformálva az eredeti koordináta rendszerbe

grid_Rz- jobb oldali szabályos generált pontsorok fájljainak neve

grid_Lz- bal oldali szabályos generált pontsorok fájljainak neve

mod_Rz- jobb oldali módosított pontsorok fájljainak neve

mod_Lz- bal oldali módosított pontsorok fájljainak neve

Rövidítések és szakkifejezések jegyzéke

NURBS felületek-(*Non-Uniform Rational Basis Splines*) nem egyenletes osztású B-szplájn felület

.pts- Pro/ENGINEER-ben a pont koordinátákat tartalmazó fájlok kiterjesztése

CAD rendszer-számítógéppel támogatott tervező rendszerek

Pro/ENGINEER Wildfire 5-3D-s parametrikus gépészeti tervező rendszer

MATLAB- egy speciális programrendszer, amely numerikus számítások elvégzésére lett kifejlesztve és emellett egy programozási nyelv is

triangulizált felület-háromszög alakú részekből álló poligon felület

JPG - (Joint Photographic Group) - Fájlformátum, amely veszteséges tömörítéssel dolgozik. Egy olyan formátumszabvány, amelyet állókép-adatok tömörítésére használnak. A maga 16,7 millió színével különösen alkalmas színes képek tömörítésére.

TIF- Elsősorban a nyomdaiparban használatos veszteségmentes képformátum, számos képszerkesztő program használja. A digitális fényképezőgépeknél a felső kategóriás, illetve professzionális készülékekben találkozunk ezzel a formátummal.

PNG - (Portable Network Graphics formátum) A Graphics Interchange Format (GIF) képformátum hiányosságainak (például korlátozott számú szín) kiküszöbölésére létrehozott, veszteség nélküli formátum.

RAW- Nyers képformátum. A RAW állományok feldolgozás nélkül tartalmazzák a szenorból kiolvasott képi információt. Ezért ez egy nagy információtartalmú formátum.

GIMP – (GNU Image Manipulation) grafikai és fotófeldolgozó szabad felhasználású szoftver

RawTherapee – RAW formátumok kezelésére alkalmas szabad felhasználású szoftver

Microsoft ICE – panorámakép készítésére alkalmas szabad felhasználású szoftver

RGB - olyan additív színmodell, ami a vörös, zöld és kék fény különböző mértékű keverésével határozza meg a különböző színeket.

szenzor zaj – a fényképezőgépek érzékelőjén különböző okok (pl. hőmérsékletváltozás) miatt keletkező zavar megjelenése a digitális képen. Jellemzően a képtől eltérő színű pixelek formájában jelenik meg.

színhőmérséklet - A színhőmérséklet a látható fény egy jellegzetessége. Egy fényforrás színhőmérsékletét az általa okozott színérzet és egy hipotetikus feketetest-sugárzó által létrehozott színérzet alapján határozzák meg.

nem pixelpontos fénymérés – átlagoláson alapuló fénymérési módszer

Anti-Aliasing szűrő - Az alias egy mellékterméke a digitális rendszerek működésének, és akkor jelentkezik, amikor a jel olyan hullámokat tartalmaz, melyek frekvenciája nagyobb mint a mintavételi frekvencia fele. A szűrő, ennek az effektusnak a korrigálását végzi.

MOIRE jelenség - Különböző hajlásszögű, egymást fedő, szimmetrikus vonal- vagy pontrácsok által okozott nem kívánatos mintázat.

script - kódrészlet

render – számítógépes digitális képalkotás, végeredménye egy felhasználható digitális kép

exponálás - megvilágítás

pixel - A digitális képfeldolgozásban a képpont (angolul pixel) egy pont egy rastergrafikus képen.

soft box - Fotográfiai világítóberendezés, lágy fény előállítására képes.

Bevezetés

A következőkben egy olyan módszert szeretnénk bemutatni, ami a régészeti és a gépészeti modellezés határterületének összekapcsolásával keletkezett. Az eljárás során egy létező műtárgy visszamodellezésére kerül sor. A módszer egy komplex folyamat, ami egy, az eredeti modellel jó közelítéssel megegyező virtuális reprodukciót biztosít számunkra. Az elkészült számítógépes testmodell alkalmas lesz számítások elvégzésére és az esetlegesen hiányzó részek rekonstruálására, ezen felül realisztikus vizualizációt biztosít a felhasználók számára.

Az eljárás számos új lehetőségre mutat rá a restaurátori munkálatok elvégzésében, biztosítja a műtárgy reprodukciójának a valósághű bemutatását, ami akár képességeikben korlátozott felhasználók számára is lehetővé teszi az adott műtárgy megismerését. A módszer kiterjeszthető a gépészeti tervezés során való felhasználásra is. Az folyamat az említetteken kívül számos más gyakorlati felhasználási lehetőséggel rendelkezik, amit a dokumentációnk során említeni fogunk.

A következőkben bemutatásra kerül a módszer néhány egyszerűsítéssel elvégzett leírása. Eredményül egy realisztikus reprodukciót kapunk, ami komplexen magába foglalja a pontos geometriát és a valósághű megjelenítést is.

1. A szakirodalom áttekintése

A kutatásunk első lépése a szakirodalom megismerése volt, mivel ezen ismeretek nélkül nem lehetne érdemi munkát készíteni, A módszer kidolgozása előtt lényeges, hogy lássuk mások milyen eredményeket értek el eddig ezen a területen.

A szakirodalom kutatásunk kiterjedt mind az írott forrásokra, mind pedig tudományos folyóiratokban publikált, a témánkkal kapcsolatos művek megismerésére is.

Ilyen integrált módszerrel viszont nem találkozunk, mely egy választott CAD rendszerben, esetünkben a Pro/ENGINEER-ben, komplexen kezelné a test valósághű rekonstruálását és vizualizációját. Így kutatásunkat két területre bontottuk, az első felében a szkennelés utáni pontos testmodell létrehozásával, a második részben pedig a fotorealistikus megjelenítéssel kapcsolatos néhány eredményt vettük számba.

Testmodell létrehozása 3D szkennelés után

A kiválasztott tárgy szkennelése után az eredmény egy pontfelhő. Számos módszer létezik arra, hogy a pontfelhőből felületmodellt hozzunk létre, ezek közül az egyik legelterjedtebbek a különböző *Reverse Engineering* programok, melyek a pontfelhőre *triangulizált felület* illesztenek, majd egy másik program(*CAD rendszer*) segítségével a *triangulizált felület*ből testmodell készíthető. A *triangulizált felületek* poligonokból állnak, így a szkennelt test görbületeinek a követése csak nehézségek árán oldható meg. Ezen eljárások hátránya, hogy a Reverse Engineering programok nem képesek a felülettel együtt a textúrát is kezelni.

A 3D szkennelés fejlődésével egyre több funkcióval rendelkező gépeket hoznak létre a szkennelési feladatok ellátására. A legújabb szkennerek már a szkennelés után létre tudnak hozni *NURBS felületeket*, melyekből később test modell készíthető. Ez az eljárás igen hasznos, viszont a vizuális megjelenítéssel ez sem foglalkozik.

A tudományos folyóiratokban való kutatásunk során találtunk a témánkhoz kapcsolódó cikkeket. Az alapprobléma, mellyel a cikkel nagy része foglalkozik, a felület rekonstruálás. A felületek rekonstruálására szükség van hiányos felületek pótlása esetén, vagy ha a valós felület virtuális modelljére van szükségünk.

A szabadformájú felületek rekonstruálása, matematikai algoritmusokkal való közelítése még napjainkban is egy érdekes terület, mellyel számos tudományos munka foglalkozik.[1] Az emberi test digitalizálása is ehhez a területhez tartik. Volt olyan tanulmány is, mely a *triangulizált felület* optimalizálásával igyekezett javítani a felület pontosságát, felületi normálisok segítségével [2].

A felület rekonstruálásra írtak már algoritmust, mely felület rekonstrukcióval foglalkozik rendezetlen ponthalmaz esetén. A felület topológiai változtatásával közelítik a pontokat, de ezen eljárás során is a *triangulizált felület* volt a kiindulási pont [3].

Az egyik tudományos munka a felület rekonstruálást a szkennelési utak segítségével végezte, melyek a szkennelés után rendelkezésünkre állnak [4].

Textúra készítés

A textúra alapját a pontos fényképezés során rögzített digitális negatívok jelentik. Ezek a képek tartalmazzák a legtöbb valós információt az eredeti beállításról, de ezek sem teljesen veszteségmentesek. Amikor digitális negatívból képet generálunk óhatatlanul még több információ vesz el. Ez a folyamat torzulást eredményez a textúra készítés közben felhasznált képeken [5]. Ennek a problémának a megoldására mi is a fényképészetben használt képkorrekciókat végeztük, amivel számos helyen találkozhatunk a szakirodalom áttekintése során.

A kutatás során kiderült, hogy korábban már foglalkoztak íves vagy gömb alakú felületek visszatorzításával (térképészetben). Egy, az MIT-n elvégzett eljárás, földgömb felületi képének visszatorzításával foglalkozik, azonban itt a fotózás normális irányban történik, és minden fényképen azonos torzítási műveletet hajtottak végre. Ez egy speciális eset, ami csak gömbre alkalmazható [6].

Az irodalom áttekintése során a leírások nem térnek ki szabálytalan és egyedi textúrák automatizált elkészítésére. Általában folytonos és szabályos mintázatú textúrákat készítenek, illesztenek és használnak.

A szakirodalom áttekintése után elmondható, hogy rendelkezésünkre állnak már módszerek a virtuális testmodell rekonstruálására, de ezek az eljárások általában csak a felület rekonstrukcióval foglalkoznak részletesen. A textúra készítés egyedi eseteivel és automatizálásával kapcsolatban kevés tudományos eredmény található. Megállapítható, hogy a tudományos eredmények szabályos geometriai transzformációkon alapuló általános esetekkel foglalkoztak eddig.

2. A kutatás célkitűzései

Célunk egy integrált módszer kidolgozása, amely segítségével a műtárgyról egy virtuális modell készül, ami geometriájában és megjelenítésében is pontosan reprezentálja az eredeti tárgyat. A dolgozat egy általunk választott 16. századi műtárgy rekonstrukciójával foglalkozik. A téma újszerűségét az általunk fejlesztett eljárás jelenti, amely a geometria rekonstrukciója mellett párhuzamosan a vizuális megjelenést is helyreállítja. Ez a módszer a gyakorlatban számos helyen alkalmazható és megkönnyíti a restaurálási folyamatokat.

A rekonstrukciót egy általunk elsajátított *CAD rendszerben* végezzük, a *Pro/ENGINEER Wildfire 5*-ben. A Pro/ENGINEER egy az iparban elterjedt tervező szoftver, mely rendelkezik pontfelhő kezelésére alkalmas modullal is. A Pro/ENGINEER-ben szabadformájú *NURBS felületekkel* dolgozunk kihasználva a rendszer textúrakezelési képességeit.

A szkennelést követően egy külső program segítségével a szkennelt pontok koordinátáit felhasználva hozzuk létre a felületet, mely illeszkedik a pontfelhőhöz. A létrehozott szabadformájú felületek már lehetőséget adnak a testmodell kialakítására, melyhez anyag rendelhető és ezután azon különböző virtuális szimulációk végezhetők.

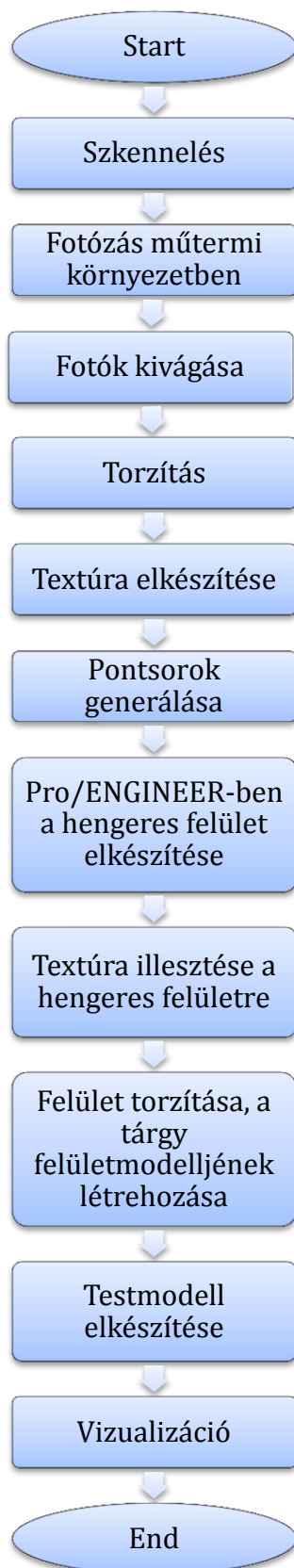
A TDK dolgozat másik célkitűzése a valósághű megjelenítéssel foglalkozik. Egy olyan textúra elkészítése a cél, ami képes a virtuális modell fotórealisztikus megjelenítésére is. A textúra elkészítését speciálisan előkészített műtermi fotózás előzi meg, ami a minél pontosabb megjelenítéshez nyújt segítséget. Ezután szabad felhasználású grafikai szoftvereket alkalmazva létrehozunk a terítéket, amit a virtuális modellre visszafeszítve alkalmazunk. Ezután foglalkozunk az elkészült modell vizualizációjával.

Az eljárás során újszerűséget jelent olyan problémák megoldása, mint például a fotózás során jelentkező torzulások kiküszöbölése. Célunk a tárgy valósághű vizualizációja és egy virtuális környezetbe illesztése. Külön figyelmet fordítunk a felület realisztikus megjelenítését segítő beállítások pontosítására. Ezután a valódi tárgy precíz mása bemutatathatóvá és felhasználhatóvá válik bárki számára.

3. A módszer bemutatása

A módszer kidolgozása során hengersizmetrikus tárggyal dolgoztunk és ennek a külső felületeit vizsgáltuk. Ezekre az egyszerűsítésekre azért volt szükség, hogy bizonyosságot nyerjünk a folyamat elvégzése után kapott eredmény létjogosultságáról. Továbbiakban az eljárás kiterjeszthető lesz bonyolultabb geometriák reprodukálására is. A további célkitűzésekben említésre kerülnek a fejlesztési lehetőségek.

A módszer bemutatása folyamatábra segítségével:

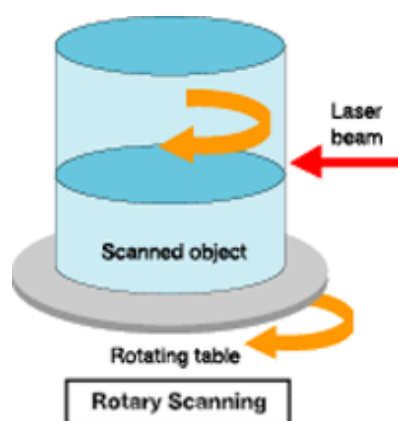


3.1. ábra A módszer folyamatábrája

3.1 A módszer előkészítő eljárások

3.1.1. 3D körszkennelési eljárás

Ideális mód a gömb alakú vagy sima felületű tárgyak gyors szkenneléséhez. Miután a tárgyat a szkennер forgó asztalához rögzítik, a lézersugár függőleges irányban pásztázza a felületet a digitális adatok létrehozására. Ha az objektum egy bizonyos területéhez nagyobb felbontás szükséges, az szkennер (LPX-250) újraszkenelés funkciója ezt lehetővé teszi a kiválasztott terület megadásával. Az eljárás elvi vázlatát az 3.2. ábra szemlélteti.



3.2. ábra A körszkennelési eljárás

3.1.2. A műtárgy fotózása műtermi környezetben

A fényképezést műtermi környezetben szükséges elvégezni a minél kisebb torzulás érdekében. Minden beállítást (*Lásd: műterem adatai*) előzetesen érdemes kiszámítani és rögzíteni. Csak így kaphatunk minél valóságosabb képeket, amelyek a látvány későbbi visszaállításához is segítséget jelenthetnek. Fontos megjegyezni, hogy a legjobb minőségű kamera sem tudja visszaadni 100%-os pontossággal a környezet viszonyait. A fotózást követő utómunkálatokhoz referenciákat - relatív-abszolút függőleges irány meghatározása (4.15. ábra), színkártya (4.14. ábra) - készítünk, ezzel segítve a számítógépen végzett pontosításokat. Figyelni kell a környezeti hatásokra (kintről beszűrődő fények, légáramlások) is, amiket minimálisra érdemes csökkenteni leárnyékolásokkal. A műtermi környezet és a fényképezőgép beállítása után, nem szabad semmin változtatni addig, ameddig minden kép el nem készült. Ezután egy forgásszimmetrikus test fényképezését körasztalon, körbeforgatással végezzük. A tárgy középvonala egybeesik a kameraállvány tengelyével. Előre meghatározott szögeket fordítunk, majd így készítjük a felvételeket. Minél kisebb szöget fordítunk, annál több kép készül, így a további munkálatok előtt célszerű eldönteni, hogy milyen pontosságban kívánjuk felhasználni az elkészült képeket.

Fényképezés utáni korrekciók:

Az abszolút szürkét megjelenítő referencia színkártyához (4.14, ábra) viszonyítva az elkészült képeken állítunk a színeken. Az expozíció mértékét növeljük, így csökkentve az árnyékolás hatását. Ezek a beállítások már megfelelően módosítják a képeket ahhoz, hogy az emberi szem számára már ne legyenek észrevehetőek a leképezés pontatlanságai. A képeket *RAW* formátumban tároljuk és használjuk tovább, mert ezek a digitális negatívok sokkal több információt tartalmaznak a képről, mint a szokványos képformátumok és az ezeken végzett színekorrekciók is kevesebb információvesztéssel járnak. Ezek után a kívánt minőségben legyártjuk a *JPG* formátumú képeket.

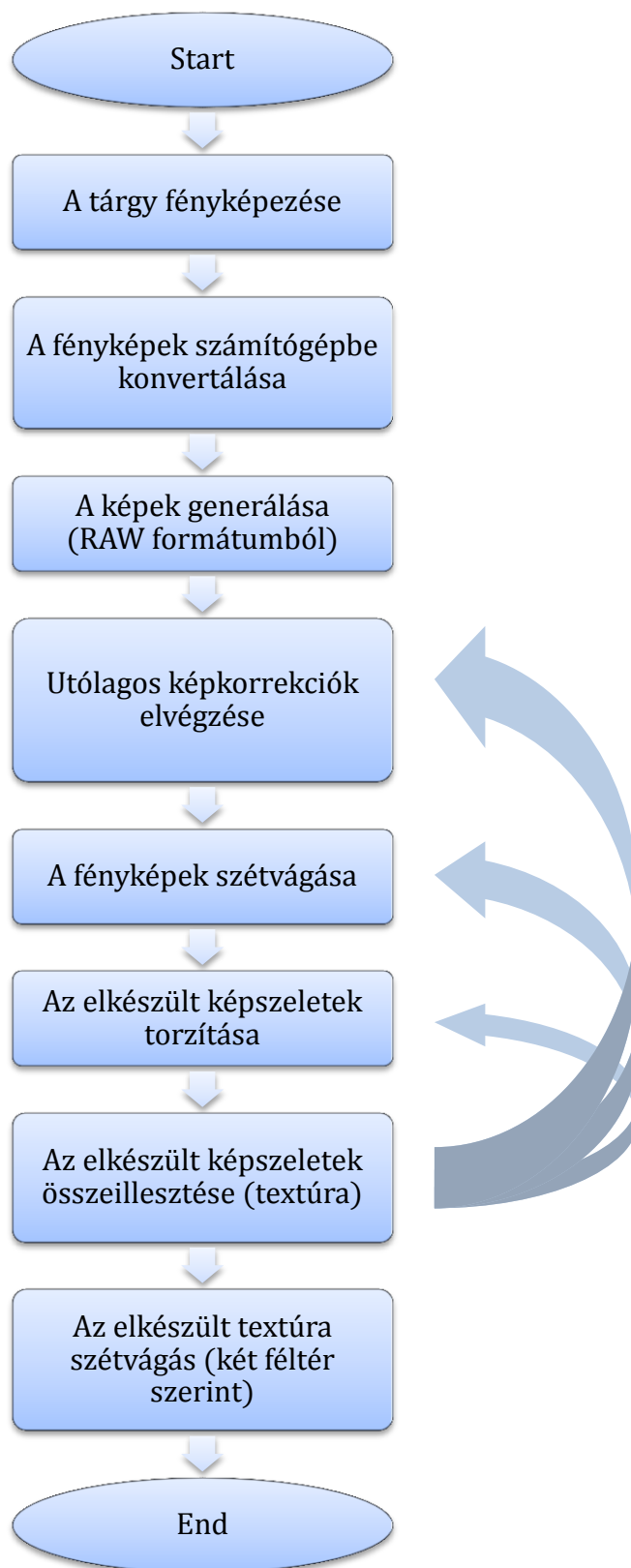
Fotózás során figyelembe kell venni a következő jelenségekből adódó pontatlanságokat:

- *szenzor zaj*
- *színhőmérséklet*
- *nem pixelpontos fénymérés*
- *Anti-Aliasing filter*
- *MOIRE jelenség*

3.2. Fotórealisztikus textúra készítés

A textúra elkészítésének az elve hasonló egy panoráma kép elkészítéséhez. Az egymással átfedésbe kerülő darabokat illesztjük össze egy nagy terítékképpé. Ehhez azonban a fotózás során elkészült képeket az adott geometriához illeszkedő méretűre kell vágnunk, torzítanunk, majd ezen műveletek után készíthetjük el, akár automatizálva is, az összeillesztett képet.

A módszer egyes lépései a következő folyamatábrán láthatóak (3.3. ábra):



3.3. ábra A fotórealisztikus textúra készítési folyamat leírása

3.2.1. Az elkészült fényképek adott méretre vágása

Mielőtt elkezdenénk kivágni a képeket, több dolgot figyelembe kell vennünk. Meg kell vizsgálni a választott tárgy profilján a felületi érintőket és az inflexiós pontokat. (Mivel forgásszimmetrikus testtel dolgozunk, ezért elég egy egyik oldalt vizsgálni.) Külön figyelmet kell fordítani a nagy görbületi ívvel rendelkező részekre és az inflexiós pontok körüli területekre. Ezek után mérlegelnünk kell, hogy egy vagy több képsort készítsünk a tárgy magasságának megfelelően a görbületeket figyelembe véve. Amennyiben kis ívekkel dolgozunk, elég lehet egy sávban elvégezni a kivágást.

Az eredeti negatívból legyártott nagy felbontású képeinket pixelpontosan szeleteljük. Az eljárás lényege, az hogy minden képen pixelpontosan megkeressük és lemérjük a tárgy középvonalát (mivel forgástestekről beszélünk), és ettől a referenciától két irányban azonos mértékben kiválasztjuk a kivágni kívánt tartományt. A fényképek tetején és alján található holtrészeket a befotózott tárgy legmagasabb és legalacsonyabb pontjához viszonyítva szintén elhagyjuk. Ezeket a méreteket kiszámítva egy téglalap alakú kijelölést kapunk, amit pixelpontosan minden képen alkalmazva kijelölünk és kivágunk.

Ezt követően egy olyan képsorozatot kapunk, ahol a forgásszimmetrikus tárgy középvonala körül lévő tartomány látszik holtrészekről mentesen. Ez a rész helyezkedik el pontosan szemben a kamerával, a kamera tengelyében, ezért a lencse szélén lévő torzítások is kevésbé hatnak rá. A felszín görbülete is pontosan szemből látszik. Ez tekinthető ideális beállításnak a képek méretre vágásához.

Minél több képet használunk, annál pontosabb és torzulás-mentesebb lesz a további munkánk, azonban ezt már manuális úton nagyon időigényes lenne elvégezni. Ezért érdemes egy olyan *script*-et írni, ami automatikusan legenerálja számunkra a megfelelően kivágott képsorozatot. Nekünk elég csak egy képen elvégezni a pixelpontos tartomány lemérését és kijelölését. Ezt a *script*-ben felhasználva a program elvégzi helyettünk a számításokat és a kivágást, majd az általunk kiválasztott célmappába belementi azokat.

Végeredményként akkor kapnánk a legpontosabb eredményt, ha a középvonalnál lévő egy pixel széles tartományt vágnánk ki, majd ezeket illeszténénk össze, de ehhez nagyobb pontosságú fokenkénti körbeforgatásra, és automatizált sorozatfelvételre lenne szükség.

3.2.2. A fényképek torzítása előzetes számítások alapján

A képszeletek kivágása után valósághű textúra készítéséhez további torzításokra van szükség. Ez azért fontos, mert azt a sávot, ahol a tárgy kerülete maximális (legnagyobb az átmérője forgásszimmetrikus tárgyak esetén), ott a képet torzulásmentesnek tekinthetjük. Viszont ahol a keresztmetszet lecsökken, inflexiót vált, ott a tárgy távolabb kerül a kamera lencséjétől, és azonos pixelszámmra több információ kerül, mivel a távolság miatt kisebbnek látszik az adott felületrész. Ezt úgy lehet elképzelni, mintha egy hengerpalást azonos nagyságú felületét kisebb helyre gyúrnánk, zsugorítanánk össze. Ezért ahhoz, hogy minél tökéletesebb kiterített téglalap alakú textúrát kapjunk, az adott magasságokhoz tartozó átmérők pixeleit, az átmérőkhöz arányosan szét kell húznunk úgy, hogy referenciának használjuk a maximális átmérőt. Elsősorban azokat az átmérőket érdemes vizsgálni, ahol nagy a görbület vagy inflexió váltás történik. Minél pontosabb eredményt szeretnénk, annál több átmérőnél kell elvégezni a kép nyújtását. A torzítás a képszelet középvonalától szimmetrikusan két irányba történik. Az ehhez szükséges számításokat a melléklet torzítási táblázat pontjában lehet megtalálni.

Ez a folyamat manuálisan elvégezve sok időt venne igénybe. Minél pontosabb eredményt szeretnénk annál több képszeletről és átmérőről van szó. Ezért érdemes erre a folyamatra is egy *script*-et írni, ami automatikusan elvégzi a torzítást és új képszeleteket generál. (lásd: mellékletek: alkalmazott *script*-ek)

3.2.3. A textúra elkészítése

A textúra elkészítését az előzőleg generált képszeletek megfelelő illesztésével valósíthatjuk meg úgy, ahogyan egy panoráma képet illesztünk össze. A képen található átfedéseket megkeressük, majd az ezeken található mintázat egymásra illesztésével egy folytonos képet alkothatunk. Ezt a folyamatot is érdemes automatizálni, mivel manuálisan sok időt venne igénybe az illesztés. Itt is igaz az, hogy minél több képet használunk fel, annál pontosabb textúrát kapunk.

Az általunk alkalmazott eljárás során egy lineáris, téglalap alakú textúra készítése a cél, mivel forgásszimmetrikus testre zsugorítjuk vissza. Azonban az előzetes torzítás (*Lásd: 3.2.2 fejezet*) során keletkezhetnek hibák, amik az illesztés további deformálódásához vezethetnek.

3.3. Felületillesztés MATLAB programban

A szkennelés után egy pontfelhő áll rendelkezésünkre. Mint ahogy azt már az előzőekben áttekintettük a pontfelhő kezelésére számos egyéb program létezik. A módszer kidolgozása során célunk egy testmodell létrehozása lesz kihasználva az adott *CAD rendszer* lehetőségeit.

A módszer során a szkennelt pontok koordinátái alapján egy általunk megírt program segítségével hozzuk majd létre a *NURBS felületeket*, melyek segítségével integráltan végezzük el a textúrák illesztését a virtuális modellre.

A módszer alapötlete az volt, hogy szabályos, köríven elhelyezkedő pontsorokat generálunk, melyek adott távolságban helyezkednek el egymástól, majd ezek segítségével felületet hozunk létre, mely körülveszi a pontfelhőt. Azért használunk köríven elhelyezkedő pontsorokat, vagyis hengeres felület, mert hengerszimmetrikus tárgyat választottunk és a textúra elkészítése is hengeres vetítésen alapult. A szabályos pontsorok koordinátáinak módosításával, melyet a pontfelhő pontjainak koordinátái szerint végzünk el, fogjuk a felületet torzítani, létre hozva a szkennelt tárgy külső felületét. A módszer kidolgozásának ebben a fázisában még egyszerűsítéseket teszünk(3.fejezet), így csak a külső felületekkel dolgozunk.

A pontsorok generálásához és a megfelelő szkennelt pontok koordinátáinak kiválasztásához szükségünk volt egy olyan programra, mely elvégzi ezeket a műveleteket. Mivel ilyen program eddig nem létezett, így a programot saját magunknak kellett megírni. A program kód megírásához a *MATLAB programot* használtuk, mivel ez az egyik legelterjedtebb programozási keretrendszer. A dolgozat elkészítése során célunk egy széleskörűen alkalmazható módszer kidolgozása volt, így hasznosnak tűnt ha a programkódot bármilyen esetben magunk tudjuk módosítani. A program létrehozása azért is szükséges, mivel az eljárást, amilyen mértékben az lehetséges automatizálni szeretnénk. A program könnyen módosítható változókkal dolgozik, így mi magunk definiálhatjuk ezeket a paramétereket az adott feladathoz illesztve.

3.3.2. Koordináta transzformációk

A megírt programban koordináta transzformációkat végzünk el. Ezek segítségével határozzuk majd meg azokat a szkennelt pontokat, melyek egy adott ponttól adott távolságra helyezkednek el. Ezeknek a pontoknak a koordinátái szerint módosítjuk majd a generált pontsorok koordinátáit.

A koordináta transzformációs műveleteket *Petrich Géza: Ábrázoló geometria* című könyv alapján végeztük el [8].

A koordináta transzformáció

Koordináta-transzformációról akkor beszélünk, ha a tárgy egy új koordináta-rendszerre vonatkozó koordinátáit határozzuk meg a régiek ismeretében. Ilyenkor a tárgyat változatlanak tekintjük, csak a nézőpont változik.

A koordináta-rendszereket derékszögű, jobbsodrású, Descartes-féle koordináta-rendszereknek tekintjük. A transzformációk egységes kezelése érdekében az ún. homogén koordinátákat is bevezetjük. Egy 3-dimenziós térbeli P pont $(x; y; z)$ Descartes-koordinátáit (inhomogén koordináták) kiegészítjük egy negyedik koordinátával, az 1-gyel. Így a P pont homogén koordinátái $(x; y; z; 1)$.

Transzformációs mátrixok

A térbeli koordináta transzformáció során az alábbi transzformációs mátrixokkal kell az adott pont koordinátáinak mátrixát megszorozni attól függően, hogy honnan hova transzformáljuk a pontot.

Az eredeti koordináta rendszerből az új koordináta rendszerbe transzformáló mátrix:

$$T_1 = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & t_x \\ \sin(\theta) & \cos(\theta) & t_y \\ 0 & 0 & 1 \end{bmatrix} \quad (1)$$

Az új koordináta rendszerből az eredeti koordináta rendszerbe visszatranszformáló mátrix:

$$T_2 = \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -t_x \\ 0 & 1 & -t_y \\ 0 & 0 & 1 \end{bmatrix} \quad (2)$$

$$t = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} \quad (3)$$

A θ az eredeti koordináta rendszer és az új koordináta rendszer azonos tengelyei által bezárt szög. A t az eredeti koordináta rendszer origójából az új koordináta rendszer origójába mutató vektor.

3.3.4. Program a felületek generálására és illesztésére

A változókat a program elején tudjuk módosítani. A legfontosabb változó a pontok egymástól mért függőleges és vízszintes távolsága (*gDens*), mely így megegyezik a szabályos generált pontsorok *z* irányú távolságával is. Ebből a változóból (*gDens*) számolja a program a szintek száma (*g*), mely függ az adott tárgy paramétereitől is, a kívánt körív átmérőjét és a pontsorok által felépített henger magasságát kell megadni.

$$g = \text{GridGen}(\text{átmérő}, \text{magasság}, gDens) \quad (4)$$

A koordináta transzformációs műveleteket, vagyis a megfelelő szken pontok megkeresését, mindig az adott *z* szinten (*zLevel*) *x-y* síkban végezzük el, majd továbblépünk a következő szintre. Ezen algoritmus alapján fut végig a program, a legelső szinttől (*zMin*) az utolsó szintig (*zMax*).

A szintek *z* irányú távolságának változtatásával tudjuk a módszer pontosságát is változtatni, mivel minél sűrűbben vesszük fel a pontsorokat, vagyis *gDens*-et csökkentjük, annál pontosabban illeszkedik a létrehozott felület az eredeti pontfelhőhöz.

A változók definiálása után a pontsorok, a *gridek* generálása következik. A dolgozat elkészítése során hengerszimmetrikus tárgyal dolgozunk, így a programot úgy írtuk meg, hogy adott átmérőjű félköríven elhelyezkedő pontokat hozzon létre. A félkörívek használta azért szükséges, mivel a *Pro/ENGINEER CAD rendszer* félterekkel modellezi. A köríven a pontok egyenletesen oszlanak el, a pontok számát és így a távolságát, valamint a pontsorok távolságát is *gDens* határozza meg.

A *gridek* generálása után meghatározzuk, hogy milyen módon mentse fájlba a program a pontsorokat. Mivel félköríven elhelyezkedő pontsorokat hoztunk létre, ezért a grideket attól függően, hogy melyik oldalhoz tartoznak kétféleképpen neveztük el, vagyis a jobb oldali *gridek* neve *grid_Rz* lett, és a baloldali *gridek* neve *grid_Lz* lett. A gridek neveinek generálásánál arra is figyeltünk, hogy azok a *z* irányú magasságtól függően minden esetben növekvő sorrendbe kerüljenek egymás alá a mappában, melybe a program a fájlokat menti. A fájlok kiterjesztése *.pts* formátum, melyet a *Pro/ENGINEER rendszer* be tud olvasni. A *gridek* generálás után a szabályos pontsorokkal többet nem dolgozik a program.

Ezután a megjelenítési beállításokat definiáljuk. A program futtatása során az elvégzett műveleteket kirajzoltatjuk, így tudjuk ellenőrizni azt, ha valahol esetleg megakadna a program. Megjelenítési beállításokat több helyen definiálunk a programban.

A programban az eddigi beállítások után egy ciklust futattunk, mely *zLevel* alapján fut majd végig, az első szint *zMin* lesz. Ebben a részben definiált műveleteket a program elvégzi az adott szinten, majd visszalép a ciklus elejére, meghatározza a következő szintet és újra lefutatja a műveleteket.

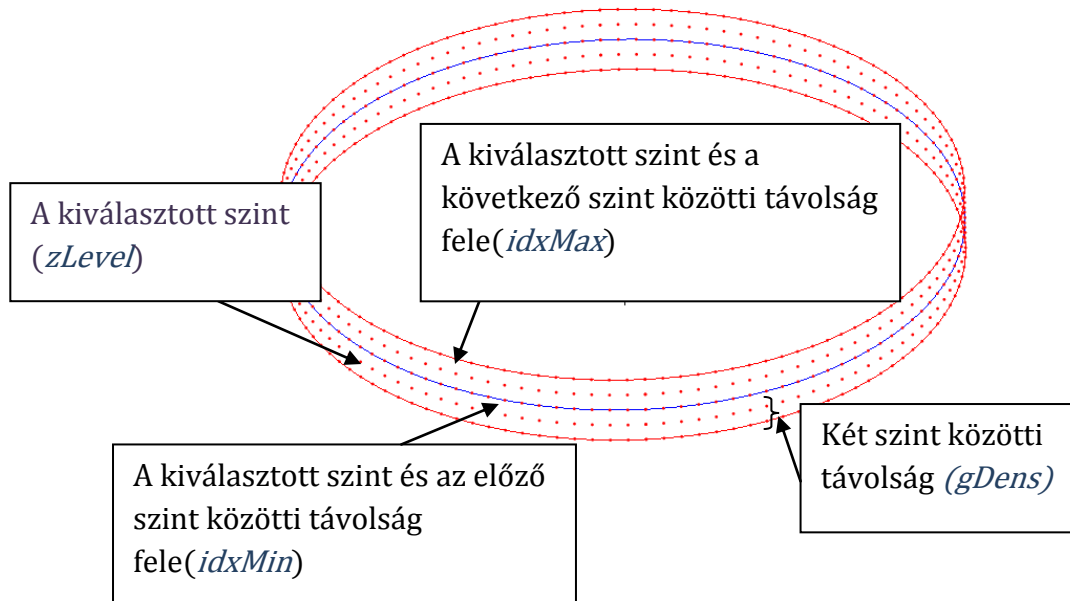
Az első lépés a szkennelés adatok beolvasása. Az adatok szöveg fájlként való behívása után, a pontok koordinátáit mátrixba rendezi a program, a homogén koordináta(hozzáadásával). A szkennelés pontok változója: *Points*.

$$Points = \begin{bmatrix} t(:,1)' \\ t(:,2)' \\ homegenCoords \\ t(:,3)' \end{bmatrix} \quad (5)$$

A megfelelő szkennelés pontok megtalálása úgy történik, hogy az adott z magasságú síkban kivágunk egy hengeres sávot, melynek a vastagsága az adott z szinttől a z szintek távolságának a fele felfelé plusz(*idxMax*), lefelé mínusz(*idxMin*) előjellel, így a sávok összeértnek. Ebben a sávban keressük majd a megfelelő szkennelés pontot (3.4.ábra)

$$idxMin = [s(:,3) \geq zLevel - \frac{gDens}{2}]_1 \quad (6)$$

$$idxMax = [s(:,3) \leq zLevel + \frac{gDens}{2}]_1 \quad (7)$$

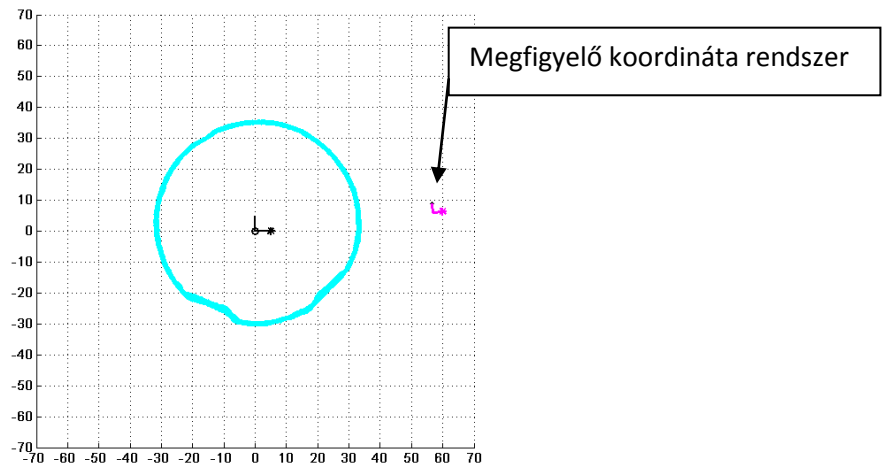


3.4. ábra A vizsgált terület

A program ezután adott számú megfigyelő koordináta rendszert(nCS) hoz létre, melyek az origó körül egyenletesen elosztva adott szöggel($theta$), adott távolságban($dist$) helyezkednek el. A megfigyelő koordináta rendszereket különböző színekkel jelöljük, melyek ciklikusan ismétlődnek (3.5. ábra).

$$angle = \frac{360}{nCS} \cdot i \quad (8)$$

$$theta = angle \cdot \frac{\pi}{180} \quad (9)$$



3.5. ábra A megfigyelő koordináta rendszer

A megfigyelő koordináta rendszerekre is egy ciklust futattunk, vagyis az ezután definiált műveleteket elvégezzük először az első megfigyelő koordináta rendszerben, majd tovább lépünk a következő megfigyelő koordináta rendszerre, így tovább sorban haladva míg az összes műveletet meg nem oldjuk az összes koordináta rendszerben. Ezután lépünk majd tovább a következő z szintre.

A megfigyelő koordináta rendszer definiálása után az első művelet az, hogy az összes szken pontot transzformáljuk az eredeti koordináta rendszerből az adott megfigyelő koordináta rendszerbe. Ehhez definiáljuk a transzformációs mátrixokat, melyek segítségével a transzformációk elvégezhetők.

Transzformációs mátrixok: ($CoordTrans1$; $CoordTrans2$).

A megfigyelő koordináta rendszerből az eredeti koordináta rendszerbe transzformáló mátrix:

$$CoordTrans1 = \begin{bmatrix} \cos(theta) & -\sin(theta) & dist(1) \\ \sin(theta) & \cos(theta) & dist(2) \\ 0 & 0 & 1 \end{bmatrix} \quad (10)$$

Az eredeti koordináta rendszerből a megfigyelő koordináta rendszerbe transzformáló mátrix egy forgató mátrix(*CoordRot*) és a lineáris transzformációt tartalmazó mátrix(*CoordMove*) szorzatából tevődik össze:

A forgató mátrix:

$$CoordRot = \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (11)$$

A lineáris transzformációs mátrix:

$$CoordMove = \begin{bmatrix} 1 & 0 & -dist(1) \\ 0 & 1 & -dist(2) \\ 0 & 0 & 1 \end{bmatrix} \quad (12)$$

A kettő szorzatából képzett transzformációs mátrix:

$$CoordTrans2 = CoordRot * CoordMove \quad (13)$$

A szken pontok transzformálása a megfigyelő koordináta rendszerbe, majd a pontok koordinátáinak mátrixba rendezése:

$$P2xyTrans = CoordTrans2 * Points \quad (14)$$

$$P2zTrans = Points - zLevel \quad (15)$$

$$P2 = vertcat(P2xyTrans, P2zTrans) \quad (16)$$

A transzformált szken pontok *P2*.

Ezután „y” koordináták szerint sorba rendezzük az összes szken pontot (*P2Ysorted*), majd az előzőekben definiált *idxMax*, *idxMin* értékek közé eső „y” koordinátájú pontokat kiválogatjuk (*P2Yselected*). A kiválasztott tartomány határai megállapításához a sorba rendezett koordináták megfelelő indexeinek az első (1) és utolsó (n) előfordulását keressük meg:

$$idxMax = [P2Ysorted(:,2) \leq 1 \cdot \frac{gDens}{2}]_1 \quad (17)$$

$$idxMin = [P2Ysorted(:,2) \leq -1 \cdot \frac{gDens}{2}]_n \quad (18)$$

$$P2Yselected = \begin{bmatrix} P2Ysorted(idxMin) \\ \vdots \\ P2Ysorted(idxMax) \end{bmatrix} \quad (19)$$

Ha megvannak a „y” koordináta alapján kiválogatott pontok ($P2Y_{selected}$), akkor az „x” koordináta szerint sorba rendezzük őket ($P2X_{sorted}$), és megkeressük a legnagyobb „x” koordinátájú pontot ($idxP2X_{sorted}$). Két ilyen pontot fog a program találni, mivel hengerszimmetrikus a tárgy, így a megfigyelő koordináta rendszerhez közelebb eső pontot, és a tárgy másik oldalán, a megfigyelő koordináta rendszertől legtávolabb eső pontot is meg fogja találni. Ezt a két pontot sorba rendezzük ($P2XY_{selected}$).

$$idxP2X_{sorted} = [P2X_{sorted}(:,1) \geq -1 \cdot radius]_n \quad (20)$$

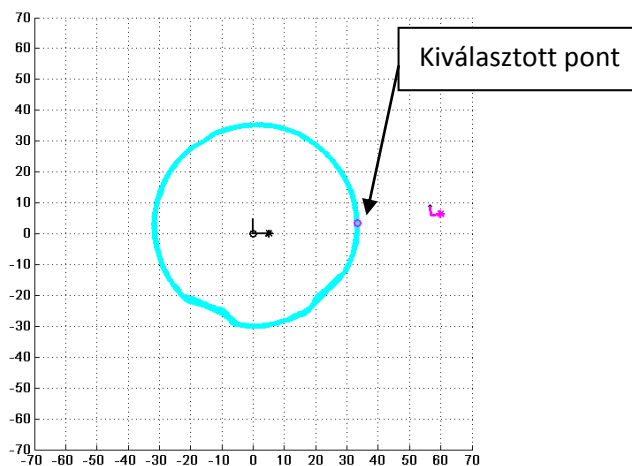
$$P2XY_{selected} = \begin{bmatrix} P2X_{sorted}(1) \\ \vdots \\ P2X_{sorted}(idxP2X_{sorted}) \end{bmatrix} \quad (21)$$

Ezután megkeressük a megfigyelő koordináta rendszerhez közelebbi pontot, vagyis azt amelyik a megtalált két pont közül a megfigyelő koordináta rendszer x tengelyéhez közelebb van ($P2_{near}$), ez lesz a kiválasztott pont. A kiválasztás után a pontot visszatranszformáljuk az eredeti koordináta rendszerbe ($P2_{selected}$).

$$P2_{selected} = CoordTrans1 * P2_{near} \quad (22)$$

Mivel hengeres felülettel vettük körül a tárgyat, de a tárgy nem szükségszerűen tölti ki a hengeres felületet, így vannak részek, ahol nincsenek szken pontok. Ha a program nem talál szken pontokat, akkor az utolsó kiválasztott szken pont koordinátáját másolja tovább, csak a z koordinátát cseréli ki az adott z szint szerinti z értékre, az x, y koordinátákat megtartja.

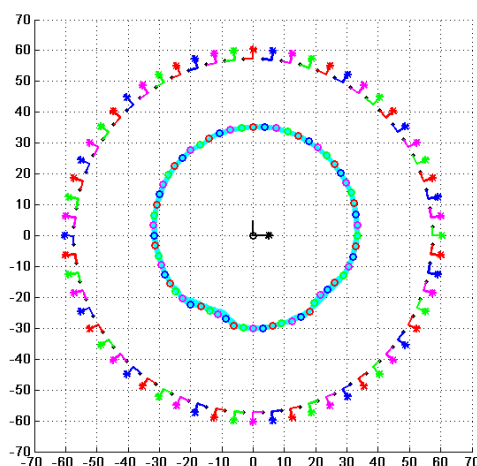
A kiválasztott pontokat az adott megfigyelő koordináta rendszerhez tartozó színnel megjelöljük. (3.6. ábra).



3.6. ábra A kiválasztott pont

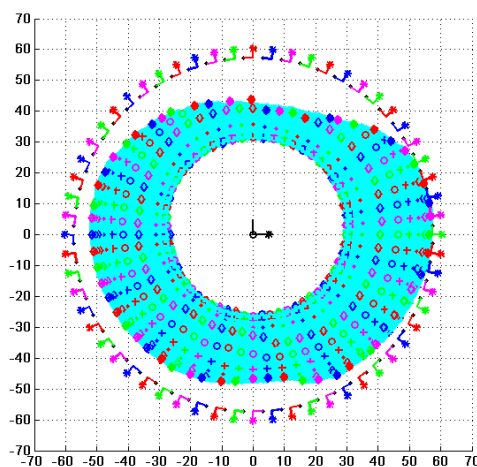
Ezután tovább lépünk a következő megfigyelő koordináta rendszerre. Miután az adott szinten az összes megfigyelő koordináta rendszerben megkerestük a pontokat (3.7.

ábra) és ezeket visszatranszformáltuk az eredeti koordináta rendszerbe, tovább lépünk a következő z szintre, ahol ugyan csak elvégezzük az előzőekben definiált műveleteket. A szintek ciklusát *zMax*-ig futtatjuk.



3.7. ábra Az adott szinten kiválasztott pontok

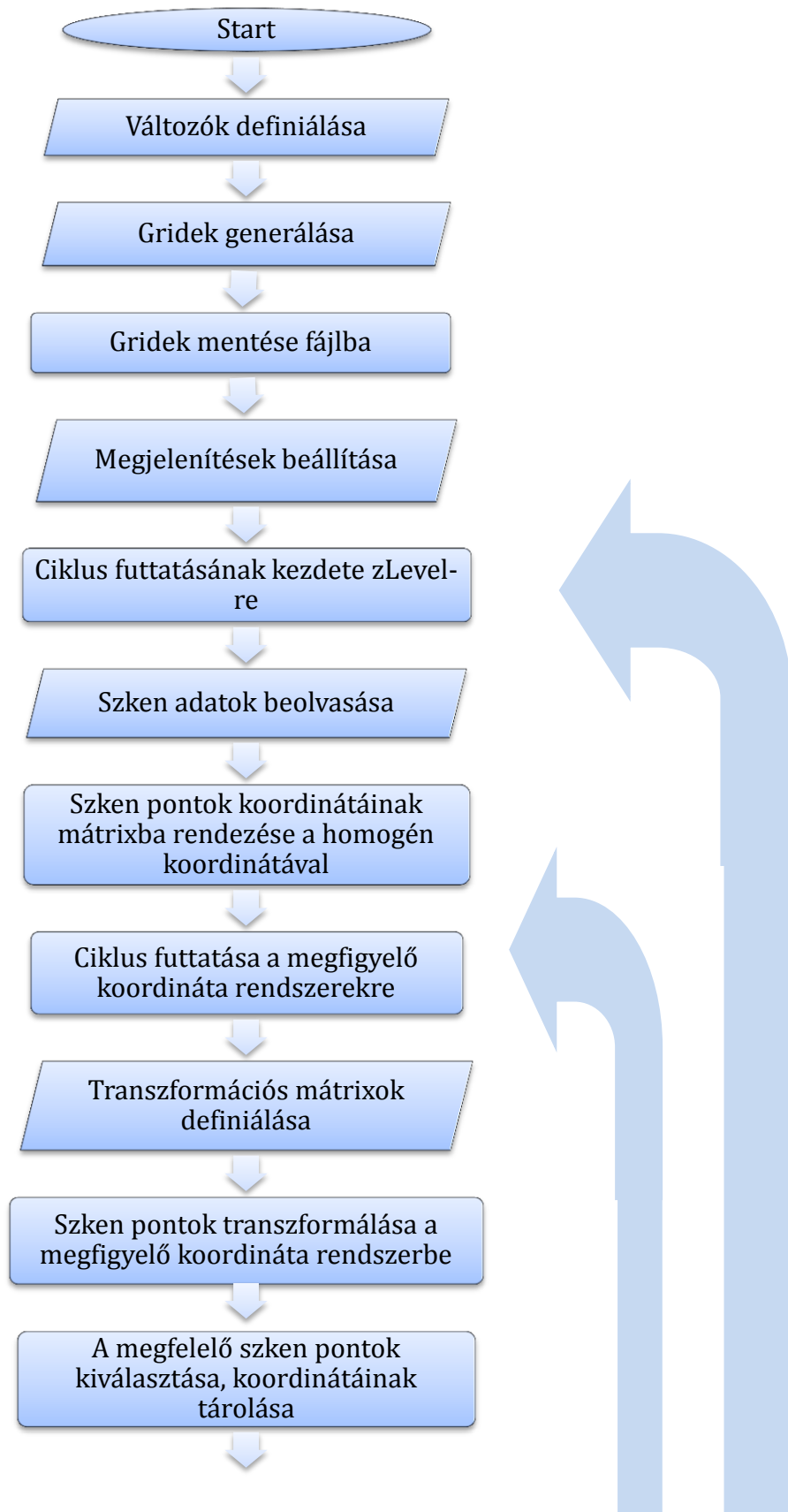
A lefutott program által generált kép (3.8. ábra).

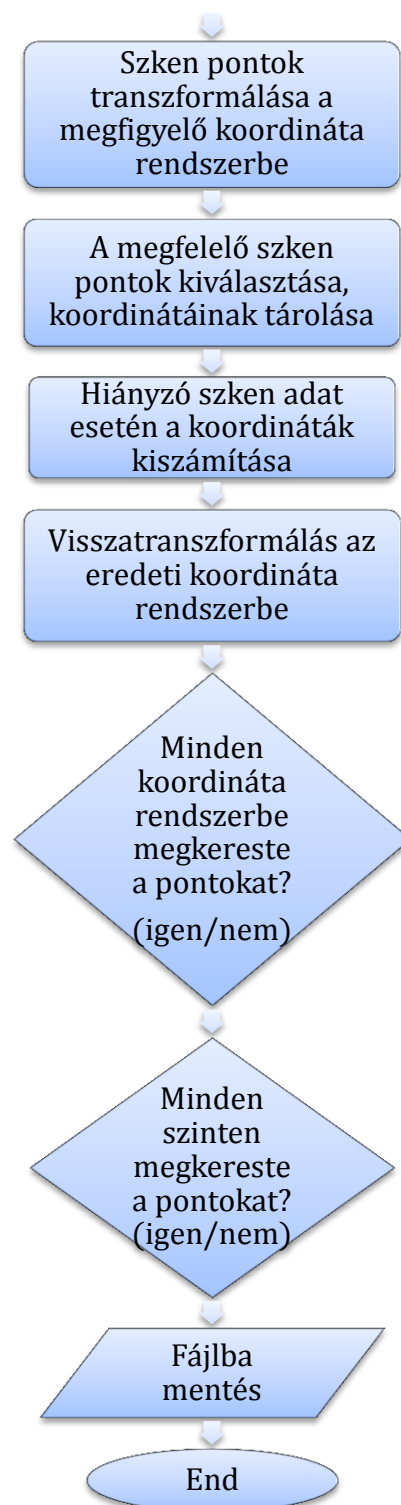


3. 8. ábra A lefutott program által kirajzolt kép

A program utolsó felében a módosított pontsorokat menti fájlba a program. A fájlba mentés során hasonlóan generáltuk a fájl nevét mint a *gridek* esetén azért, hogy a szabályos generált pontsorokat és a hozzájuk tartozó módosított pontsorokat könnyen lehessen összepárosítani, azonosítani. A jobb oldal módosított pontsorainak neve *mod_Rz* lett, míg a bal oldal módosított pontsorainak neve *mod_Lz*. A módosított pontsorok számozása és kiterjesztése is azonos a *gridek* esetén leírtakhoz.

Folyamatábra: A program felületek generálására és illesztésére



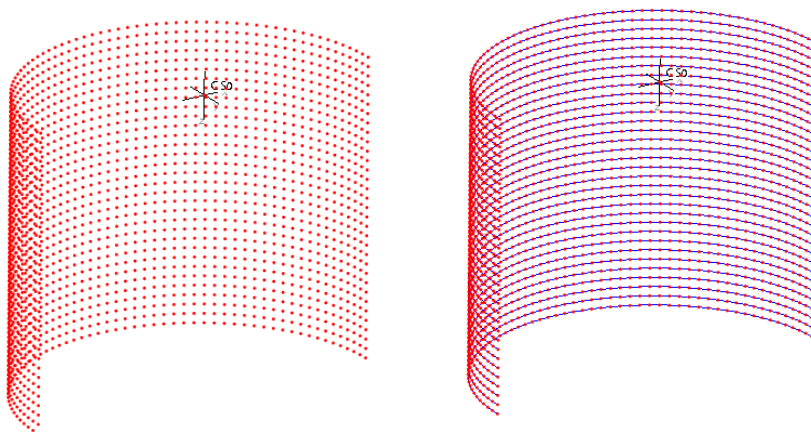


3.9. ábra A program folyamatábrája

3.4. A 3D-s testmodell létrehozása Pro/ENGINEER-ben

Miután a *MATLAB* megírt program segítségével létrehoztuk a *grideket* és a módosított pontsorokat a módszer további részét, vagyis a 3D-s testmodell létrehozását a *Pro/ENGINEER CAD rendszerben* végezzük el.

3.4.1. A pontsorok behívása, a hengeres felület létrehozása

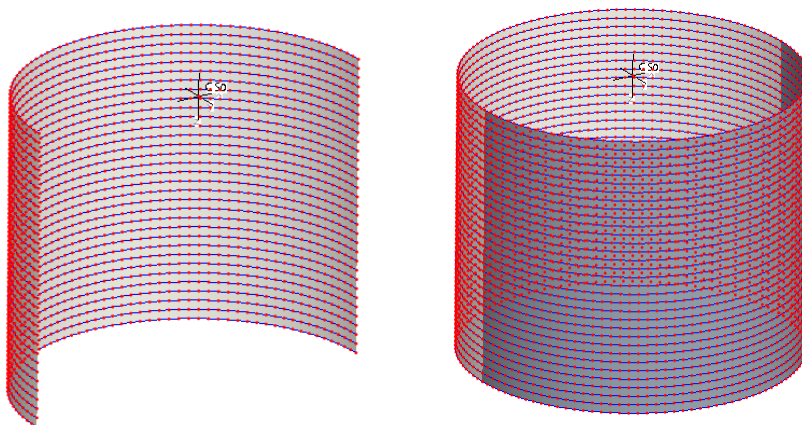


3.10. ábra a) A generált pontsorok; b) A pontsorokra felvett szplájnok

A Pro/ENGINEER programba egymás után beolvassuk először az egyik féltér(jobb vagy bal oldal) generált szabályos pontsorait(*grideit*), majd a másik féltér pontsorait. (3.10.ábra a)).

Ezután a gridekre *B szplájn* görbéket illesztünk. A *Pro/ENGINEER*-ben a B- *szplájn* olyan szabad görbe, mely illeszkedik a pontsor összes pontjára. (3.10.ábra b)).

Ezt azért szükséges megtenni, mivel felületet csak görbékre lehet illeszteni. Ha a görbék adottak a felület elkészítése gyors és egyszerű. A módszer ezen részének befejezése után két felület rész áll rendelkezésünkre(3.11.ábra).



3.11. ábra a) Félhengeres felület; b) A teljes henger felület

3.4.2. A textúra illesztése

A felület elkészítése után a felületre illesztjük a textúrát. A módszer során azért nem rögtön a módosított pontsört hívtuk be, mivel az egyik fő célunk az volt, hogy a tárgy textúráját is valósághűen megtartsuk, így a befotózott képet a hengeres felületre kell illeszteni, majd ahogy módosítjuk a pontokat a textúra követi a változtatásokat. Azért használunk hengeres felületet, mert a *Pro/ENGINEER*-ben hengeres vetítést alkalmazunk, így a felületre torzítás nélkül vetítjük az elkészített textúrát.

A textúrát szét kell vágni két részre, mivel két félhengeres felület hoztunk létre, majd a megfelelő felületrészre kell illeszteni a hozzátartozó textúra részt. A szétvágásnál a hengeres felület részeket kell figyelembe venni azt, hogy a pontsorok módosítása után hogy fog állni a tárgy.

A hengeres felület feltextúrázása során nagy hangsúlyt kell fektetni a pontos illesztésre, mert a legkisebb eltérés is a textúra torzulását idézi elő.

3.4.3. A módosított felület, majd a testmodell elkészítése

A textúra illesztése után kezdhetjük a felépített pontsorok koordinátáinak a módosítását, pontsorról pontsorra. Ezután azoknak a felület részeknek a kivágása következik, ahol már nem voltak szkenn adatok, de a megírt program másolta az előző pont koordinátáit, így van felület ezeken a részeken. Ezek a felületek már nem részei a tárgynak, így eltávolításuk szükséges. Végül elkészítjük a testmodellt.

Az így elkészült a testmodellhez anyagot rendelhetünk, így már tömeget mérhetünk és egyéb számításokat, szimulációkat is végezhetünk. A testmodell felhasználásával akár a hiányzó részek visszamodellezése is lehetséges.

3.5. Fotorealistikus megjelenítés a Pro/ENGINEER PhotoLux moduljában

A korábban leírt eljárás során elkészült testmodell vizualizációját a műtermi környezettel azonos beállításokkal rendelkező renderelt képpel és videóval tesszük lehetővé. Célunk, az eredeti fényképpel megegyező, virtuális reprodukció. Ezt a teljes környezet és annak beállításainak *Pro/ENGINEER* render moduljában történő visszaállítással oldjuk meg. Ezt követően vizsgáljuk az eredeti modell és végeredményül kapott kép hasonlóságait és esetleges eltéréseit.

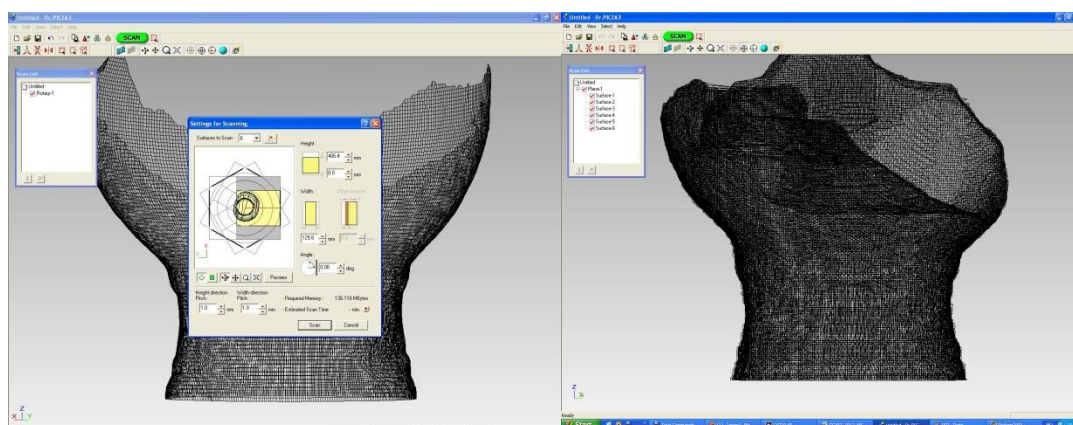
A folyamat során virtuálisan rekonstruáljuk a műtermet (lásd: műterem beállításai) az eredeti világítással és beállításokkal együtt. Ezt követően a textúrával ellátott testmodellhez anyagot rendelünk, majd az eredeti fényképből kimásoljuk az eredeti anyag színét tartalmazó RGB színkódot és ezt is virtuálisan visszaállítjuk a modellen. Addig változtatjuk a modell vizualizációját befolyásoló effektusokat, ameddig a valósággal megközelítőleg azonos képet kapunk eredményül.

4. A módszer alkalmazása a kiválasztott műtárgyon

4.1. A technológiát előkészítő eljárások

4.1.1. A műtárgy szkennelése

A műtárgy szkennelését az Óbudai Egyetem közreműködésével végeztük. A műtárgy méretéből adódóan a szkennelőkörasztalára könnyen rá tudtuk helyezni, (*Lásd: szkennelőkörasztal*) és nem akadályozta semmi a zavartalan körbefordulást. Esetünkben a körszkennelés választása tűnt ideálisnak, mert egy megközelítőleg forgás szimmetrikus test külső felületének letapogatására volt szükség (4.12 ábra). Az eljárás során, a gép alján elhelyezett körasztal folyamatosan körbefordult, így a lézersugár függőleges irányban pásztázta és továbbította az előzetesen beállított sűrűségben (0,2mm x 0,2mm) letapogatott pontkoordinátákat. A szkennelőkörasztal a beépített szoftver segítségével azonnal rögzítette, és leképezte a lézer által érzékelt pontokat. Minden ponthoz pontos koordináták tartoznak (x, y, z). Eredményül egy pontfelhőt kaptunk, amit a *CAD rendszerben* is meg tudunk tekinteni.



4.12. ábra a/ A 3D szkennelőkörasztal beállítási adatai b/ a 3D szkennelőkörasztal által letapogatott pontfelhő

Szűrés:

A felület mázassága nem okozott visszaverődést és csillogást, ezért lézer szkennelés közben nem keletkeztek hibás pontok. Az eljárás hátránya, hogy a felület egyenetlenségeiből adódóan, azokban a tartományokban, ahol a műtárgy hátsó fala magasabbra nyúlt, mint az első fala, és azokon a részeken, ahol a törésvonalra is ráért a szkennelőkörasztal, ott többlet pontok is keletkeztek, melyek már nem a külső felület részei. A pontfelhőt behívva a *Pro/ENGINEER* programba a perem részeken a számunkra nem szükséges pontokat, melyek csökkentenék az eljárás pontosságát a törésvonalak mentén, manuálisan könnyen törölhetjük. A manuális törlés nem a legpontosabb eljárás, erre a problémára megoldást jelentene, ha nem csak a külső szkennelt felülettel dolgoznánk, hanem a belső és a perem felületeket is mi hoznánk létre. (*Lásd 3.1. A módszer bemutatása*)

4.1.2. A műtárgy fotózása műtermi környezetben

A műtárgyat egy szerszámgép körasztalába rögzített tokmányára helyeztük, majd ezzel végeztük el a nagy pontosságú 360 fokos körbeforgatást. A körasztal mechanizmusa biztosította a pontos körbeforgatást. A képek 10 fokonként készültek (4.13 ábra) a műtermi beállításokon semmit sem változtatva (*Lásd: 3.1.2 fejezet*), azonos magasságban. A beállítások és a műtermi adatok a mellékletben találhatóak.



4.13. ábra A műtárgy fényképezése 10 fokonkénti körbeforgatással

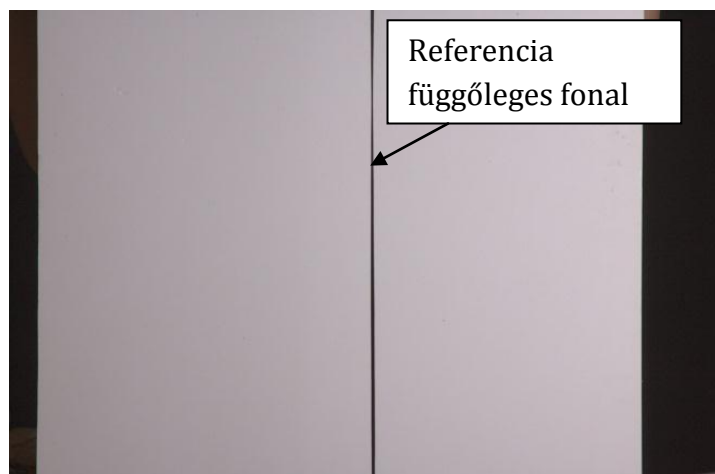
A műtárgy belsejét papírral béleltük azért, hogy azokon a helyeken, ahol a hátsó fal magasabbra nyúlik, mint az első, ott a fényképeken ne látszódjon ez a hátsó felület. A váza peremének kontúrja fehér háttér előtt sokkal pontosabban rajzolódik ki. Ez megkönnyíti a további textúra készítő eljárások elvégzését. A függőleges irány kijelöléséhez referenciaként egy fonalat feszítettünk ki (4.15 ábra).

Fényképezés utáni korrekciók:

A *RAW* formátumú digitális negatívak *JPG*-vé generálását a *RawTherapee* nevű szabad felhasználású programban végeztük. A színeket az abszolút szürke referenciakártyához (4.14 ábra) igazítva módosítottuk. Az expozíciót növeltük, ezáltal a műtárgyba gyűrt papír fehér színe néhol beégett, csökkentek rajta a részletek. Ez a későbbi textúra készítésnél lesz fontos (Lásd: 4.2.3. fejezet). Ezután legyártottuk a *JPG* formátumú képeket a negatívból 3514x2340 pixel felbontásban.



4.14. ábra Az abszolút szürkét megjelenítő színekártya (képkorrekciókhoz szükséges)

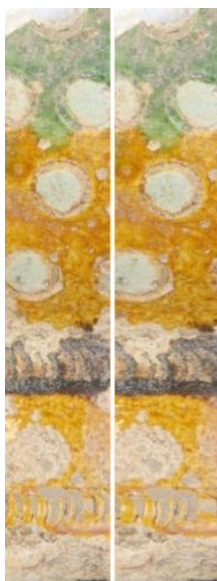


4.15. ábra A fényképeken értelmezett abszolút függőleges referencia

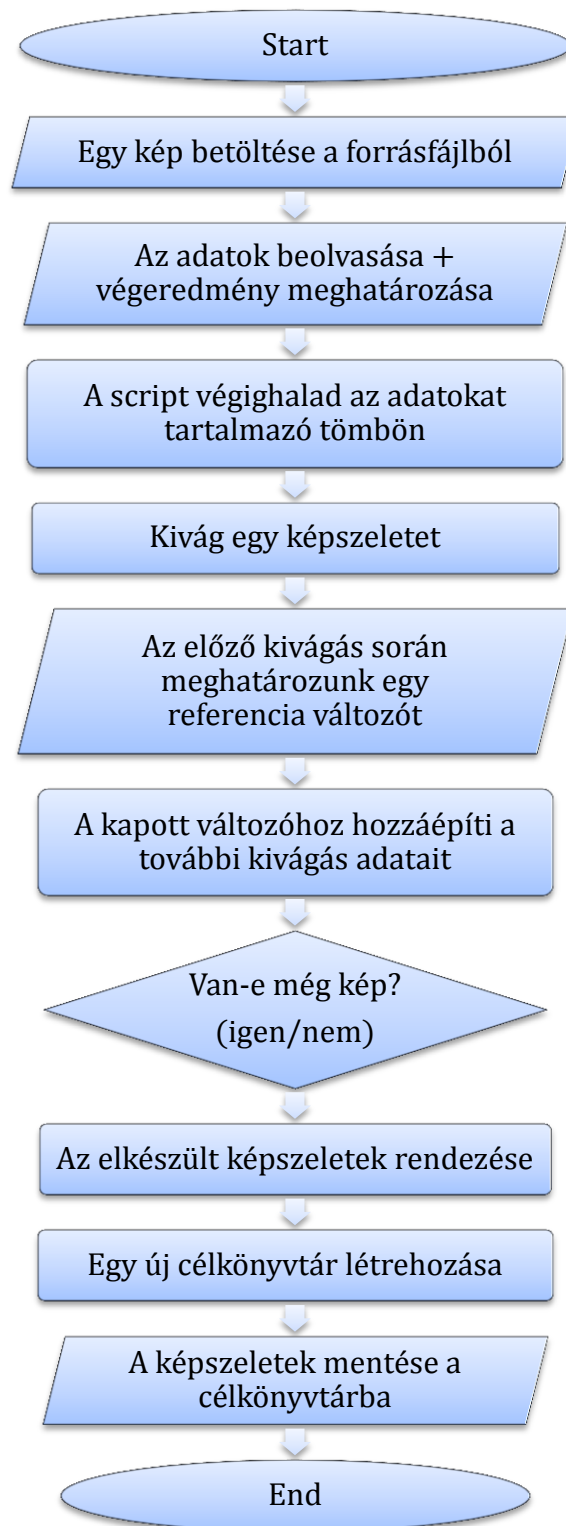
4.2. Fotórealisztikus textúra készítés

4.2.1. Az elkészült fényképek adott méretre vágása

A műtárgyunk esetében 10 fokenkénti 360 fokos körbeforgatást alkalmaztunk (*Lásd 3.1.2*). Ez 36 db fényképet eredményezett (*4.13 ábra*). Megkerestük azt a képet (8434_14.jpg), ahol a műtárgy legmagasabb pontja került középre, és ezen a képen végeztük el a hasznos tartomány pixelpontos lemerését. A 200 pixel széles tartományt választottuk (*4.16 ábra*) (100-100 pixel a középvonaltól (*Lásd: 3.2.1*)), mert így a képszeletek közötti átfedés elegendő, mégsem túl széles ahhoz, hogy a torzítás miatt később rossz minőségű textúrát kapjunk. Adatok: a képszelet szélessége 200 pixel, magassága 1330 pixel. Ezek esetünkben a váza középvonala a jobb szélről 1713 pixel (x koordináta) távolságra helyezkedik el, a kép aljától pedig 686 pixelre (y koordináta). Miután a kijelölt terület lemerése megtörtént elkészítettük az automatikus kivágáshoz szükséges *script*-et (*Lásd: Mellékletek: alkalmazott scriptek*). A *script*-ben meghatároztuk a kiinduló koordinátákat és a szeletkészítés paramétereit, majd ezek után automatikusan legenerálta a képszeleteket.



4.16. ábra a/ Torzítatlan képszelet b/ Torzítás utáni megfelelően szétnyújtott képszelet

A script működési elve:*4.17. ábra A kivágást végző script működési elve*

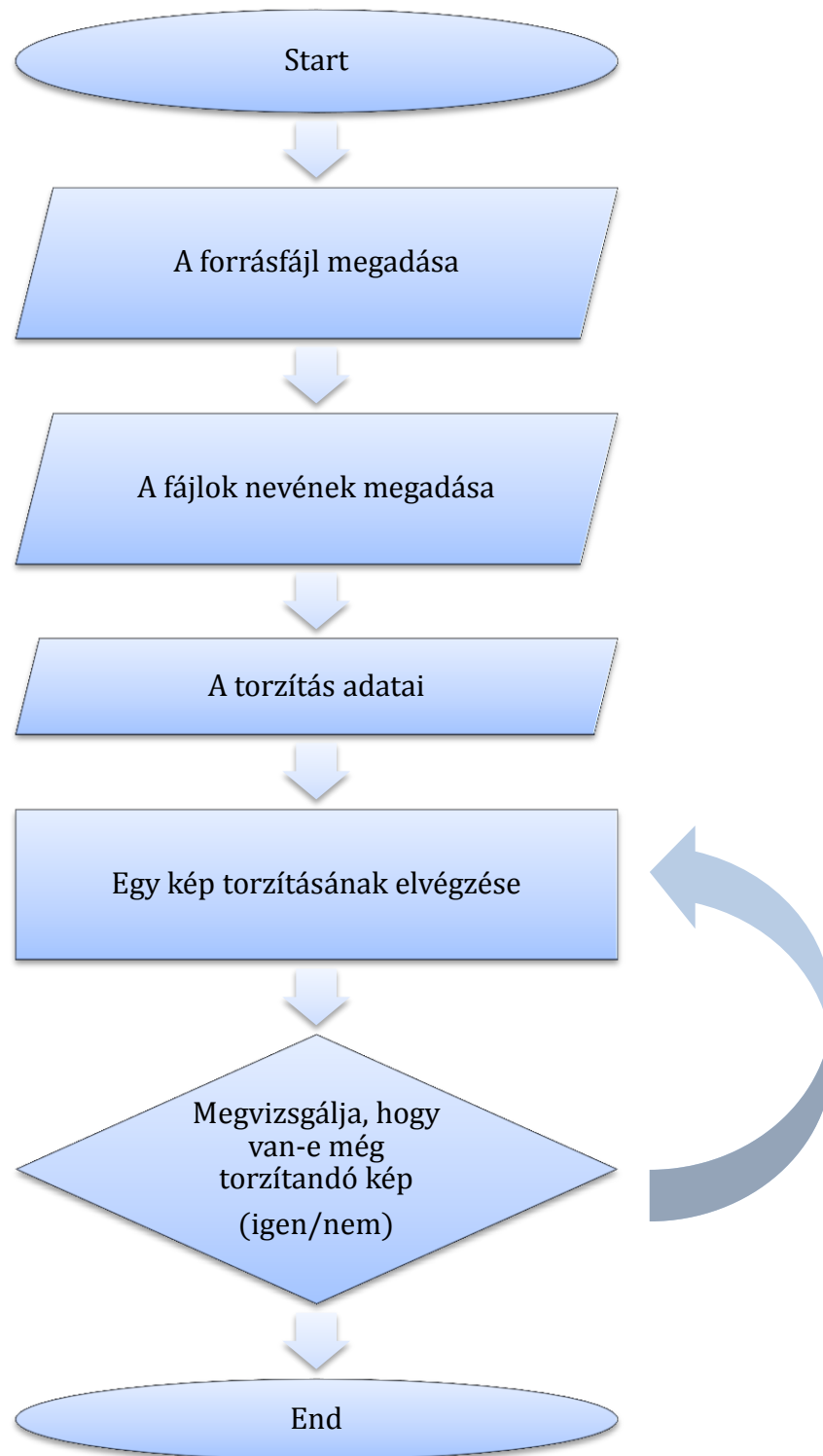
Az általunk választott műtárgy esetében a textúra készítés első fázisában nem volt egyértelmű eldönteni, hogy egy vagy több sávban érdemes kivágni a képszeleteket a görbületváltások és a felület mintázata miatt. Az összeillesztés fázisában kísérletet tettünk (*Lásd: 4.2.3 fejezet*) több fajta és különböző minőségű textúra elkészítésére. Ezért több kivágott képsorozat is elkészült, de a *script* (*Lásd: Mellékletek: Alkalmazott scriptek*) segítségével könnyű volt változtatni az adatokon, ezért visszamenőleg tudtuk korrigálni az összeillesztésnél (*Lásd: 4.2.3. fejezet*) felmerült pontatlanságokat. A legjobb eredményt a teljes váza magasságát tartalmazó képkivágással értük el.

4.2.2. A fényképek torzítása előzetes számítások alapján

Esetünkben a műtárgy feszíni görbülete és keresztmetszet változása nem túl nagy. Található rajta görbületváltás is. Mivel a képek egy magasságban és egy sávban lettek elkészítve és kivágva, ezért egy torzítási táblázat alapján készítettük el a széthúzást legeneráló scriptet (*Lásd: Mellékletek: Alkalmazott scriptek*).

A *script* (*Lásd: Mellékletek: Alkalmazott scriptek*) folyamatábrája bemutatja a program működését. Kezdeként megadjuk az elkészült fényképek helyét, majd definiáljuk a torzítás mértékét (*Lásd: Torzítási táblázat*). A *script* egy ciklusként minden képen egymást követően elvégzi a torzítást, majd menti az elkészült képeket.

A script működési elve a következő oldalon látható.

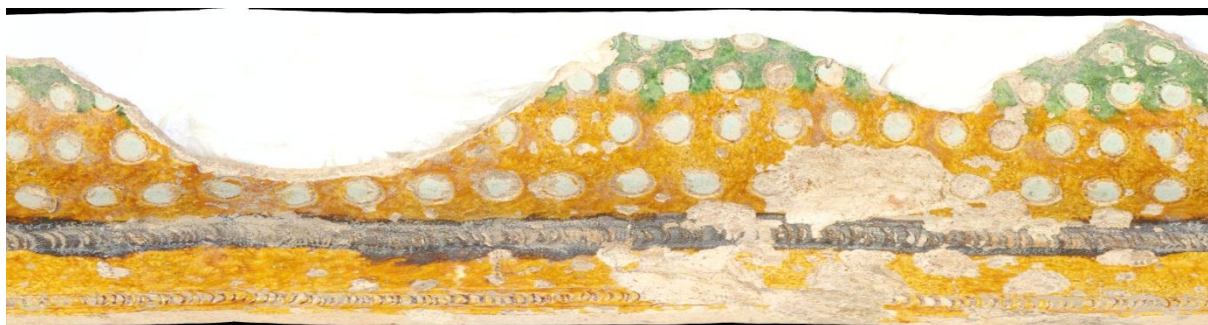
A script működésének elve:*4.18. ábra A torzítást végző script működési elve*

4.2.3. A textúra elkészítése

A textúra elkészítését *Microsoft ICE* szabad felhasználású szoftver segítségével végeztük el. Az előzetesen legyártott, 36 db megfelelően torzított képszeletet a program automatikusan egy *JPG* fájlá illesztette össze.

Az illesztés során több probléma merült fel, amit a korábban alkalmazott technikák (fényképezés, szeletelés, torzítás) során bekövetkezett pontatlanságokból adódtak. Ezen a ponton többszöri visszalépésre volt szükség a megfelelően valósághű textúra létrehozásához. Ezeket a problémákat a fejezetben később fogjuk bemutatni.

Több textúra változat is elkészült, a minél realisztikusabb megjelenítés érdekében, azonban a túlexponált képszeletekből generált kép torzult a legkevésbé (4.19 ábra). Ezt az eredményt tapasztalati úton értük el. Próbálkoztunk a teljes háttér eltávolításával is illesztés előtt, de ez a változat is rosszabb minőségű képet eredményezett (4.20 ábra). A peremrészek megtartása-elhagyása is változtatott az eredményen. A túlexponált teríteken is látható egy enyhe hullám, de ez valószínűleg az egyedileg, manuálisan készített váza felületi pontatlanságaiból adódik.



4.19. ábra Végső textúra változat (túlexponált)

Próbálkoztunk a fehér részek kifestésével, de az eredmény kevésbé lett pontos, mint a túlexponált változat.



4.20. ábra Végső textúra változat (kifestett)

A textúra készítés során a leképezés pontatlanságából adódó hibák:

- Nem lineáris illesztés (4.39 ábra)
Sok esetben, ha a torzítás mértéke túl nagy, keskeny vagy alacsony képcsíkokat használunk a panorámakép készítő program kúppalástot készít a képekből. A háttér jelenléte a képeken tovább ront az összeillesztésen.
- Szín pontatlanságok (4.47 ábra)
Az elkészült textúra variációk közül néhányon megfigyelhetünk színtorzulást, ami az illesztő program pontatlanságából adódik.
- Lineáris illesztés hullámossága (4.40-4.46 ábrák)
Sűrűn díszített vagy mintázott felületeknél a képszeletek annyira torzulnak, hogy az illesztésnél akár több képszelet is fedésbe kerül, ami tovább ront a textúra illesztésén. Hullám keletkezhet a mintában.

A hibás textúrákra a Mellékletek: Textúra változatok pontjában találhatunk példákat.

Szétvágás

A textúra szétvágásához a testmodell egyik fél terének középvonalát használtuk. Ez alapján megkerestük a fényképszeletek közül azt, amelyiknek a középvonala megközelítően megegyezik a fél tér középvonalával (4.21 ábra). A terület középső képszelete: JPG_8446.

A számítások előtt fontos megjegyezni, hogy csak az x koordinátákkal számolunk. Ezt az eljárást és a pixelekkal történő számításokat *GIMP*-ben végeztük el. Először lemértük a szelet középvonalának távolságát a teljes textúra bal szélétől.

A textúra magassága konstans $y=1221$ pixel

Egy szelet szélessége: 200 pixel, aminek a középvonala 100 pixelnél található.

$$x_{k\text{szelet}}=100 \text{ pixel}$$

A teljes textúra szélessége: 4608 pixel.

$$x_{\text{össz}}=4608 \text{ pixel}$$

A ráillesztett szelet középvonala a textúra bal szélétől: 1224 pixelnél található.

$$x_{1\text{közép}}=1224 \text{ pixel}$$



4.21. ábra Az egyik fél textúra középvonalának megkeresése

Az előzőleg kiszámolt értéket referenciának vesszük, és két irányban kiterjesztjük a kijelölést akkorára, ami megegyezik az alkalmazni kívánt fél textúra nagyságával, majd elvégeztük a kivágást.

A kivágás két szélének kiszámolása (pixelszám a bal szélétől):

$$V_1 = x_{1\text{közép}} + x_{\text{féltextk}} = 1224 + 1152 = 2376 \text{ pixel}$$

$$V_2 = x_{1\text{közép}} - x_{\text{féltextk}} = 1224 - 1152 = 72 \text{ pixel}$$

A fél textúra: 2303 pixel, aminek középpontja (x irányban): 1152 pixelnél található.

$$x_{\text{féltext}}=2303 \text{ pixel}$$

$$x_{\text{féltextk}}=1152 \text{ pixel}$$

A másik fél textúra a következő módon áll elő: egy 2232 pixel jobb oldalról ($x_{\text{össz}} - V_1$) és egy keskeny 72 pixel nagyságú szeletből, ami korábban a teljes textúra bal oldaláról került levágásra. Ezt a két megmaradt szeletet, a már korábban alkalmazott Microsoft ICE nevű programban illesztettük össze.

Áttetsző textúra készítése

Mivel a műtárgy külső felülete mázas, azonban az idő során sok helyen lepattogzódott, ezért célszerű a textúránkat is az eredeti modell szerint elkészíteni. Ennek a műveletnek akkor lesz jelentősége, amikor a vizualizáció során meg kell jelenítenünk az eredeti anyagot és annak tulajdonságait, ami eltér a zománcozott felülettől a lepattogzott és belső felületeken.

Először a tárgy körvonala mentén (a perem részeket hagyva) körülvágtuk a képeket, majd ezeken a felülethez pontosított textúrákon végeztük a további változtatásokat.

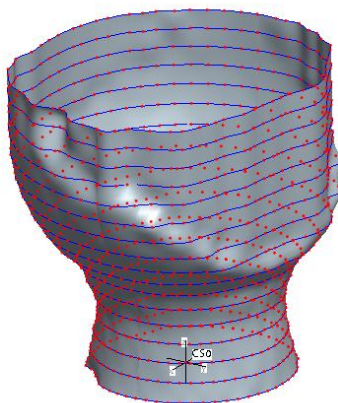
A textúra átlátszóságához szükséges műveleteket *GIMP*-ben végeztük el. Az eljárás lényege, hogy a lepattogzott felületeken kilátszó agyag *RGB* színkódját kimásoltuk a befoltozott képekről és ezt, mint színtartományt, kijelöltük a textúrán. Mivel a kijelölt területek azok, amelyekre nincsen szükségünk és átlátszóak, ezért a kijelölés inverzét külön rétegre másolva mentettük el, majd a háttérbe került teljes textúra áttetszőségét 50%-ra csökkentettük (4.22/b ábra). Ugyanezt a műveletsort elvégeztük a másik textúrán is, majd *PNG* formátumba mentjük őket. A *PNG* alkalmas képfájlok átlátszóságot is tartalmazó háttérrel való tárolására.



4.22. ábra a/A fél textúrák szétvágott változata b/ A fél textúrák félig áttetsző, végső változata

4.3. A MATLAB programban a pontsorok generálása

A pontsorokat generáló program futtatása során a pontok távolságának a végső változatban 3 mm választottunk ($gDens=3$), mivel a manuális módosítás miatt ilyen sűrűségben még elvégezhető a pontsorok felvétele és módosítása. A dolgozat megírása során viszont a 6 mm távolságú pontsorok felhasználásával létrejövő felület modellt is elkészítettük. (4.23. ábra)



4.23. ábra Felületmodell 6 mm pont távolság esetén

A többi változót a műtárgy adatai alapján kell beállítani, vagyis a hengeres felületnek körül kell vennie a vázát, így a váza átmérőjénél nagyobb átmérőt kell a pontsorok számára meghatározni. A váza rekonstruálása során az átmérőt 96 mm-nek választottuk. A pontsorok maximális magasságának definiálásánál is a váza legmagasabb pontja volt a meghatározó. Ezt a értéket 115 mm-nek választottuk. A függvény, mely ezen adatokkal dolgozik g .

$$g = \text{GridGen}(96, 115, gDens) \quad (23)$$

Mivel ezen paramétereket a program elején kell meghatározni, ezért megváltoztatásuk egyszerű, a program gyorsan lefuttatja az esetleges változtatásokat.

A szkennelés pontok behívásánál a váza kör szkennelt 0,2 mm-szer 0,2 mm sűrűségű pontjait hívtuk be. A szkennelés pontok szűrését a *Pro/ENGINEER*-ben végeztük el a *MATLAB* programba való behívás előtt. (Lásd 3.1. 1. 3D-s körszkennelési eljárás.)

A program működését Lásd 3.3.4. Program a felületek generálására és illesztésére

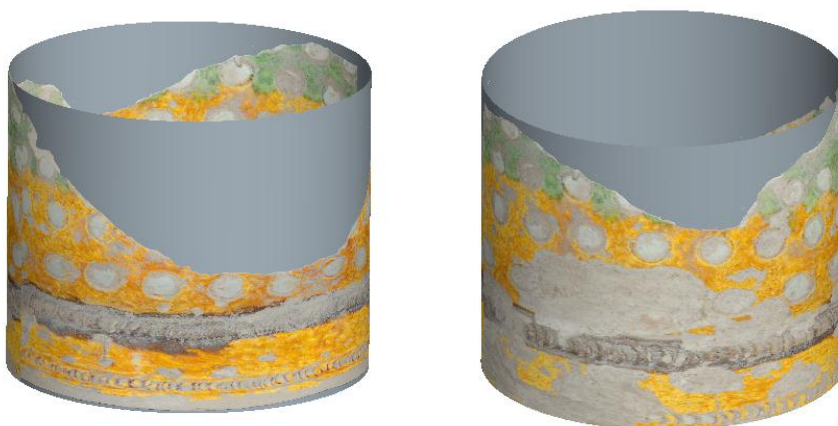
4.4. A műtárgy 3D-s rekonstrukciója a Pro/ENGINEER-ben

A szabályos pontsorok behívásához és módosításához makrókat hoztunk létre, melyek segítségével a manuális tevékenységet gyorsíthatjuk. A módszer végső kidolgozásnál cél az eljárás automatizálása, a dolgozat során viszont még sok esetben nem volt lehetőség kiküszöbölni olyan manuális tevékenységeket, mint a pontsorok egyenkénti behívása, a szplájnok, majd a felületek felvétele, és végül a pontsorok módosítása. További célkitűzésként szerepel ezen műveletek automatizálása.

A hengeres felületet az előzőekben ismertetett módon hozzuk létre. (Lásd 3.4.1.)

A felület részek elkészítése után a textúra illesztése következik. A textúrát két részre kell vágni a két félhenger felület alapján. (Lásd 4.2.3. A textúra elkészítése)

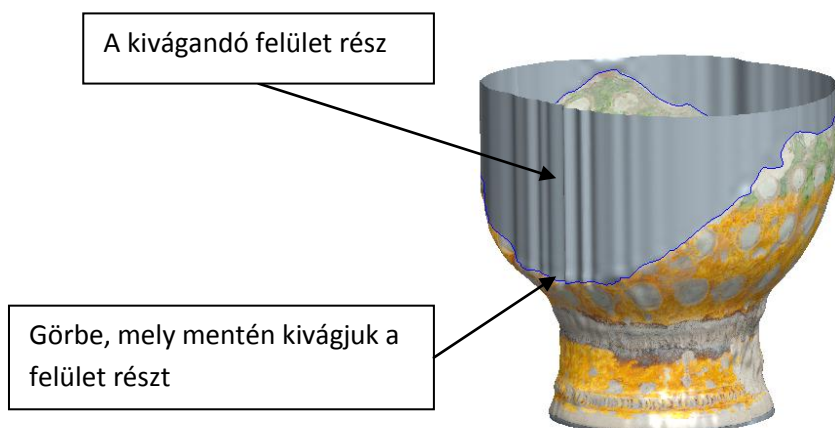
A textúra illesztése hengeres koordináta rendszerben történik, hengeres vetítéssel. A textúra részeket nagyon pontosan kell az adott felület részekhez illeszteni. A feltextúrázott felület.(4.24.ábra).



4.24. ábra A feltextúrázott felület

Ezután elvégezhetjük a pontsorok módosítását.

Ha elkészült a feltextúrázott, módosított felület, akkor ki kell vágni azokat a részeket, melyek már nem a műtárgy részei. A kivágást a textúra és a váza törésvonala segíti, mivel a törésvonal felett a textúra átlátszó, így az általunk felépített felület látszódik. A kivágáshoz görbéket hozunk létre, melyekkel követjük a váza törésvonalát(4.25.ábra), majd ezek mentén kivágjuk a többlet felület. (4.26.ábra)



4.25. ábra A kivágást segítő görbe



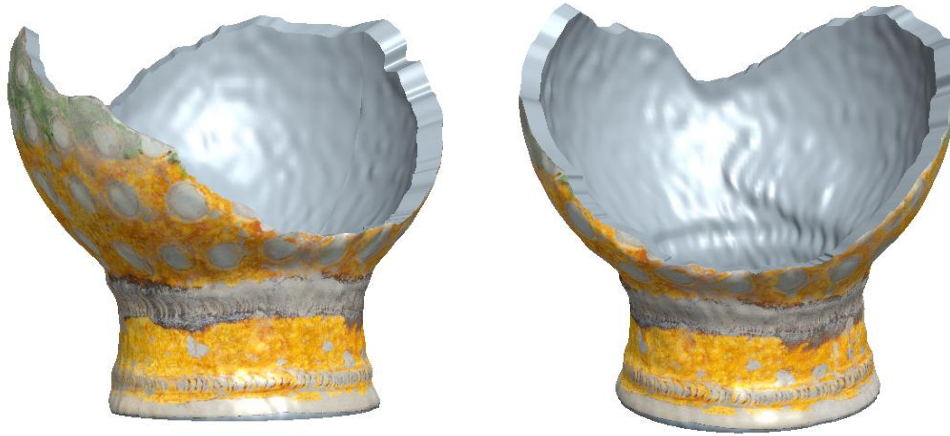
4.26. ábra A feltextúrázott felület a többlet felületek kivágása után

A módszer kidolgozásának ezen fázisában csak a külső felület szkenn pontjait használtuk (Lásd 3. A módszer bemutatása), viszont a tárgy alján nem voltak szkenn pontjaink, így az alsó felületet nekünk kell definiálni, ehhez a legalsó módosított pontsor segítségével zárt, sík felületet hozunk létre. (4.27. ábra)



4.27. ábra Az alsó felület

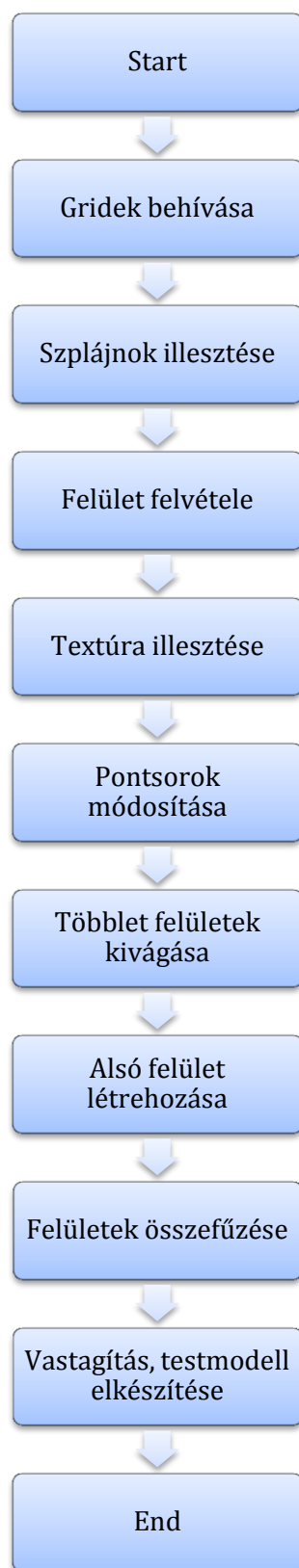
Ha az összes felület rendelkezésünkre áll, a felületrészeket össze kell fűzni, ezután elvégezhető a vastagítás, létrehozva a testmodellt. A vastagítás során felléphetnek kitöltési problémák, de a műtárgyunk esetén ilyen jellegű probléma nem merült fel, a vastagítást elvégezte a Pro/ENGINEER. (4.28.ábra)



4.28. ábra Az elkészült testmodell

Az elkészült testmodellhez ezután anyagot rendelünk és elkezdjük a műtárgy vizualizációját

A műtárgy 3D-s rekonstrukciója



4. 29. ábra A testmodell elkészítésének folyamatábrája

4.5. A műtárgy vizualizációja

A műtárgy vizualizációját a *Pro/ENGINEER* Photolux render programmoduljában végeztük el áttetsző textúra alkalmazásával (4.30 ábra).

Műterem rekonstruálása:

A fényképezés során beállított, majd lemért elrendezés adataival a *Pro/ENGINEER* munkaterében egy vázlat segítségével újra létrehoztuk a stúdió elrendezését (*Lásd: Műterem beállításai*). A műterem háttérének *RGB*(34,7;34,7;34,7) színkódú szürke színt állítottunk, ami nem veri vissza a fényt. A beállításokat rögzítettük a műtárgyhoz, ezért forgatás során a műterem beállításával együtt mozog, és nem kell minden egyes renderelési folyamat előtt visszaállítani.

Fények beállítása:

Az alapértelmezett világítást kikapcsoljuk és az előzetesen kimért és beállított két pontban a *softbox*-okkal megegyező fényforrásokat helyezünk el (*Lásd: Műterem beállításai*). A környezet felől nem érkezik semmilyen megvilágítás. A kamerákat a vázához rögzítettük, ezáltal megakadályozva a beállítás elmozdulását.



4.30. ábra Az áttetsző textúra alkalmazása a modellen

Anyagválasztás:

A programból automatikusan előhívható anyagok között találtunk a kerámia csoportban kőként definiált anyagot, de ennek az állítási lehetőségei nem tették lehetővé a valósághű megjelenítést. Ezért az általánosként definiált anyagtulajdonság beállításával hoztuk létre az agyaghoz hasonló megjelenítést. Itt a következő értékeket állítottuk be:

Intensity/Intenzitás	100
Ambient/Környezet	30
Highlight color/Becsillanás színe	RGB (153;153;153)
Shine/Fényesség	35
Highlight/Becsillanás intenzitása	30
Reflection/Visszaverődés	33
Transparency/Áttetszőség	0

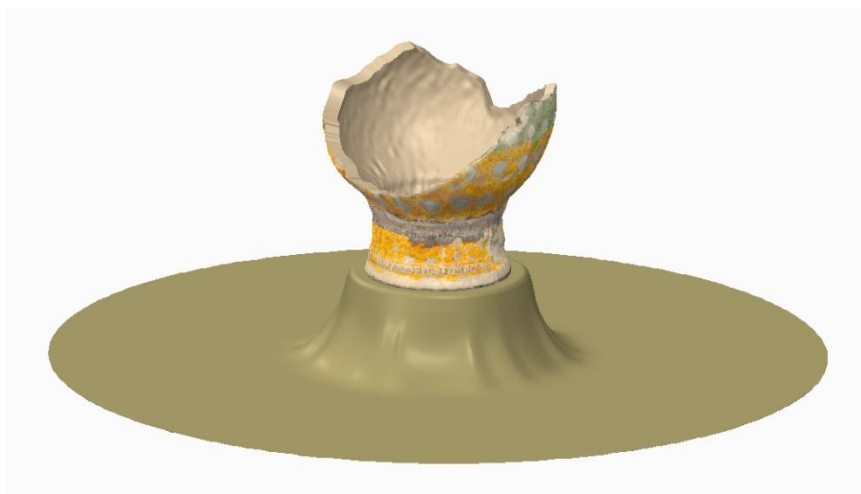
Színválasztás:

A beállított anyag színét a korábban fényképezett képen rögzített agyag színének *RGB* színkódjával megegyezőre állítottuk. (Lásd: 4.1.2. fejezet) (4.34/b ábra) *RGB* (220;196;160).

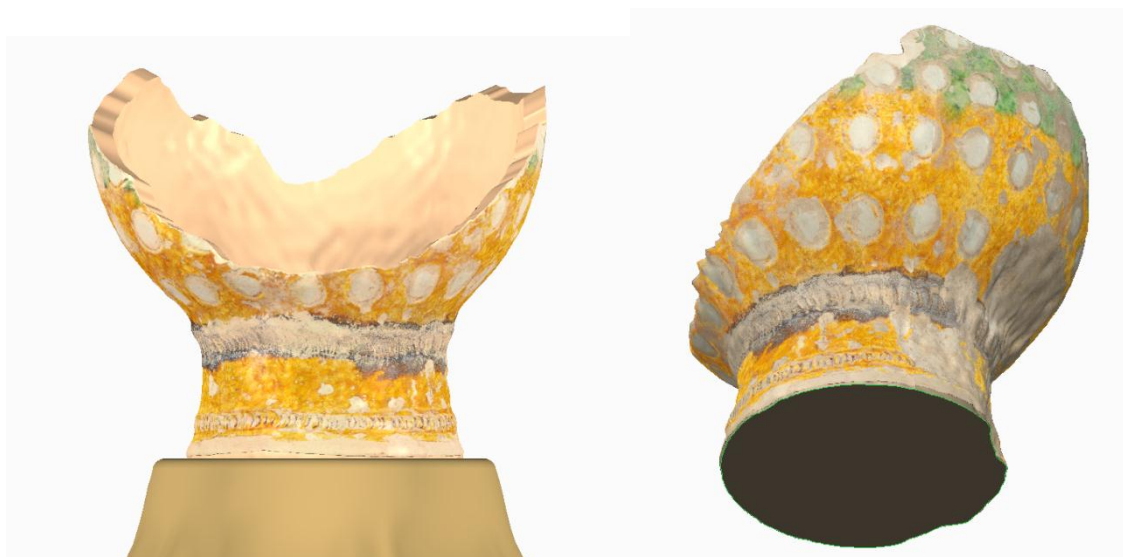
További beállítások:

A próbálkozásainkat vázlat szintű *renderek* készítésével végeztük, majd miután sikerült megtalálni a megfelelő intenzitásokat és a végső elrendezésnél a *render* minőségét maximumra állítottuk. A *render*t sugárkövetéses technológiával végezte a program.

A végső beállításnál, a műtárgyat egy lepellel leterített körasztalra helyezve mutatjuk be(4.31 ábra).



4.31. ábra A végső renderelési beállítás



4.32. ábra a/ A végső textúrázott modell b/ A textúra illeszkedés pontosságát bemutató modell

A vázán a textúrák illesztését pontosan sikerült elvégezni (4.32 ábra), ezért *renderelés* során könnyen be tudtuk mutatni a műtárgyat 360 fokos körbe forgatással.

Különböző beállítások a végső renderelés előkészítésére:



4.33. ábra Különböző renderelési beállítások

Az előző képeken látható (4.33 ábra), hogy hosszas és precíz állításokat igényelt a realisztikus vizualizáció elkészítése. Ha a végső, maximális minőségben renderelt képet összehasonlítjuk az azonos beállításokkal rendelkező műtermi fényképpel észrevehetünk eltéréseket, azonban jól látszik, hogy minden beállításunk helyes volt. Ezt a becslések helyzete és a megközelítően azonos színek igazolják. Az eltérések mérési pontatlanságból adódhatnak. Kiértékelésünk alapján, a két képen (4.34a-b ábra) azonos tartományban megjelenő *RGB* színek kb. 80%-os pontossággal egyeznek.

Végeredményül a váza külső felületének jó közelítéssel azonos mását kaptuk.(4.34a ábra)

Végső renderelt változat:



4.34. ábra a/ A végső renderelt kép b/ A műteremben azonos beállításokkal elkészült fénykép

Összefoglalás

A dolgozatban részletesen ismertetett, majd egy példán alkalmazott eljárást mutattunk be. Megállapítható, hogy a kezdetben elvként végiggondolt módszert a gyakorlatban is sikeresen alkalmaztuk. A folyamat során, az eredetileg eltervezett munkamenetben történtek módosítások és finomítások, de végül egy látványos és realisztikus vizualizációt sikerült előállítani. A TDK dolgozat célkitűzésének eleget tettünk és további fejlesztési lehetőségekre derült fény, ami további finomításokat jelenthet az eljárás menetében a tárgy minél valósághűbb reprezentálása érdekében.

A kutatás eredményei

A kutatásunk eredménye a kiválasztott műtárgy valósághű reprezentálása egy új módszer segítségével. Mint ahogy azt a szakirodalom áttekintése során is láthattuk a felületek rekonstruálása és a valósághű megjelenítése egy olyan probléma melyeknek még nincs tökéletesen kidolgozott, egzakt módja, így az általunk kialakított eljárás lehetőséget ad ezen két probléma együttes, komplex kezelésére, mivel a módszer a tárgy rekonstruálását és vizuális megjelenítését együtt kezeli. Az általunk kiválasztott program (Pro/ENGINEER) lehetőséget ad ezen műveletek elvégzésére, bár eddig ilyen jellegű műveletekre ezt a rendszert jellemzően nem használták.

A véglegesen kidolgozott eljárással bármilyen tárgy virtuális rekonstruálása lehetséges lesz. Eddigi kutatásunkkal egy integrált módszer kialakításának első fázisát dolgoztuk ki, ami már ilyen módon is számos helyen felhasználható.

Az eredmények hasznosítása

A módszer kidolgozásának sok tekintetben van létjogosultsága, mivel a felhasználási lehetősége nagyon sokrétű.

Egy virtuális testmodell segítségével hiányos műtárgyak pótlása lehetséges. A dolgozatunk kidolgozása során felvettük a kapcsolatot a Magyar Iparművészeti Múzeummal, akik érdeklődtek a módszer iránt, mivel számukra sok esetben problémát jelet egy-egy tárgy restaurálása, viszont egy virtuális testmodell sok esetben segítené a munkájukat. Törött és hibás műtárgyak pótlására is alkalmas az elkészült virtuális testmodell.

A modellt a műtárgy eredeti környezetéhez is illeszthetjük, így készítve róla képeket, mely segítségével egy múzeumban a látogatók el tudják képzelni az adott tárgy eredeti felhasználási módját, így hozva közelebb az emberekhez a történelmet.

Egy másik felhasználási lehetőség lenne, hogyha már rendelkezünk a tárgyról 3D-s modellel, ami térben mozgatható, körbejárható és pontos mása az eredeti tárgynak, akkor ezekből a modellekből a múzeumok virtuális tárlatot is készíthetnek, ahol az összes lelet a múzeumban elfoglalt helyén, a tényleges múzeumi környezetben található. Ez a felhasználási lehetőség napjainkban egyre fontosabb, mivel a múzeumok is igyekeznek kihasználni a technika fejlődését. Erre jó lehetőséget ad egy virtuális tárlat.

Egy további lehetséges felhasználási mód, az elkészült számítógépes modell alapján a leletet gyors prototípusgyártással való újra nyomatása, így az bárki számára hozzáférhetővé válna, és lehetőség nyílna külső beavatkozás nélküli, roncsolás mentes vizsgálatokra régi műtárgyakon. Ennek az eljárásnak a lényege, hogy a testmodellt hagyományos megmunkálásoktól eltérően, ahol a modell lebontó eljárások segítségével készül el, itt egy 3D nyomató segítségével polimer porból rétegről rétegre anyag hozzáadás útján alakítjuk ki a kívánt geometriát. Az elkészült modell pontos mása az eredetinek. Az újra nyomtatás előnye, hogy a polimer modell kézbe vehető, műtárgyak esetén akár a látogatók által is, mivel nem kell az esteleges sérüléstől, károsodástól tartani. A leletekből így olyan tárlat is készíthető, melynek lényege, hogy a látogató kézbe veheti az adott tárgyat, így lehetőség nyílik látássérültek számára megrendezésre kerülő kiállításra is. Másik fontos előnye a másolatoknak, hogy olyan leletekből is lehetne tárlatot készíteni, amik valamilyen módon fokozott védelmet igényelnek, és emiatt nem kiállíthatóak, például fényérzékenyek vagy rendkívül értékesek.

Az eddigi felhasználási lehetőségek a műtárgy rekonstrukció körével foglalkoztak, viszont a kidolgozott módszer számos más esetben is használható. Az iparban sok esetben lenne szükség alkatrészek szkennelés utáni testmodelljére, mely a teljes módszerrel elvégezhető lenne.

További célkitűzések

Mivel az eljárás során a következő egyszerűsítéseket alkalmaztuk, emiatt a továbbiakban célkitűzésink között szerepel ezen egyszerűsítések kiküszöbölése.

Egyszerűsítések

- hengersizmetrikus tárggyal dolgozunk, hengeres felületet veszünk fel kezdésként
- csak külső felülettel dolgozunk, elhanyagoljuk a belső felületeket és a perem részt
- a MATLAB programban a megfelelő szken pontok kiválasztása, csak a legnagyobb x koordinátájút választjuk, ez nem minden esetben a legjobb
- a végső modellen a pontok távolsága 3 mm, ez lehetne kisebb, úgy még pontosabb eredményt kapnánk (de a manuális módosítás miatt 3 mm választottunk)
- a textúra illesztés (kézzel mozgatjuk a megfelelő pozícióba)
- szplájnok hibája a félterek csatlakoztatásánál, illetve a törésvonal követésénél
- vastagítás esetén a belső oldal képzése
- a műtermi fényképezés során a forgatást, 10 fokként végeztük el, pontosabb eredményt kapnánk, ha sűrűbben készítjük a felvételeket

További célkitűzés

- ne csak hengeres tárgyra legyen alkalmas, szoborszerű felületekre is
- belső és perem felületek létrehozása is
- a MATLAB programban a megfelelő szken pontok kiválasztásának pontosítása
- a Pro/ENGINEER-ben a pontok, szplájnok felvétele, automatikus módosítása
- textúra illesztés pontosítása
- felesleges felületrészek kivágásának pontosítása
- ha külső-belső felülettel dolgoznánk akkor csak ki kellene tölteni a felületek közötti részt, nem állnának fent a vastagítás esetleges problémái
- felületre normális irányban történő fényképezés és ennek további torzítása
- a fényképezés utáni korrekciók csökkentése

Ábrajegyzék

3.1. ábra A módszer folyamatábrája.....	14
3.2. ábra A körszkennelési eljárás.....	15
3.4. ábra A vizsgált terület.....	23
3.5. ábra A megfigyelő koordináta rendszer.....	24
3.6. ábra A kiválasztott pont.....	26
3.7. ábra Az adott szinten kiválasztott pontok.....	27
3.8. ábra A lefutott program által kirajzolt kép.....	27
3.9. ábra A program folyamatábrája.....	29
3.10. ábra a) A generált pontsorok; b) A pontsorokra felvett szplájnok.....	30
3.11. ábra a) Félhengeres felület; b) A teljes henger felület.....	31
4.12. ábra a/ A 3D szkennер körszkennelési beállításai b/ a 3D szkennер által letapogatott pontfelhő.....	33
4.13. ábra A műtárgy fényképezése 10 fokenkénti körbeforgatással.....	34
4.14. ábra Az abszolút szürkét megjelenítő színkártya (képkorrekciókhoz szükséges).....	35
4.15. ábra A fényképeken értelmezett abszolút függőleges referencia.....	35
4.16. ábra a/ Torzítatlan képszelet b/ Torzítás utáni megfelelően szétnyújtott képszelet.....	36
4.17. ábra A kivágást végző script működési elve.....	37
4.18. ábra A torzítást végző script működési elve.....	39
4.19. ábra Végző textúra változat (túlexponált).....	40
4.20. ábra Végző textúra változat (kifestett).....	41
4.21. ábra Az egyik fél textúra középvonalának megkeresése.....	42
4.22. ábra a/ A fél textúrák szétvágott változata b/ A fél textúrák félig áttetsző, végző változata.....	43
4.23. ábra Felületmodell 6 mm pont távolság esetén.....	44
4.24. ábra A feltextúrázott felület.....	45
4.25. ábra A kivágást segítő görbe.....	46
4.26. ábra A feltextúrázott felület a többlet felületek kivágása után.....	46
4.27. ábra Az alsó felület.....	46
4.28. ábra Az elkészült testmodell.....	47
4.29. ábra A testmodell elkészítésének folyamatábrája.....	48
4.30. ábra Az áttetsző textúra alkalmazása a modellen.....	49
4.31. ábra A végző renderelési beállítás.....	50
4.32. ábra a/ A végző textúrázott modell b/ A textúra illeszkedés pontosságát bemutató modell.....	51
4.33. ábra Különböző renderelési beállítások.....	51
4.34. ábra a/ A végző renderelt kép b/ A műteremben azonos beállításokkal elkészült fénykép.....	52
4.35. ábra: A váza.....	59
4.36. ábra A ROLAND LPX 250 3D laser szkennер.....	61
4.37. ábra A műterem beállításai (mm).....	62
4.38. ábra A stúdióban használt soft boxok.....	63
4.39. ábra Torzítási táblázat.....	66
4.40. ábra Nem lineáris illesztés.....	67
4.41. ábra Lineáris illesztés pontatlansága (felső képcsík).....	67

4.42. ábra Lineáris illesztés pontatlansága (felső képcsík).....	67
4.43. ábra Lineáris illesztés pontatlansága (képcsík).....	67
4.44. ábra Lineáris illesztés pontatlansága (képcsík).....	67
4.45. ábra Lineáris illesztés pontatlansága (képcsík).....	68
4.46. ábra Lineáris illesztés pontatlansága (képcsík).....	68
4.47. ábra Lineáris illesztés pontatlansága (alsó képcsík).....	68
4.48. ábra Szinpontatlanság.....	68

Irodalomjegyzék

- [1] Montegranario, H., Espinosa, J., C.-A. (2007): **A regularization approach for surface reconstruction from point clouds**, Applied Mathematics and Computation 188 (2007), ELSEVIER, pp.: 583–595, ISSN: 0096-3003
- [2] Liu, S.; Wang, C. C. L. (2010): **Orienting unorganized points for surface reconstruction**, Computers & Graphics 34 (2010), ELSEVIER, pp.:209–218, ISSN: 0097-8493
- [3] Chang, C.M.; Leymarie, L.L.; Kimia, B. B. (2009): **Surface reconstruction from point clouds by transforming the medial scaffold**, Computer Vision and Image Understanding 113 (2009), ELSEVIER, pp.: 1130–1146, ISSN:1077-3142
- [4] Alberts, C. A. (2004): **Surface reconstruction from scan paths**, Future Generation Computer Systems 20 (2004), ELSEVIER, pp.: 1285–1298, ISSN: 0167-739X [2]
- [5] Dékán, I. (2008): **Digitális fényképezés alapoktól a szakmai ismeretekig**, Zagora 2000 Kiadó, ISBN:9789630633192
- [6] Shawn Becker , V. Michael Bove , Jr.: **Semiautomatic 3-D model extraction from uncalibrated 2-D camera views** SPIE Symposium on Electronic Imaging konferencia, San Jose, California, USA, 1995. február, pp.:447-461
- [7] Schindler, K.; Kager, H.: **Concept of a system for the computer-aided reconstruction of antique monuments** International CIPA Symposium konferencia, Olinda, Brazília, 1999. október 3-6.
- [8] S. Sotoodeh, A. Gruen, T. Hanusch: **Integration of structured light and digital camera image data for the 3d reconstruction of an ancient globe** ISPRS Congress Beijing 2008 konferencia, Peking, Kína, 2008. július 3-11., pp.: 367-372
- [9] Petrich, G.; Lőrincz P. (1981): **Ábrázoló geometria** Nemzeti Tankönyvkiadó Zrt., Budapest, ISBN: 9789631949759

Melléklet

A műtárgy adatai

A kisméretű váza, metéltmázás technikával készült, fehéragyagból. A metéltmázás technika a 15. sz. közepén jelenik meg Itáliában, Magyarországon 1490-től ismerünk ilyen díszítésű díszkerámiát Budáról. A mintát a puha agyagba karcolják és a mezőket különböző színű mázakkal töltik ki (általában barna, zöld, sárga színt használtak). Valószínűleg a köznemesség és a polgárság körében lehetett elterjedt a használatuk, sokáig megőrizhették, előkelő pohárszékekben tarthatták őket. Főleg folyadéktároló edényeket (korsó, kancsó) és tálakat készítettek ezzel a technikával. Ezek mindegyike egyedi darab volt, nem mindennap használták őket. Az általunk felhasznált váza 17. század második feléből származik!

A váza magassága: 96 mm

Átmérője: 115 mm



4.35. ábra: A váza

Ásatás jelentése

Szécsény, Haynald u. 7.

(Nógrád megye) Á, KK, Kú

2011. június 27. – július 15. ill. augusztus 1-18. között próba- és megelőző feltárást végeztünk Szécsény – Haynald utca 7. szám alatt, ahol a Ferences Szegénygondozó Nővérek egy meglévő épület bővítésével öregek otthonát és kápolnát kívánnak építeni.

Az építkezésre védett középkori lelőhelyen (33000, Szécsény – Középkori városmag lh.), Szécsény történelmi városmagjában, a 14. századi eredetű ferences kolostor, a középkori vár és a piactér közti területen került sor.

A területet az új épület alapozási árkaik helyén, összesen nyolc szelvényben kutattuk meg. Eltekintve az épület közvetlen közelében lévő modern csatornák és a terület keleti felében jelentkező újkori épület alapozási árkaik bolygatásától, mindenütt jelentős régészeti rétegeket figyeltünk meg, már a jelenlegi szinttől mért 40-50 cm-es mélységtől kezdődően. A 17-18. századi leletekben rendkívül gazdag, 18. század közepére keltezhető planírozási réteg alatt az egész területen megtaláltuk a 17. század végi város pusztulási szintjét. Ebből különböző helyeken eltérő típusú jelenségeket dokumentáltunk, így előkerült többek között egy gerendafödemes pince, egy meszelt falú tárolóverem, paticsfalú ház(ak) deszkaburkolatos padlója, rajta gabonamagvakat is tartalmazó réteggel, felette pedig vastag, égett, paticsos omladékkal. Mindezekhez rendkívüli minőségű leletanyag tartozott: évszámos habán edények, török csészék töredékei (az egyiken arab felirat is olvasható), reneszánsz, metéltmázás díszítésű tál, 16. századi díszkerámia, rézedények töredékei, és egy angliai textilplomba. A tároló verem fedésének és a ház padlójának a fái olyan jó állapotban maradtak meg, hogy lehetőség nyílt dendrokronológiai vizsgálatokra. Grynaeus András eredményei szerint a verem és a ház egykorú, s építésükre legkorábban 1679-ben került sor. Ugyanakkor a pusztulási réteg a város 1683-as visszafoglalásához köthető.

Az egyik legfontosabb megfigyelésnek tekinthető egy 16. század második felére – 1600 körülre keltezhető habarcsos, köves járószint a 17. századi elpusztult épületek alatt. Ez a középkori ferences kolostor telkét kelet felől határoló keskeny, síkatorszerű utcának a nyomvonala lehet, amely a középkorban is létező, ma Haynaldról elnevezett utcát kötötte össze a város piacterével.

Néhány objektum betöltésében másodlagos helyzetű késő középkori és Árpád-kori kerámiatöredékek is előkerültek. Ehhez a két korszakhoz tartozó ép jelenség nem került elő, ezek mindegyikét a kora újkori verem és pincék pusztíthatták el. Azonban a feltárás által nem érintett és a kora újkorban nem beépített területen ilyenek előfordulhatnak.

A feltárásban segítségünkre voltak az ELTE-BTK Régészeti Intézet hallgatói: Szalai Emese, Varga Emese és Vargha Mária, valamint a Kubinyi Ferenc Múzeum munkatársa: Guba Szilvia. Munkájukat ezúton is köszönjük.

Mordovin Maxim – Zandler Krisztián

3D szkennerek adatai

Tapintás nélküli technológiát képviselő berendezés. A lézeres fejjel rendelkező egységek mind CLASS II osztályú sugárforrást kaptak, ami nem ártalmas az emberi szemre.

ROLAND LPX 250 3D laser szkennerek.

Háromdimenziós tárgyak szkennelésére alkalmas, a digitális modellt STL vagy DXF (poligon) és IGES (NURBs felület) fájlformátumra képes alakítani. Maga a berendezés a (4.35. ábrán) látható, nyitott burkolattal, ahol látható a munkatér a forgóasztal.



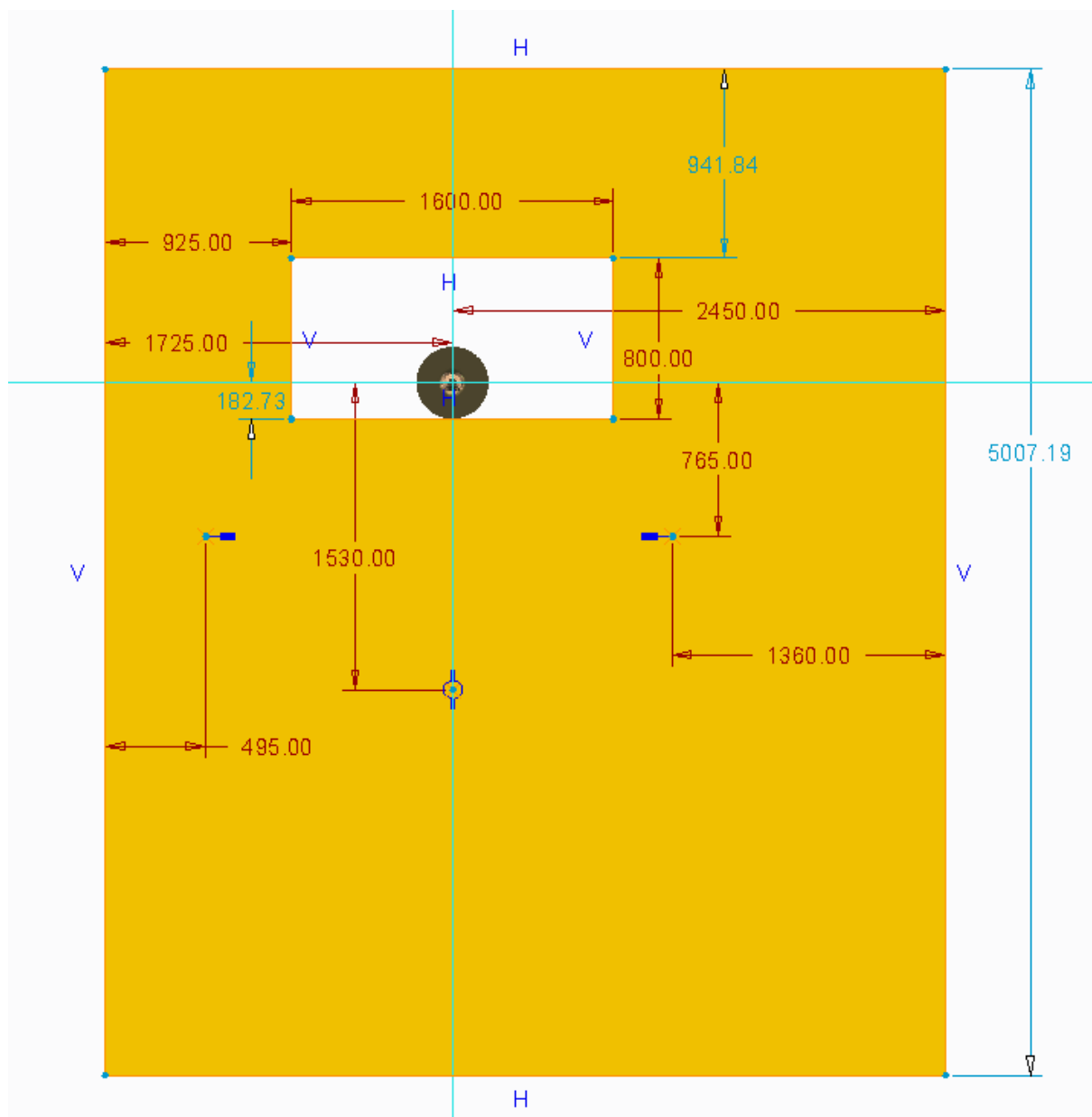
4.36. ábra A ROLAND LPX 250 3D laser szkennerek

A szkennerek jellemzői röviden:

- Érintés nélküli, 3D lézer szkennelés
- Átalakítja a beszkennt adatokat poligon és NURBs felületre
- Importálható STL, DXF és IGES fájl formátum az ipari standard 3D és modellező szoftverekhez
- Nagy szkennelési terület - max. 406.4 mm x 254 mm
- Dual mód -- sík vagy kör szkennelés az optimális teljesítmény érdekében

A kör- és sík szkennelési mód lehetővé teszi, hogy a különböző formákat a leghatékonyabban lehessen beszkenntelni. Ezáltal bármilyen objektum szkennelése elvégezhető a sima felületű testektől a bonyolultabb modellekig. A felbontást a felhasználó állíthatja be, maximum 0.2mm-ig.

Műterem adatai



4.37. ábra A műterem beállításai (mm)

Műterem beállításai:

Az ábrán a műteremben elhelyezett eszközök helyét mutatjuk be a műtárgy helyzetéhez méretezve milliméterben.

A fotózás során világításnak *softbox*okat alkalmaztunk. A lámpák adatai a következők:



4.38. ábra A stúdióban használt soft boxok

- 11,5-11,6 ISO-100
- 11-11,2 ISO-100

A műtermi fényképezés nem igényel speciális felszerelést. Szórt és egyenletes világítással dolgoztunk, állandó színhőmérséklettel, és így a színek is hitelesen visszaadhatóak. Amennyiben megfelelően szórt a fény, akkor a tárgy felületi jellegzetességei, hibái (törés, lepattant részek) se vetnek árnyékot, nem okoznak különösebb torzulást.

Alkalmazott scriptek

Kivágás

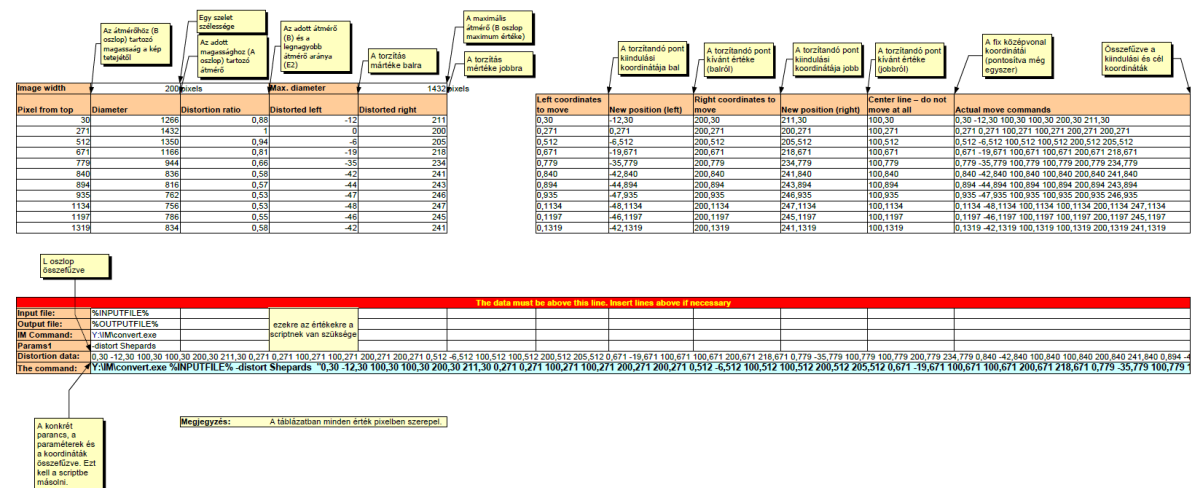
```
@echo off
setlocal EnableDelayedExpansion
REM *****
REM ***          SETUP PART          **
REM *****
REM Set image source directory
set SourceDir=Y:\HarmadikVEGLEGES\converted
REM Object centerine X coordinate (pixel)
set CenterXC=1713
set SliceWidth=200
set /A SliceLeft=%CenterXC% - %SliceWidth% / 2
set PADTop=686
set Padding=0
REM Set Slice heights
set SliceHArray[1]=1330
set IMDir=Y:\IM
REM *****
REM ***          END SETUP PART          **
REM *****
for %%a in (%SourceDir%\*.jpg) DO call :MakeSlices %%a
goto Cleanup
:MakeSlices
echo Processing image %1
for /f "usebackq delims==[]" tokens=1-3" %%i IN (`set SliceHArray`) DO call
:MakeOneSlice %%1 %%j %%k
goto Fin
:MakeOneSlice
rem Params: File Step Height
rem Where the File is the file to process
rem      Step is number of slice to create
rem      Height is the image height
rem Set it to zero
set SliceHeightDone=0
set /A StepsDone=%2 - 1
rem Compute how much pixels already cropped
FOR /L %%K IN (1,1,%StepsDone%) DO set /A SliceHeightDone=!SliceHeightDone! +
!SliceHArray[%%K]!
set /A SliceTop=%PadTOP% + %SliceHeightDone%
set CornerLeftTopX=%SliceWidth%
set /A CornerLeftTopY=!SliceHArray[%2]! - %Padding%
set CornerRightBottomX=%SliceLeft%
set /A CornerRightBottomY=%SliceTop% + %Padding%
```

```
%IMDir%\convert -crop
%CornerLeftTopX%x!SliceHArray[%2]!+%CornerRightBottomX%+%CornerRightBottomY% %1 %1-%2.tiff
goto Fin
:Cleanup
echo Doing cleanup
for /f "usebackq delims==[] tokens=1-3" %%i IN (`set SliceHArray`) DO rmdir /S /Q
%SourceDir%\%%j
for /f "usebackq delims==[] tokens=1-3" %%i IN (`set SliceHArray`) DO mkdir
%SourceDir%\%%j
for /f "usebackq delims==[] tokens=1-3" %%i IN (`set SliceHArray`) DO move
%SourceDir%\*-%%j.tiff %SourceDir%\%%j > nul
:Fin
```

Torzítás

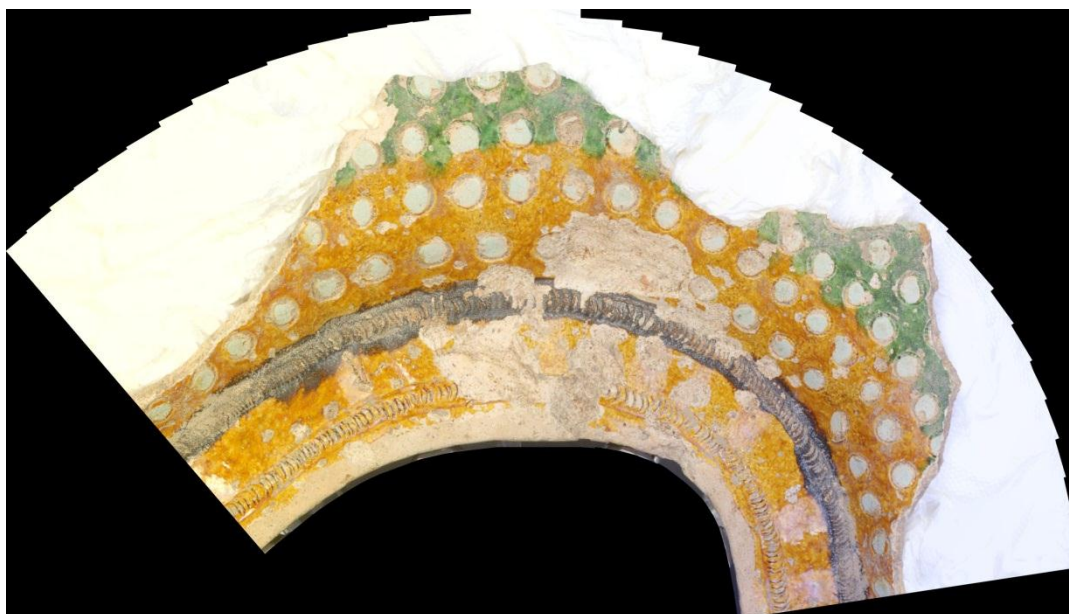
```
@echo off
goto MakeLoop
:DoConvert
set INPUTFILE=%1
set OUTPUTFILE=%1-dist.tiff
rem **** REPLACE THE LINE CALCULATED BY THE EXCEL SHEET
Y:\IM\convert.exe %INPUTFILE% -distort Shepards "0,30 -12,30 100,30 100,30 200,30
211,30 0,271 0,271 100,271 100,271 200,271 200,271 0,512 -6,512 100,512 100,512
200,512 205,512 0,671 -19,671 100,671 100,671 200,671 218,671 0,779 -35,779
100,779 100,779 200,779 234,779 0,840 -42,840 100,840 100,840 200,840 241,840
0,894 -44,894 100,894 100,894 200,894 243,894 0,935 -47,935 100,935 100,935
200,935 246,935 0,1134 -48,1134 100,1134 100,1134 200,1134 247,1134 0,1197 -
46,1197 100,1197 100,1197 200,1197 245,1197 0,1319 -42,1319 100,1319 100,1319
200,1319 241,1319" %OUTPUTFILE%
goto Fin
:MakeLoop
for %%f in (Y:\HarmadikVEGLEGES\converted\1\*.tif) do call :DoConvert %%f
:Fin
```

Torzítási táblázat



4.39. ábra Torzítási táblázat

Textúra változatok



4.40. ábra Nem lineáris illesztés



4.41. ábra Lineáris illesztés pontatlansága (felső képcsík)



4.42. ábra Lineáris illesztés pontatlansága (felső képcsík)



4.43. ábra Lineáris illesztés pontatlansága (képcsík)



4.44. ábra Lineáris illesztés pontatlansága (képcsík)



4.45. ábra Lineáris illesztés pontatlansága (képcsík)



4.46. ábra Lineáris illesztés pontatlansága (képcsík)



4.47. ábra Lineáris illesztés pontatlansága (alsó képcsík)



4.48. ábra Szinpontatlanság

MATLAB programkód

Surface generation and approximation

```

clc;
clear;

HALT=0;
% Grid generation

gDens=3;

g=GridGen(96,115,gDens);
zMin=1;
zMax=size(g,1);

for i=zMin:zMax
    iGrid(:,:)=g(i,:);
    if iGrid(1,3)<100
        filename=['grid\grid_Rz_0',num2str(iGrid(1,3)),'.pts';...
            'grid\grid_Lz_0',num2str(iGrid(1,3)),'.pts'];
    else
        filename=['grid\grid_Rz_',num2str(iGrid(1,3)),'.pts';...
            'grid\grid_Lz_',num2str(iGrid(1,3)),'.pts'];
    end
    if iGrid(1,3)==0
        filename=['grid\grid_Rz_00',num2str(iGrid(1,3)),'.pts';...
            'grid\grid_Lz_00',num2str(iGrid(1,3)),'.pts'];
    end

    for j=1:2
        if j==1
            rGrid=iGrid(1:size(iGrid,1)/2+1,:);

[rwGrid(:,1),rwGrid(:,2),rwGrid(:,3)]=pol2cart(rGrid(:,2)*pi/180,rGrid(:,1),rGrid(:,3));
            save(filename(j,:), 'rwGrid', '-ASCII');
        else
            lGrid=iGrid(size(iGrid,1)/2+1:size(iGrid,1),:);
            lGrid(size(iGrid,1)/2+1,:)=iGrid(1,:);

[lwGrid(:,1),lwGrid(:,2),lwGrid(:,3)]=pol2cart(lGrid(:,2)*pi/180,lGrid(:,1),lGrid(:,3));
            save(filename(j,:), 'lwGrid', '-ASCII');
        end
    end
end

nCS=size(g,2);

```

```

%%
% Preparation for the visulaization

viewAngles=[0,90];
pointColorBase=[...
    1 0 1;...
    0 0 1;...
    1 0 0;...
    0 1 0];

markerTypeBase=[...
    'o';...
    '+';...
    '.';...
    'd'];

% Create the long enough color matrix for the observer's CS and selectes points
pointColor=pointColorBase;
markerType=markerTypeBase;
for i=1:ceil(nCS/4)
    pointColor=vertcat(pointColor,pointColorBase);
    markerType=vertcat(markerType,markerTypeBase);
end

figure1 = figure('Position',[300 50 532*1.2 500*1.2]);
axes1 = axes('Parent',figure1,'YTick',(-70:10:70),...
    'XTick',(-70:10:70));
xlim(axes1,[-70 70]);
ylim(axes1,[-70 70]);
view(axes1,viewAngles);
grid(axes1,'on');
hold(axes1,'all');
set(axes1,'FontName','Arial','FontWeight','demi','FontSize',12);

% O1 original CS (CS=Coordinate System) origin,
O1=[0 0]';

plot(O1(1),O1(2),'O','LineWidth',2,'Color','k');
line([0,5,0,0],[0,0,0,5],'LineWidth',2,'Color','k');
plot(5,0,'*', 'LineWidth',2,'MarkerSize',7,'Color','k');

hold on;
%%
% Grid point processing

nearPoints=zeros(nCS,3);

```

```

for j=zMin:zMax

zLevel=g(j,1,3);

% Szken data handling
%%
clear s ss;
ss=load('szken_torles2.pts', '-ascii');
s=sortrows(ss,3);
%%
% Finds points the actual Zlevel±Dens/2 Z value
idxMin=find(s(:,3)>=zLevel-gDens/2,1,'first');
idxMax=find(s(:,3)>=zLevel+gDens/2,1,'first');
t=s(idxMin:idxMax-1,:);

nP=size(t,1);
homogenCoords=ones(1,nP);
Points=vertcat(t(:,1)',t(:,2)',homogenCoords,t(:,3)');

% Draw the selected szken points (at the selected Z level) in the figure1
plot(Points(1,:),Points(2,:),',' , 'MarkerSize',7,'LineWidth',2,'Color','c');

% Nearest pont selection

radius=g(1,1);

for i=1:nCS

angle=360/nCS*i;

theta = angle * pi/180;
[dx,dy]=pol2cart(theta,radius);
dist=[dx dy]';

% Coordinate transformation from the observer's to the original CS
CoordTrans1=[cos(theta) -sin(theta) dist(1);
             sin(theta)  cos(theta) dist(2);
             0          0      1];

% Coordinate transformation from origin to observer's CS
CoordRot=[cos(theta)  sin(theta) 0;
          -sin(theta) cos(theta) 0;
          0          0      1];

CoordMove=[1 0 -dist(1);
           0 1 -dist(2);
           0 0  1];

CoordTrans2=CoordRot*CoordMove;

```

```

% O2,AX2,AY2: observer's CS origin and axes
O2=O1+dist;
AX2=CoordTrans1*[3 0 1]';
AY2=CoordTrans1*[0 3 1]';
A2yDist=CoordTrans1*[0 tan(2*pi/nCS/2)*radius 1]';

plot(A2yDist(1),A2yDist(2),'','LineWidth',2,'MarkerSize',7,'Color','k');

line([O2(1),AX2(1),O2(1),AY2(1)],[O2(2),AX2(2),O2(2),AY2(2)],'LineWidth',2,'Color',
pointColor(i,:));
plot(AX2(1),AX2(2),'*','LineWidth',2,'MarkerSize',7,'Color',pointColor(i,:));

% Transformation of the whole point array from the original CS to the observer's CS:
P2xyTrans=CoordTrans2*Points(1:3,:);
P2zTrans=Points(4,:)-zLevel;
P2=vertcat(P2xyTrans,P2zTrans);

P2Ysorted=(sortrows(P2',-2));

idxMax=find(P2Ysorted(:,2)<=1*gDens/2,1,'first');
idxMin=find(P2Ysorted(:,2)>=-1*gDens/2,1,'last');
P2Yselected=P2Ysorted(idxMax:idxMin,:);

P2Xsorted=(sortrows(P2Yselected,-1));
idxP2Xsorted=find(P2Xsorted(:,1)>=-1*radius,1,'last');
P2XYselected=P2Xsorted(1:idxP2Xsorted,:);

% Find the closest point to the x axis:
nearP2=sqrt(P2XYselected(:,2).^2+P2XYselected(:,4).^2);
[nearVal,nearIDX]=min(nearP2);
P2near=P2XYselected(nearIDX,1:3);

P2selected=CoordTrans1*P2near';

if isempty(idxP2Xsorted)
    nearPoints(i,3)=zLevel;
else
    nearPoints(i,1)=P2selected(1,1);
    nearPoints(i,2)=P2selected(2,1);
    nearPoints(i,3)=zLevel;
end

plot(nearPoints(i,1),nearPoints(i,2),markerType(j),'MarkerSize',7,'LineWidth',2,'Color',
pointColor(i,:));

% HALT=1 stops the program running, use CTRL+C to exit
if HALT
    pause;

```

```
end

end

if zLevel<100
    filename=['grid\mod_Rz_0',num2str(zLevel),'.pts';...
            'grid\mod_Lz_0',num2str(zLevel),'.pts'];
else
    filename=['grid\mod_Rz_',num2str(zLevel),'.pts';...
            'grid\mod_Lz_',num2str(zLevel),'.pts'];
end
if zLevel==0
    filename=['grid\mod_Rz_00',num2str(zLevel),'.pts';...
            'grid\mod_Lz_00',num2str(zLevel),'.pts'];
end

% Write the points matrix into the 'filename' file in text format ('-ASCII')
for j=1:2
    if j==1
        RnearPoints=nearPoints(1:size(nearPoints,1)/2+1,:);
        save(filename(j,:), 'RnearPoints', '-ASCII');
    else
        LnearPoints=nearPoints(size(nearPoints,1)/2+1:size(nearPoints,1),:);
        LnearPoints(size(nearPoints,1)/2+1,:)=nearPoints(1,:);
        save(filename(j,:), 'LnearPoints', '-ASCII');
    end
end

end
```