

SZAKDOLGOZAT

Szikszai László

Debrecen

2010

Debreceni Egyetem
Informatika kar

TETSZŐLEGES DIGITÁLIS FOTÓZÁSSAL KAPCSOLATOS
PROBLÉMA FELDOLGOZÁSA –
3D MODELLEZÉS / 3D SZKENNELÉS

Témavezető:
Dr. Tornai Róbert
egyetemi adjunktus

Készítette:
Szikszai László
programtervező-informatikus

Debrecen
2010

Tartalomjegyzék

Bevezetés	1
A digitális fényképezés és alkalmazásai.....	2
Történet.....	2
Működési elv és alapfogalmak.....	3
Alkalmazás.....	5
A modellezésről.....	6
Szkennelés	7
3d scan programok összehasonlítása	10
Arc-, testmodellezés.....	10
LOOXIS FaceWorx	11
FaceGen Modeller.....	14
3DSOM.....	18
Összegzés.....	24
OBJ megjelenítő program.....	25
Az OBJ fájl formátum.....	25
A szoftver fejlesztési folyamata.....	25
Összefoglalás	28
Köszönetnyilvánítás	30
Irodalomjegyzék	31

Bevezetés

Manapság az ipar és a mindennapi élet legtöbb területének működése – például a biztonsági rendszerek, korszerű gyártó sorok, az igazságszolgáltatás – elképzelhetetlen lenne a digitális fotózás, alapvetően a digitális technika vívmányai nélkül.

A tudomány és technika egyre jobban fejlődő és tért nyerő területe a 3D szkennelés, a valós világ dolgainak megörökítése, modellezése, minél pontosabban és a lehető legegyszerűbb módon. Magának a szkennelésnek több módja van, ebből az egyik, amikor több fényképet készítünk a szkennelendő objektumról (ez lehet tárgy, ember, stb.), különböző megvilágítások, szemszögek alkalmazásával, majd ezeket dolgozzuk fel.

A dolgozatomban első részében a digitális fényképezés kialakulásáról, történetéről, illetve annak legfontosabb fogalmairól lesz szó. Tisztában kell lennünk az alapvető fogalmakkal, ugyanis amikor elkészítjük a fényképeket a szkennelendő objektumról, ezen beállítások helyes megadása nagyon fontos.

Ezután megismerkedünk a modellezéssel, pontosabban a 3D modellezés elméletével és technikáival, ezek közül a 3D szkenneléssel részletesebben.

Majd bemutatásra kerül három szoftver, melyek egy-egy modellezési módhoz kötődnek, ezek alapvető működése, tulajdonságai, összehasonlítása.

A dolgozat utolsó részében pedig egy 3D objektumról alkotott modell (OBJ) megjelenítésére alkalmas szoftver fejlesztéséről lesz szó.

Azért választottam ezt a szakdolgozati témát, mert úgy érzem, hogy maga a fényképezés és az informatika képfeldolgozás és grafikai része közel áll hozzám. Maga a téma is érdekel és kihívást jelent számomra, hogy szakmailag tárgyaljam és elemezzem azt. A szoftver jó lehetőség arra, hogy fejlesztésekor felhasználjam azokat a programozási és grafikai ismereteimet, melyeket az egyetememen eltöltött éveim alatt szereztem.

A digitális fényképezés és alkalmazásai

Történet

1884 óta több mint 100 évig a George Eastman által feltalált celluloid film monopólium helyzetű volt a fényképezésben [1]. Ebből is a 35mm-es film vált a legelterjedtebbé, olyannyira, hogy még a mai időkben is a legtöbb mozi filmet erre az alapanyagra rögzítik. Ebből adódóan a fotósok a 35mm-es negatívra vonatkoztatott értékeket használták, ezeket szokták meg. Így a digitális fényképezőgépeknél - habár nem vagy csak pár típusnál éri el vagy közelíti meg az érzékelő mérete a 35mm-es formátumot – erre a formátumra adjuk meg a jellemzőket (pontosabban a fókusztávolságot).

Itt kell megemlítenem, mielőtt a digitális fényképezőgép „megszületne”, Dulovits Jenő-t. Ő volt az aki az első magyar fényképezőgép szabadalmat jegyzi. Ez volt a Duflex, az első egylencsés, egy aknás, tükörreflexes, pentaprizmás, szemmagasságú keresős, beugró blendés, oldalhelyes és nem fejtetőn álló képet mutató kisfilmes fényképezőgép [2].

Egy digitális fényképezőgép alapvetően abban különbözik egy analóg fényképezőgéptől, hogy analóg esetben a látott kép mását rögzítjük filmen vagy egyéb fényérzékeny anyagon, ezzel szemben digitális esetben ez a kép bitkódokká alakul és az így kapott adathalmazt tároljuk valamilyen memóriamodulban, majd ezen adatokból valamilyen dekóder (szoftver) segítségével lesz újra látható kép.

A digitális technika előfutárai a videokamerák voltak, ahol a kamera digitális jelet tárolt a használt szalagon, melyből a videomagnó állított elő újra képet.

A digitális kamera, illetve a digitális fotózási technika létrejöttét tulajdonképpen a hidegháborúnak köszönhetjük, ugyanis az egyes hatalmak hírszerzései már műholdak segítségével szándékoztak kémkedni, azaz főleg képeket készíteni. Ám az űrben lehetetlen vagy költséges lett volna fotólaborokat létesíteni. Ezért volt szükség egy olyan technika kifejlesztésére, amely megoldja ezeket a hibákat.

Ezekből adódóan 1969 októberében a Bell cég laboratóriumában megszületett az első CCD érzékelő, amely a digitális képalkotás lelkét jelenti.

Az első kísérleti gépre – mely már a digitális képalkotási technikát alkalmazza – „csupán” 6 évet kellett várni, ugyanis ekkor alkották meg a Kodak mérnökei azt a gépet, amely 100×100 pixel felbontású digitális képet volt képes készíteni. Ám ez a gép a labor fogja maradt. Újabb 6 év múlva a Sony mérnökei megalkották a Mavica nevű, cserélhető objektíves fényképezőt, amely egy 1,44 inch-es hajlékonylemezre 50 db 0,72 MP (megapixel) felbontású képet volt képes rögzíteni. Ám ekkor még az elkészült képek megtekintéséhez külön olvasóegységet kellett alkalmazni.

Ezután már viszonylag rövid időn belül léptek színre az újabb kamerák és szabványok, melyek a digitális képalkotás fejlődését eredményezték. Ilyen volt az 1988-ban megjelent Fuji DS-1P mely már cserélhető memóriaegységre rögzített. Ekkor jelent meg a JPEG formátum, mely a képek tárolásának és kezelésének egyszerűsödését eredményezte. Ezután 1991-ben készült el az első professzionális felhasználásra megalkotott digitális gép, a Kodak DCS100, amely valójában egy átalakított analóg Nikon F3 volt.

A '90-es évek közepére a digitális gépek egyre elérhetőbb áron és jobb minőségben készültek. Ekkor alakultak ki a különböző gépkategóriák, és megszülettek az egyre nagyobb kapacitású és olcsóbb memóriakártyák. 2003-ban jelent meg az első elérhető áru digitális tükörreflexes fényképezőgép, a Canon 300D.

Ma már bárki számára elérhető minőségű és áru gépek léteznek, és már a legtöbb mobiltelefon is tartalmaz egy egyszerű (VGA) digitális kamerát. Ebből adódóan a digitális fényképezőgép a mindennapi életünk része lett.

A digitális fényképezés – mint technika, és mint lehetőség – ilyen nagy elérhetősége felveti azt a kérdést, hogy megmarad-e alkotási eszközként vagy egy fogyasztási cikké „silányul”.

Működési elv és alapfogalmak

A mai digitális fényképezőgépek működési elvét [3], [4] a következőképpen foglalhatnánk össze:

1. A rögzíteni kívánt tárgyról visszaverődött fény az objektíven keresztül, különböző optikai módosítások után az érzékelőre kerül.

2. Eközben a – legtöbbször az objektívben elhelyezett – rekesz (blende) által meghatározott nyílás befolyásolja, hogy mennyi fény jut a gép belsejébe.
3. Ugyancsak a bejövő fény mennyiségének változtatására szolgál (legfőbbképp) a zárszerkezet, mely csak egy bizonyos ideig van nyitva, azaz csak egy bizonyos ideig engedi át a fény az érzékelőre, majd ezután újra lezár.
4. Miután a kép elérkezett az érzékelőre, az elektronika leolvassa a jelet, ezzel kinyerve abból a látott képet, mint egy adathalmazt. Megjegyzendő, hogy az érzékelőről nem színinformációkat olvas le az elektronika, hanem az egyes pixelek fényerősség-értékeit, majd ebből valamilyen algoritmus állítja elő a színeket.
5. Ezután ezt az adathalmazt – géptől és beállításoktól függően – megjeleníti a gép LCD kijelzőjén, illetve tárolja azt a memória modulon, vagy nyers (RAW) vagy tömörített (JPEG, TIFF) formátumban.

Az egyes fázisoknak, illetve az egyes alkotórészeknek meg vannak a maguk tulajdonságai, beállításai, melyik nagymértékben befolyásolják az elkészült kép minőségét és milyenségét.

Ilyen a **záridő**, mely a zárszerkezet jellemzője. Azt adja meg, hogy mennyi ideig van nyitva a zárszerkezet. Ez az érték 1/8000 s-től akár 60 s-ig terjed – illetve egyes gépeknél az ún. B-záridő használatával, az exponáló gomb nyomva tartásáig, vagy T-záridő esetén az exponáló gomb két lenyomása közötti ideig. Rövid záridő használata esetén „megfagyaszthatjuk” a mozgást, akár egy kolibri szárnycsapásait is megfigyelhetjük. Hosszú záridő esetén az érzékelő észlelni fogja a mozgást is, tehát elmosódott kép készül, illetve helyes használat esetén szebb, nagyobb művészi értékű képet is készíthetünk.

Másik ilyen jellemző a **blende** érték. Ez a tulajdonság, egy viszonyszám, amely a használt objektív fókusz távolságához viszonyítva adja meg, hogy a rekesz lamelláinak „látható” része hányadát takarja el a nyílásnak. Ezzel tudjuk például befolyásolni a mélységélességet, azaz hogy a fókuszált tárgy (pont) előtt és után mennyi az a távolság, ahol a tárgyról (pontokról) még éles képet kapunk.

Az analóg fényképezőgépekben használt filmektől örökölt tulajdonság az **érzékenység**, vagy elterjedtebb nevén az ISO érték. Ez azt határozza meg, hogy mennyi fény szükséges egy normál expozíciós értékű kép előállításához. Minél nagyobb ez a szám (pl.: ISO 6400) annál

kevesebb fényre van szükségünk, annál nagyobb rekeszértéket vagy kisebb záridőt használhatunk, ugyanazon kép elkészítéséhez, illetve minél kisebb ez az érték (pl.: ISO 64) annál több fény szükséges, azaz annál hosszabb záridő illetve kisebb rekeszértéket szükséges használnunk.

Fontos tulajdonság még a fókusztartomány, **fókusztávolság** [URL1]. Ez a nagyítás mértékét adja meg, azt, hogy az objektív milyen mértékben nagyítja meg az általunk látott képet. A továbbiakban, ha fókusztávolságról, illetve annak értékéről van szó, akkor annak 35mm-es formátumra vonatkoztatott értéke fog szerepelni.

Legtöbbször ezek a tulajdonságok nagymértékben összefüggenek egymással. Például a bemozdulás elkerülés végett célszerű a használt fókusztávolság reciprokánál nem nagyobb záridőt megadni. Ez alapján, ha a fókusztávolság 100 mm, akkor 1/100 s záridővel még viszonylag biztonsággal fotózhatunk, nem lesz – érezhetően – bemozdult a kép, azonban már 1/80 s záridővel nagy ennek a valószínűsége.

Alkalmazás

Mint már említettem, a digitális fotótechnika megtalálható a mindennapi élet szinte bármely területén, legyen szó az orvostudományról, az űrkutatásról, a biztonságtechnikáról vagy a szórakoztatásról. Minden területen más és más szempontokat veszünk figyelembe.

Egyik ilyen alkalmazási terület az ipar. Itt is több célra használják magát a technikát. Például egyes gyártósorokon az automatizáláshoz elengedhetetlen, hogy figyeljék a futószalagon haladó termékeket. Mivel ezek legtöbbször nagyon nagy sebességgel haladnak, ún. nagy sebességű kamerát alkalmaznak, amelyek több száz vagy ezer fps (Frames Per Second – egy másodperc alatt rögzített képek száma [URL2]) sebességűek. Ma már létezik ezekből olyan is, amely több millió fps sebességű.

Fontos alkalmazási terület még a biztonságtechnika, ahol legtöbbször fekete-fehér kamerát alkalmaznak.

A modellezésről

A tágabb értelemben vett modellezéskor a valós világ valamely részének, objektumának a tulajdonságait, viselkedését próbáljuk meg rekonstruálni, megörökíteni vagy egyes esetekben egyszerűbb, érthetőbb formára hozni.

„A háromdimenziós számítógépes grafikában a **3D modellezés** az a folyamat, amely matematikailag ábrázol tetszőleges háromdimenziós objektumot.” [URL3]

3D modellezés esetén magát az objektumot, annak méreteit, kinézetét, esztétikai tulajdonságait próbáljuk megörökíteni, esetlegesen annak későbbi rekonstruálása céljából. A modellezési folyamat végterméke a **3D modell**.

Ha a 3D modellből újra 2D (síkbeli kép) modellt szeretnénk előállítani, akkor renderelést kell végeznünk.

Magának a modellnek az elkészítése – a modellezés – alapvetően háromféleképpen történhet: manuálisan, valamely algoritmus által vagy szkenneléssel. Mivel egy ilyen modell legkisebb elemei a pontok (térbeli pontok), illetve azok koordinátái, manuális előállítás alatt a pontok kézzel történő megadását értjük. A későbbiekben a modell előállításának algoritmus, illetve szkennelés általi előállítását tárgyaljuk részletesebben.

A modell algoritmus általi előállításakor – legtöbbször – egy vagy több digitális képből állítjuk elő a pontokat. Ennek a módszernek is több megközelítése van. Legelemibb eljárás – legalább kettő kép esetén – amikor ismerjük a kamerák, a nézőpontok térbeli helyzetét, így van egy vonatkoztatási rendszerünk. Ezután meg kell keresnünk a modell valamelyik pontját mind a két képen, majd az adott képeken való elhelyezkedésükből, és a kamerák térbeli elhelyezkedéséből kiszámolható az adott pont térbeli helye. Újra kereshetünk 1-1 pontot, ugyanakkor célszerűbb, ha a már megtalált ponttól indulunk ki, hisz ekkor nagyobb a valószínűsége, hogy egyező pontot találunk. Az utóbbi esetben – amikor nem „vaktában keresgélünk” – már az algoritmus futása közben elkészíthetjük a modell testhálóját. Ha azonban nem ezt a módszert alkalmazzunk, akkor az algoritmus végére csak egy pontot kapunk, amelyre például neurális hálózatok alkalmazásával feszítünk ki testhálót.

A Szkenneléses eljárásról a következő alfejezetben ejtek szót.

Magát a modellt többféle módon ábrázolhatjuk. A legalapvetőbb (legtöbbször ezt használják a fent említett technikák) a poligonmodellezés, amikor pontjaink vannak a 3D-s térben, összekötve szakaszokkal, és ezek együtt egy poligonhálót (testhálót) alkotnak. Nagy hátránya ennek az ábrázolási módnak, hogy például nem síkfelületet csak közelítően tud ábrázolni. Ha pedig minél nagyobb pontossággal szeretnénk közelíteni egy görbült felületet, akkor az adathalmazunk elérheti azt a szintet, amikor már kezelhetetlen, túl nagyméretű. Ennek az ábrázolási módnak egy továbbfejlesztett változata a textúrázott-poligonmodellezés, amikor a textúrát is tároljuk, ezzel gyorsabb a renderelés.

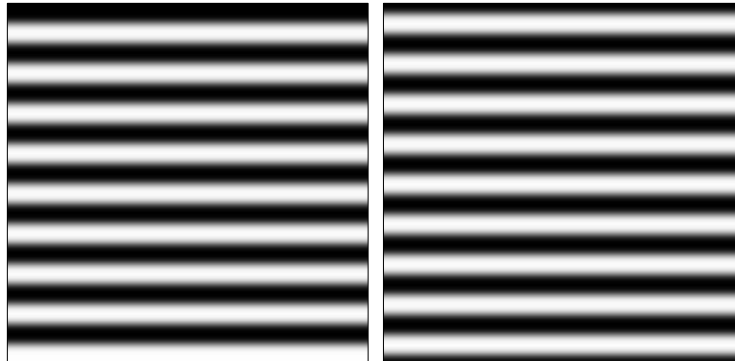
Léteznek egyéb ábrázolási technikák is, például: NURBS modellezés, Splines & Patches modellezés és Primitív-modellezés.

Szkennelés

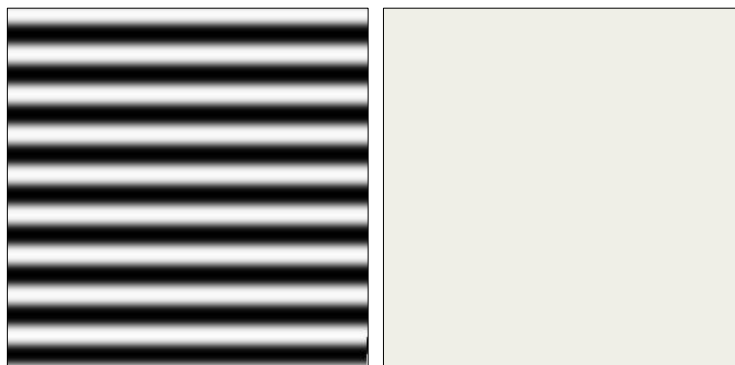
Szkenneléses eljárás esetében két fő technikát különböztetünk meg. Ez első, amikor fényképeket készítünk az objektumról. Erről a technikáról már volt szó az algoritmusos eljárás kapcsán, azonban a kép elkészítésének technikája miatt ide is sorolhatjuk. Ennek az eljárásnak is több féle megvalósítása van. Ezek közül kettő:

- Több képet készítünk különböző nézőpontokból. Ekkor legtöbbször valamilyen fix kalibráló ábrát is alkalmazunk. Ennek az ábrának a segítségével tudjuk meghatározni a kamera helyzetét, illetve, hogy épp milyen szemszögből készítettük a képet. A későbbiekben tárgyalt 3DSOM Pro nevű szoftver is ezt az eljárást támogatja. Ennél az eljárásnál legtöbbször a háttér kimaszkolása után egy kontraszt vagy körvonalelemzést hajtunk végre.
- Több képet készítünk, azonban ekkor nem feltétlenül több nézőpontból. Az eljárás lényege, hogy a témát különböző képekkel világítjuk meg (legegyszerűbben egy projektor használatával). Egy ilyen képsorozat lehet, amikor több (minimum 3) olyan képet használunk, ahol fehér és fekete csíkok váltják egymást, egész pontosan olyan képeket, ahol a fény erőssége függőleges irányba szinuszosan írható le. Egy ilyen

sorozatot mutat az 1. ábra (1.-2. fázis) és a 2. ábra (3.-4. fázis). Ezen kívül szokás még plusz egy teljesen világos képpel is levilágítani a tárgyat, a textúra miatt.



1. ábra, levilágítás 1.-2. fázis, saját készítés



2. ábra, levilágítás 3.-4. fázis, saját készítés

Ezen technika alkalmazásakor legtöbbször nyugalomban levő tárgyakat digitalizálunk. Ekkor a fényképező beállításakor célszerű – a minél nagyobb mélységélesség eléréséhez – nagy blende értékkel (pl.: F16.0) dolgozni, illetve a kép zaj elkerülése érdekében kis érzékenységgel (pl.: ISO 100). Ekkor expozíciótól függően a gép automatikája akár több másodperces záridőt is használhat, tehát állványt kell használnunk, a bemozdulás elkerülése végett.

Másik fő technika a **lézeres szkennelés**. Ekkor egy lézerforrásból több trillió fotont küldünk a szkennelendő objektumra. Ezeknek csak kis része verődik vissza, melyeket egy optikán keresztül érzékelünk és dolgozunk fel. Mivel a lézer pozicionálása nagyon gyors, ezért hamar teljes modellt kaphatunk az objektumról. Sokféle lézer-szkennert létezik. Ezek közül a két fő típus, illetve gyártó a DAVID 3D Scanner [URL4] és a Handyscan 3D Scanner [URL5].

A két típus között az alapvető különbség, hogy a David 3D Scanner (3. ábra) külön tartalmaz 1 db lézert, amely 1 sávban világítja meg az objektumot, és erről egy külön kamera egység készíti a képet. Itt nekünk kell a lézert végighúzni az objektumon. Illetve ebben az esetben szükségünk van egy speciális hátfalra, amely a pozícionálást segíti.



3. ábra DAVID 3D Scanner, forrás: http://www.david-laserscanner.com/gfx/start/david_3d_scanner_4s.gif

Handyscan 3D Scanner (4. ábra) esetében pedig egyetlen kézi készülékről van szó. Ekkor az egész folyamat azzal kezdődik, hogy az objektumra jelzőpontokat helyezünk, majd a kézi szkennelvel „végig pásztázzuk” az objektumot. A kézi eszközbe 2 db lézer van beépítve, illetve egy optika (kamera), ami érzékeli a lézerek által letapogatott képrészletet.



4. ábra, Handyscan UNISCAN 3D Scanner, forrás: <http://www.creaform3d.com/images/uniscan.jpg>

Ha az egész objektumot teljesen letapogattuk az eszközzel elkészül az objektum 3D modellje, illetve, számítógéphez csatlakoztatva pedig a modell elkészítésének a folyamatát is nyomon követhetjük a képernyőn. Ehhez hasonló lézerszkennel még a Z Corporation ZScanner™ [URL6] sorozat is.

Mindkét fent említett gyártó forgalmaz szoftvereket a saját eszközeihez.

3d scan programok összehasonlítása

A legtöbb 3d szkennelési rendszer gyártó szolgáltat programokat a termékeik használatához. A következőkben különböző, független gyártók által készített szoftverek próbaverzióját mutatom be és elemzem. A bemutatott három szoftver mindegyike azt a technikát, eljárást használja, amikor fényképek segítségével állítja elő a modellt.

Az első két szoftvert csak arcmodellezésre használjuk. Ezen szoftverek vizsgálatához egyik jó barátomról készített egy előnézeti és két profil képet fogok használni (5-6-7. ábra).



5. ábra, akos_front.jpg,
forrás: saját készítés



6. ábra, akos_left.jpg,
forrás: saját készítés



7. ábra, akos_right.jpg,
forrás: saját készítés

Arc-, testmodellezés

Modellezéskor lehetnek olyan „egyszerűbb” esetek, amikor tudjuk, hogy mi a modellezett tárgy, mik a jellemző tulajdonságai. Ekkor ezeket a jellemzőket határozzuk meg, amelyekből már egy algoritmus által generálhatjuk a modellezendő objektumot legjobban közelítő modellt.

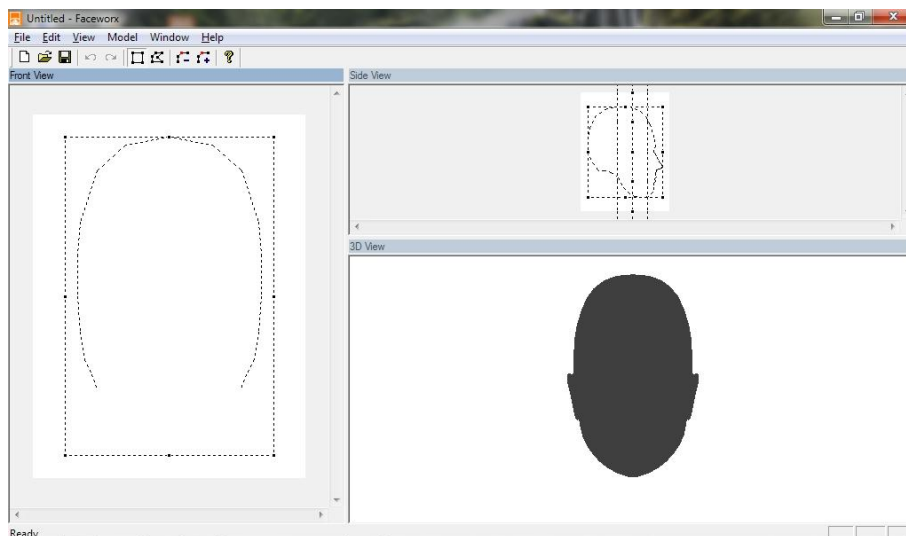
Ilyen például az arcmodellezés, ugyanis az arcnak vannak meghatározó pontjai, vonalai, melyek helyzete és aránya – többek között – a nemtől, kortól és a rassztól is függ. Ezen jellemzők alapján pedig legtöbbször sikeresen előállítható a modell.

Másik ilyen modellezési eset lehet, amikor valamilyen „szabályos” geometriai alakzatot, például egy gömböt szeretnénk modellezni. Feltételezzük a nézőpont (kamera) koordinátáinak ismeretét. Alkalmazzuk a kör középpontjának meghatározásánál használt módszert kiterjesztve a 3 dimenziós térre. Ekkor a következő (saját) eljárással meghatározhatjuk a gömb középpontját és a sugarát, amelyekből már generálható a modell.

1. Határozzunk meg a gömb felületén négy különböző pontot – legyenek sorra A, B, C és D – melyek nincsenek egy síkban.
2. Vegyük az A, B és C pontokat. Határozzunk meg például az AB és a BC szakaszok szakaszfelező merőleges síkját – legyenek sorra S_{AB} és S_{BC} . Tudjuk, hogy a gömb középpontja egyenlő távolságra van a gömb felszín pontjaitól, és hogy a szakaszfelező sík minden pontja egyenlő távolságra van a szakasz végpontjaitól. Ezekből kifolyólag a gömb középpontja illeszkedik a síkra. (Ebből következik, hogy az S_{AB} és az S_{BC} felezik a gömböt.)
3. Mivel az A és a B pontok különbözőek, ezért az S_{AB} és az S_{BC} metsző síkok lesznek, és metszetük egy olyan egyenes lesz, amely tartalmazza a gömb középpontját – nevezzük az egyenest e_{ABC} -nek.
4. Ismételjük meg a 2. és 3. lépést a B, C és D pontokkal. Ennek eredménye az e_{BCD} egyenes.
5. Mivel mind az e_{ABC} és az e_{BCD} is tartalmazza a gömb középpontját, és mivel az ABC és a BCD háromszög nem illeszkedik ugyanarra a síkra, a két egyenes egy pontban, a gömb középpontjában – legyen O – metszi egymást.
6. A középpont ismeretében, pedig ha veszünk például az A pontot, akkor e két pont távolsága pedig a gömb sugara. Így a gömb már paraméteres alakban felírható.

LOOXIS FaceWorx

Maga a program a LOOXIS cég [URL7] fejlesztése. Fő funkciója, hogy egy front (előnézeti) és egy side (oldalnézeti, profil) képből előállítja az adott fej, arc modelljét. A programban manuálisan kell meghatároznunk – az adott képek segítségével – az arc fontosabb, meghatározó vonalait, pontjait. A program, ezen információk felhasználásával, valós időben hozza létre a 3 dimenziós modellt, amelyet meg is tekinthetünk.



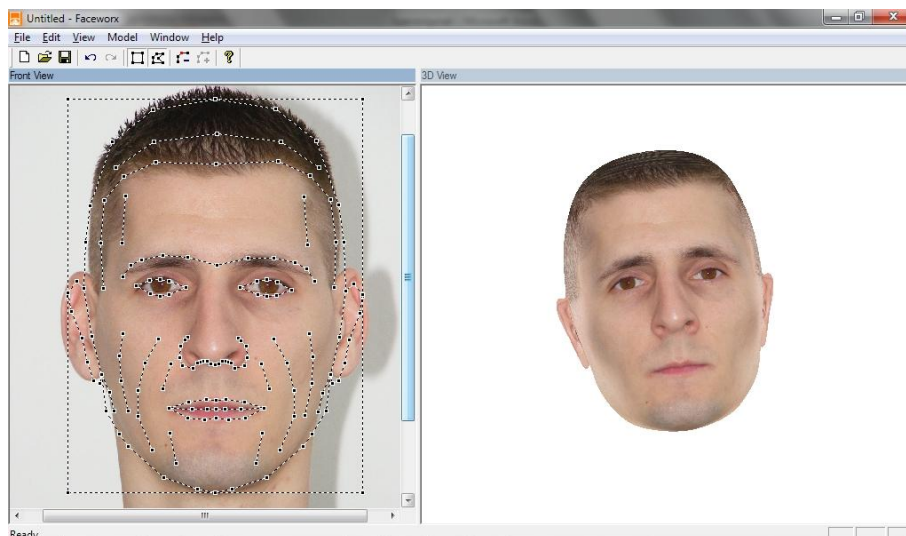
8. ábra, LOOXIS Faceworx, kezdőképernyő, forrás: képernyőkép (saját)

A program indítása után a 8. ábrán látható kép fogad minket, ahol látható a program 3 fő nézete: a **Front View** (előlnézet), a **Side View** (oldalnézet, profil) és a **3D View** (3d modell nézet). A nézetek megjelenítését és elrendezését a **Window** menüpont segítségével tudjuk befolyásolni. Az egyes nézetekben a navigálás – elsődlegesen – az egér segítségével lehetséges. Nagyítás mind a három nézetben a görgő segítségével lehetséges. A 3D nézetben a bal egérgombbal a modellt forgathatjuk, a jobb egérgombbal pedig a modell pozicionálása végezhető. A másik két nézetben a bal egérgombbal a jellemzők pozicionálhatjuk.

Észrevehetjük, hogy a Front és a Side nézetben szaggatott vonalak, illetve négyzetek jelzik a főbb vonalakat, pontokat. Alap esetben csak a fej körvonalai láthatóak. Szerencsés, ha a **View / Points / All** menüpont segítségével bekapcsoljuk az összes jellemző megjelenítését.

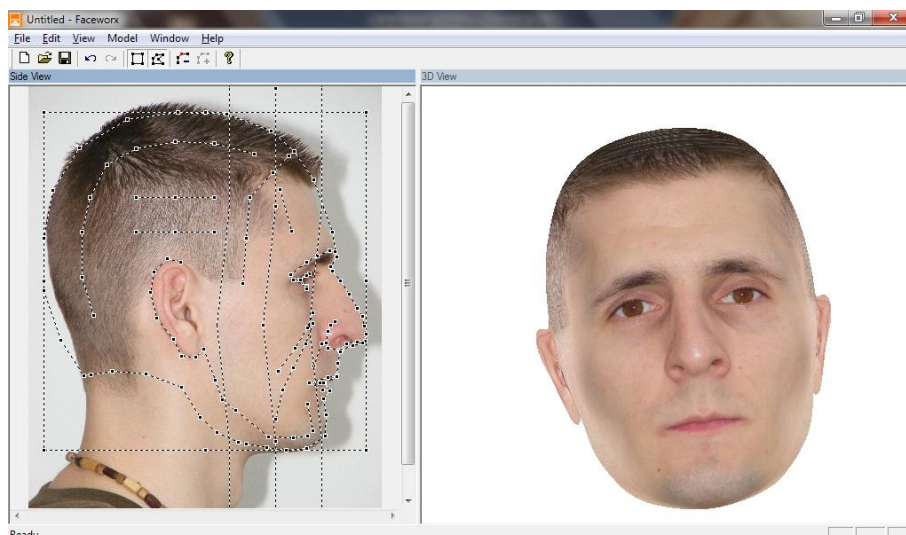
A következőkben a program működésének bemutatására a fentebb említett példaképek felhasználásával létrehozunk egy 3D modellt.

1. Váltunk a **Window / Front and 3D** menüpont segítségével nézetet, majd a **Modell / Load Front** menüpont segítségével töltjük be az *akos_front.jpg* (5. ábra) képet. Miután megjelent a kép, pozicionáljuk a főbb pontokat (orr, szemöldök, száj és körvonalak). Ha az előzőeket sikeresen végrehajtottuk, akkor a 9. ábrához hasonló képet kapjuk.



9. ábra, LOOXIS Faceworx, Front és 3D nézet, forrás: képernyőkép (saját)

2. Váltunk a **Window / Side and 3D** menüpont segítségével nézetet. Ekkor a Side és a 3D nézet lesz látható. A **Modell / Load Side** menüpont segítségével töltjük be az *akos_right.jpg* képet (7. ábra). A kép megjelenése után, ezen is végezzük el a pozicionáló lépéseket. Végeredményként a 10. ábrán levő képet kell kapnunk.



10. ábra, LOOXIS Faceworx, Side és 3D nézet, forrás: képernyőkép (saját)

Miután elkészültünk a jellemzők manuális meghatározásával, lehetőségünk van a kész modellt elmenteni. Amennyiben még tökéletesíteni szeretnénk modellünket a programmal, akkor célszerű a **File / Save** menüpont segítségével a Faceworx saját (FWX) fájlformátumába

menteni. Lehetőségünk van még a **File / Export** menüpont segítségével OBJ [URL8] fájlformátumba elmenteni.

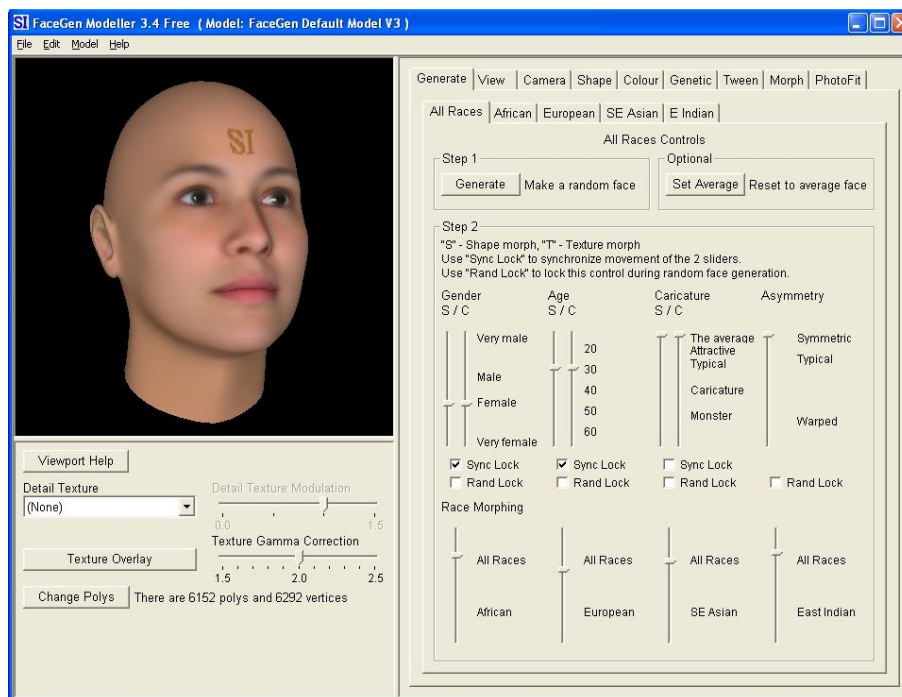
A program előnye, hogy viszonylag kicsi és gyors. Sok jellemzőt lehet beállítani, a belátásunk szerint, így maga a modell is bármikor „szerkeszthető”. Hátránya épp az egyik előnyében rejlik, ugyanis a sok jellemzőt néha nehézkes átlátni, és a szabad pozicionálás sem mindig megoldható.

FaceGen Modeller

A programot a Singular Inversions [URL9] cég fejlesztette, illetve fejleszti, mely már 1998 óta foglalkozik az emberi arc modellezésével, annak elméleti és matematikai hátterével. Közben magát a programot az EA (Electronic Arts) játékfejlesztő cég többször használta a játékaiban található arcok megformálására.

Ez a program a Faceworx-el ellentétben a modell képekből történő rekonstruálásán kívül más lehetőséget is biztosít a felhasználónak. Az emberi arc több éves megfigyeléséből és elemzésből, és egyéb erről a területről származó tanulmányok segítségével meg tudtak konstruálni egy olyan felületet, ahol olyan környezeti és egyéb jellemzőket is be tudunk állítani a modellel kapcsolatban, mint a kor, a rassz, az arc szimmetriája, a nemi jellegek, illetve a valóság-hűség (mennyire legyen karikatúraszerű), melyek nagyban meghatározzák az arc bizonyos tulajdonságait.

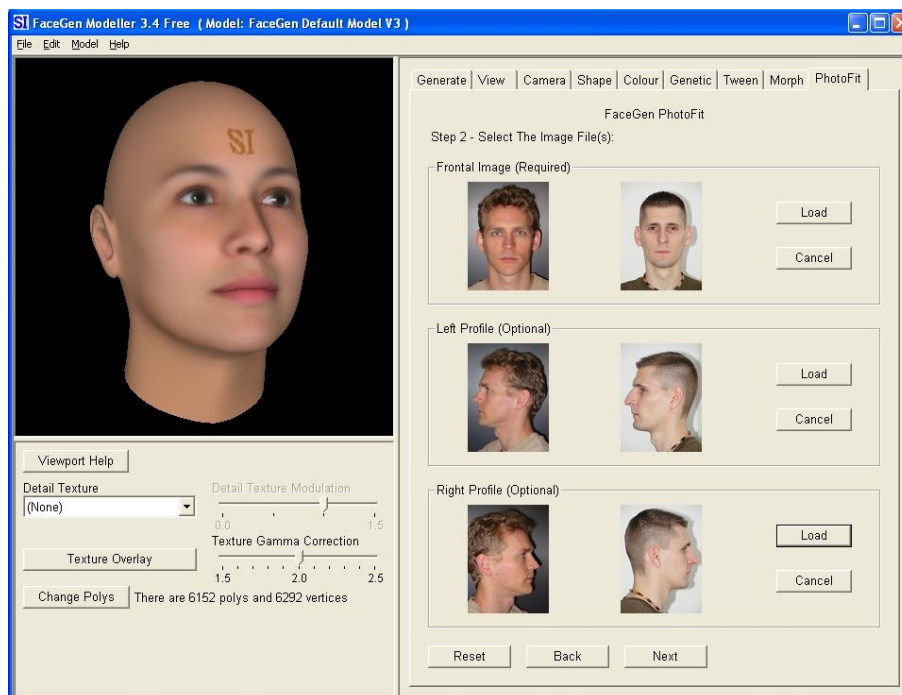
A program kezdőképernyőjét a 11. ábrán láthatjuk. (A modellen látható „SI” felirat a próbaverzió korlátozásai közé tartozik.)



11. ábra, FaceGen Modeller, kezdő ablak, forrás: képernyőkép (saját)

A program alapértelmezetten, az elindulásakor generál számunkra egy standard arcot. Ez a jobb oldalt, számos fülön levő elemekkel konfigurálhatjuk. A következőben nem térek ki részletesen a konfigurálási lehetőségekre, ellenben a Faceworx-höz hasonlóan állítsuk elő a példaképekből a modellt. Ezt a következőképp tehetjük meg:

- I. Kattintsunk a legfelső fülek közül a **PhotoFit** fülre. Ezután láthatjuk, hogy milyen képekre lesz szükségünk, illetve több információt is megtalálhatunk itt a folyamatról. Fontos megjegyezni, hogy konfigurációtól függően 2-5 perc (gyengébb konfiguráción akár 15 perc) is lehet egy kép feldolgozásának az időtartama.
- II. A **Next** gombra kattintva ahhoz az ablakhoz jutunk, ahol megadhatjuk a felhasználni kívánt képet. Itt a **Load** gombok segítségével töltsük be a megfelelő – az 5-6-7. ábrán szereplő – példaképet (12. ábra).



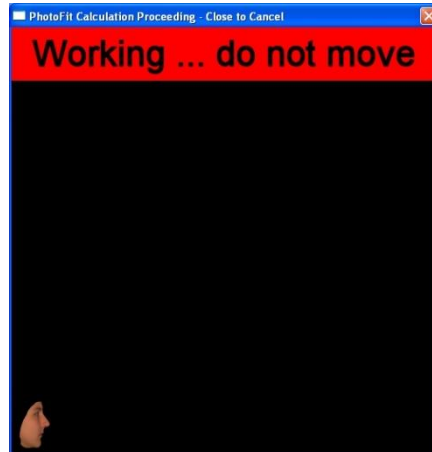
12. ábra, FaceGen Modeller, képek megadása, forrás: képernyőkép (saját)

- III. A **Next** gombra kattintva meghatározhatjuk a különböző nézetekhez tartozó, jellemző pontokat, amihez a program egy-egy mintaképet is mutat a felhasználó számára. Ezeket a pontokat a minták alapján helyezük a megfelelő helyekre. A 13. ábrán a pontok lehetséges elrendezését látjuk a különböző nézeteken.



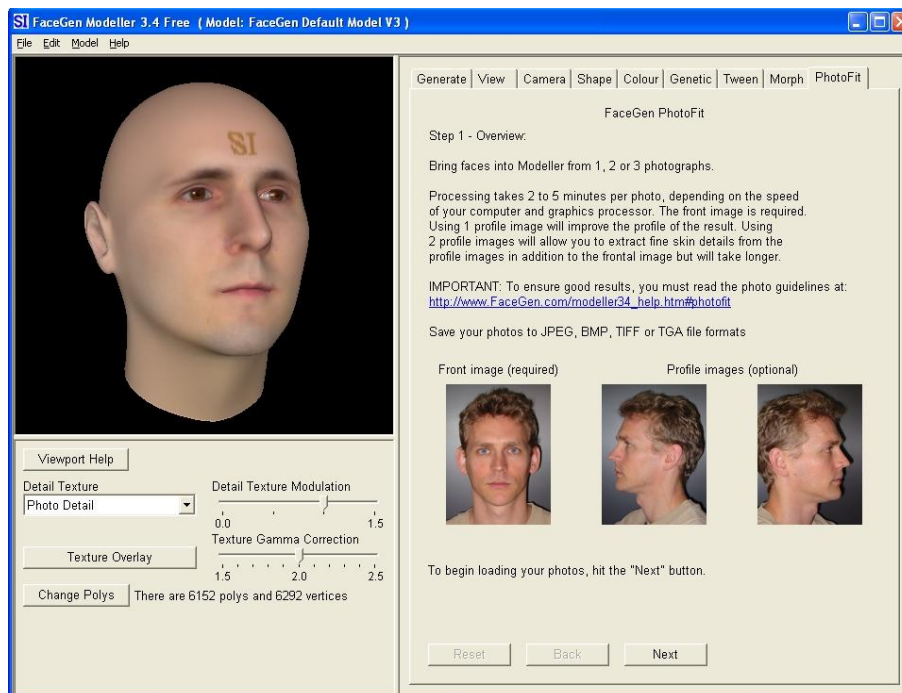
13. ábra, FaceGen Modeller, Jellemző pontok helyzete, forrás: képernyőkép részletek (saját)

- IV. Miután meghatároztuk az összes pontot eljutunk a folyamat 4. lépéshez, ahol ténylegesen elindíthatjuk a folyamatot. Ha rá kattintunk a **Start Now** feliratú gombra, a program elkezd a folyamatot. Ekkor megjelenik a 14. ábrán látható ablak, ahol a folyamat állapotát a bal alsó sarokban levő animáció mérete jelzi.



14. ábra, FaceGen Modeller, folyamatjelző ablak, forrás: képernyőkép (saját)

- V. Miután a folyamat elkészült visszakapjuk a kezdőképernyőt, azzal a különbséggel, hogy az általános modell helyett az újonnan előállított modell lesz látható (15. ábra).



15. ábra, FaceGen Modeller, a folyamat eredménye, forrás: képernyőkép (saját)

A modell nézeten a navigálás az egér segítségével lehetséges. Az egér bal gombjával a modellt forgathatjuk, a jobb gombbal pedig közelíthetjük, illetve távolíthatjuk a modellt.

Az elkészült modellt lehetőségünk van a program saját (FG) fájlformátumába menteni (**File / Save**), ami akkor célszerű, ha ezután még szerkeszteni szeretnénk a modellt a programmal. Ezen kívül a program teljes verziójában lehetőségünk van a **File / Export** menüpont segítségével a modellt az elterjedtebb OBJ fájlformátumba menteni.

Nagy előnye a programnak, hogy a modellt – viszonylag – automatikusan alkotja meg. Másik előnye, hogy a modellről a **File / Save Image** menüpont segítségével képet is készíthetünk.

Hátránya, hogy ha a modell nem hasonlít eléggé, kevés lehetőségünk van direkt módon finomítani rajta. Bár a **Shape** fül alatt lehetőségünk van az arc legtöbb részletének méretét és formáját állítani, ám ezeket pontosan, illetve önállóan nehéz megváltoztatni.

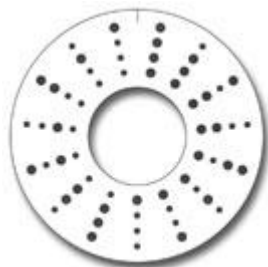
Összességében a program a rengeteg lehetősége miatt jó választásnak mondható.

3DSOM

A 3D Software Object Modeller (3D SOM [URL10]) a Creative Dimension Software Ltd. terméke, mellyel kisebb méretű tárgyakat tudunk modellezni.

Az alap elképzelés az, hogy vegyünk egy kamerát, amelynek tudjuk a helyét, és fényképezzük le több szögből a tárgyat, majd ebből a program előállítja a modellt.

A képek elkészítésénél egy kalibrációs ábrát (16. ábra) és homogén hátteret kell használnunk.



16. ábra, 3D SOM Pro, kalibrációs ábra, forrás: <http://www.3dsom.com/download/img/mat.jpg>

Az ábra közepére, és lehetőség szerint valamilyen pódiumra kell raknunk a modellezendő tárgyat. Ezután a kalibrációs ábrán látható 15 szögből (a 15 db 4-es pont csoport) kell lefényképeznünk. Célszerű a fényképeket állványról, önkioldóval készíteni. Ezen kívül használjunk minél nagyobb rekeszértéket, illetve megfelelően beállított vakut (lehetőségeink szerint). A lényeg, hogy minél homogénebb legyen a tárgy környezete. Ez azért fontos, mert a program a tárgy kontúrja alapján dolgozik, a kontúrponatok térbeli helyzetéből konstruálja meg a modellt.

Szükség van ezen kívül még 2 magasabbról készített képre, melyek előlről és hátulról mutatják a tárgyat. Amennyiben a tárgy alja, annak textúrája is fontos a modellezésnél, arról is készítenünk kell 5 képet, pontosabban 1-et, amelyen csak az alja látható, és 4-et magasabb szögből (hogy látható legyen a tárgy oldala is) és mindet más irányból.

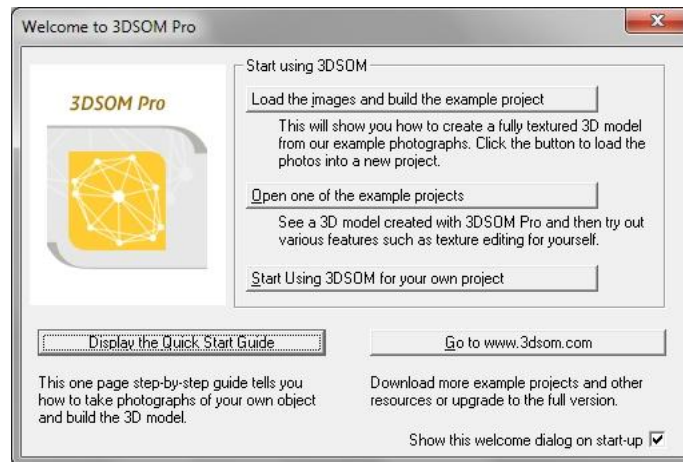
A modellezéshez én egy díszfigurát választottam. Az elkészült képekről készült montázs a 17. ábrán látható.



17. ábra, 3D SOM projekthez készült képek, forrás: saját fotó

A program elindításakor alapértelmezésként egy üdvözlőképernyő (18. ábra) jelenik meg, ahol lehetőségünk van saját projektet készíteni (*Start Using 3DSOM for your own project*), egy már elkészített példaprojektet szerkeszteni (*Open one of the example projects*), illetve egy

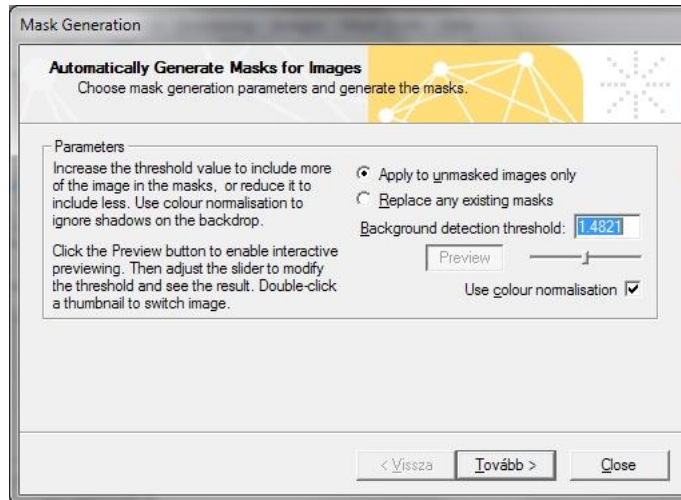
példa projekt képeit felhasználva végigkísérni egy projekt készítésének menetét (*Load the images and build the example project*).



18. ábra, 3D SOM, üdvözlőképernyő, forrás: saját képernyőkép

Válasszuk a harmadik opciót. Ezután egy üres projekt fogad bennünket. Töltsük be a képeinket az **Images / Add Images...** menüpont segítségével. Ezután mivel még nem kalibráltuk a kameránkat, a program rákérdez, hogy elindítsa-e a kamera kalibrációját végző funkciót. Válasszuk a Nem-et. Ez után a program felkínálja, hogy a gyorsabb működés végett miniatűröket készít a képeinkből, illetve, hogy beállítja azok pozícióját (álló-fekvő kép-e az adott kép, illetve melyik szögből fotóztuk). Miután ezt elvégezte a felső sorban megjelennek a képeink miniatűrjei.

Ezután a **Processing / Mask All Images** menüpont segítségével léphetünk tovább a projekt elkészítésében. Ekkor a felugró ablakban (19. ábra) állíthatjuk be azt a küszöbértéket, amely segítségével a kimaszkolandó területet meghatározza a program. A beállított értéket a **Preview** gomb segítségével tesztelhetjük. Amennyiben helyes az érték és csak a modellezendő tárgy nincs kimaszkolva, akkor a **Tovább** gombra kattintva a program elvégzi a maszkolást az összes betöltött képre.



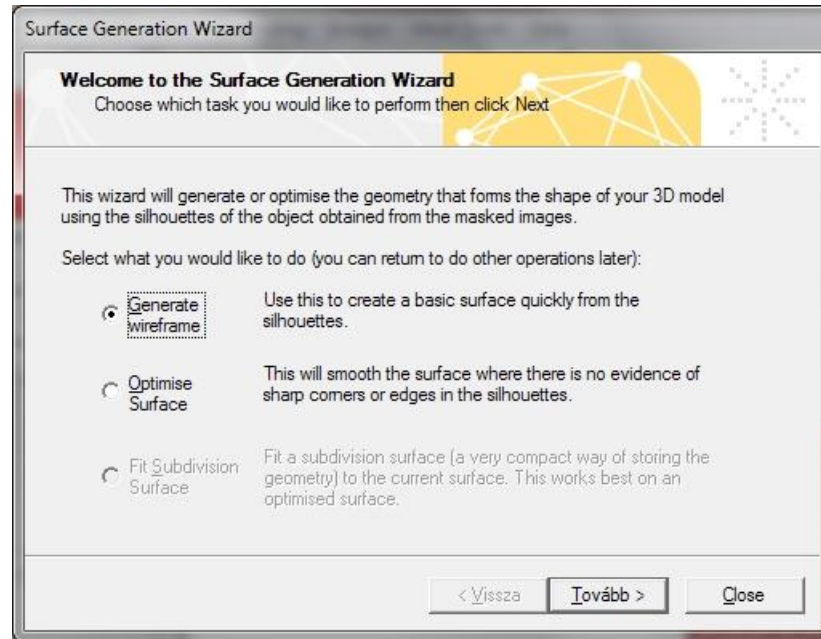
19. ábra, 3DSOM, maszkolási ablak, forrás: saját képernyőkép

Amennyiben valamelyik maszk nem tökéletes, lehetőségünk van alakítani rajta, mégpedig a következőképp:

1. A miniatűrök között keressük meg a képet és az egér jobb gombjával kattintsunk rá. Ekkor a helyi menüből válasszuk ki az **Edít mask** menüpontot.
2. A bal oldalt levő sávból a **Paint mask** szekció alatt válasszunk az alsó négy féle eset közül. Célszerű az első, körecsetet választani. Majd attól függően, hogy hozzá akarunk-e adni területet a maszkhoz, vagy el akarunk venni abból, a közvetlenül a csúszka fölötti lehetőségek közül kell az elsőt (hozzáadás), vagy a másodikat (elvétel) választani.
3. Ez után már csak a kezűgyességünkön múlik, hogy milyen lesz a maszk. Mivel nem lehet teljesen pontos küszöbértéket megadni, ezért előfordulhat, hogy olyan részek is kimaszkolódnak, amelyek a tárgy részei, illetve az árnyékok miatt sok rész nem maszkolódik ki, ezért a maszk szerkesztése is fontos lépés a minél élethűbb modell elkészítéséhez.

A projekt következő lépéseként a **Processing / Generate surface** menüponttal utasíthatjuk a programot, hogy a megadott képeinkből állítsa elő a modellt.

A folyamat egy ablakkal (20. ábra) indul, ahol három lehetőségünk van, melyek egyben magának a modell elkészítési folyamatának három állomása.

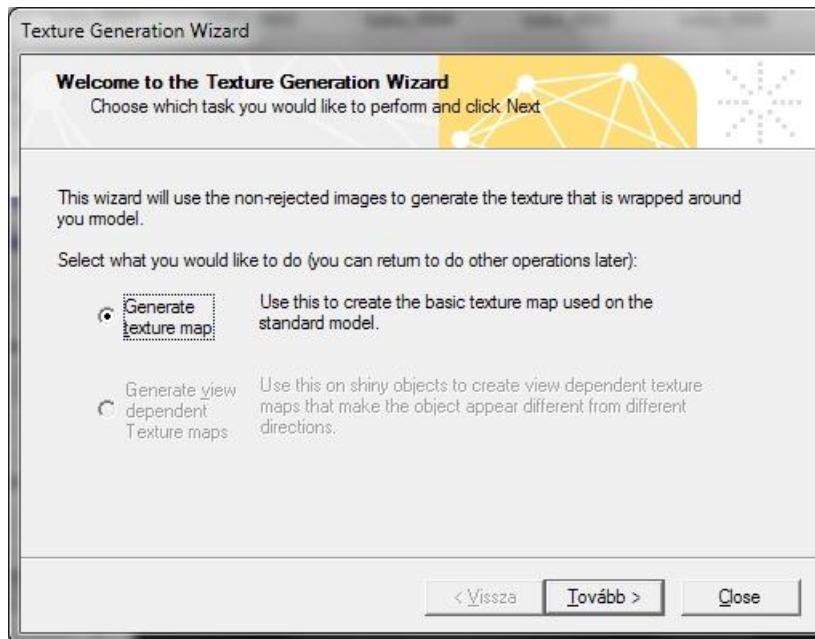


20. ábra, 3DSOM, felületgenerálás, forrás: saját képernyőkép

A folyamat során elkészítünk egy durvább modellt, majd ez egyre finomítjuk, végül pedig a legapróbb részletekig tökéletesítjük (lehetőségeinkhez mérten).

Először az első opciót kell kiválasztani, majd a **Tovább** gombra kattintani. Ekkor 1-2 beállítást még kér tőlünk a program, de ezeket hagyhatjuk az alapértelmezett értéken. A lépés befejezése után a 20. ábra ablakához jutunk vissza, ahol már a második opciót kell választanunk. Ekkor beállíthatjuk, hogy a modell mennyire illeszkedjen a tárgyhoz, mennyire legyen ahhoz hasonló. Itt nyugodtan választhatjuk az **Exact fit** beállítást. Mivel ilyenkor több számítást végez a program, ez a része a folyamatnak eltarthat több percig is.

Miután ezzel készek vagyunk, a folyamat befejező fázisa következik, amikor a képek alapján egy textúrát készít a modellnek a program. Ezt a folyamatot a **Processing / Generate Texture Maps...** menüpont segítségével indíthatjuk el. Ekkor a 21. ábrán látható ablak nyílik meg, ahol két lehetőségünk van.



21. ábra, 3D SOM, textúra ablak, forrás: saját képernyőkép

Először csak az első opció elérhető, amelyben elkészít a program egy alap textúrát a modellhez. A **Tovább** gombra történő kattintás után – mint a folyamat eddigi fázisaiban – beállíthatjuk a textúrázás paramétereit. Az egyik beállítás a textúra részletessége. Itt minél nagyobb számot adunk meg, annál részletesebb, és értelemszerűen annál nagyobb méretű lesz a textúra (memória, fizikai szint). A **Tovább** gomb megnyomásával elkezdődik a textúrázás, melynek eredményét a modellen rögtön láthatjuk.

A textúrázási folyamat következő lépése, amikor több textúrát generál a program a modellhez, méghozzá nézetenként. Tehát ez nem jelenti, hogy a modell textúrája részletesebb lesz, vagy pontosabb, pusztán azt érjük el vele, hogy a programban történő megjelenítéskor élethűbb képet kapunk a modelltől. Ezen részfolyamatnak egyetlen paramétere van, méghozzá, hogy hány plusz textúrát készítsen. Minél nagyobb számot adunk meg paraméternek, annál több szemszögből készít különálló textúrákat a program. Ebből kifolyólag a modell programbeli megjelenítése lassabb lesz, mivel már kisebb mozdításokkor is új textúrát kell betöltenie a programnak a modellre.

Az elkészült projektünket a **File / Save** illetve **Save as** menüpontok segítségével elmenthetjük a program saját SOM formátumába. Lehetőségünk van továbbá a **File / Export** menüpont segítségével több egyéb formátumba is menteni.

Fontos előnye a programnak, hogy lehetőségünk van Java, illetve Flash alapon is menteni a projektet (a modellt). Ez azt jelenti, hogy a program egy már előre elkészített megjelenítő alkalmazásba ágyazza bele a modellünket, így azt egyéb megjelenítő program nélkül is meg tudjuk nézni.

A program előnye, hogy viszonylag automatikusan dolgozik, viszont ez hátránynak is tekinthető, ugyanis amennyiben a program a folyamat során hibát vét (rosszul határozza meg a modell pontjait), nincs lehetőségünk beavatkozni, esetlegesen manuálisan meghatározni a pontokat, csak annyit tehetünk, hogy az aktuális részfolyamatot megismételjük más paraméterekkel.

Hátránya még, hogy sok előkészítő munkát igényel a program használata. Például számomra, csak a harmadik – általam elkészített – képsorozat volt olyan minőségű, hogy a programban azt fel tudjam használni. Ezen kívül a program eljárásának kontrasztelemező része nem tökéletes, ugyanis konkrét testek esetében a test „lyukait” nem ismeri fel, nem tudja kezelni.

Összegzés

Manapság temérdek mennyiségű 3D-s modellező, illetve megjelenítő szoftver létezik. Ezek közül a fentebb tárgyalt három szoftver olyan, melynek már a próbaverzióját is majdnem teljes funkcionalitással lehet használni, illetve viszonylag könnyen és jó eredményt tudunk elérni velük.

Ha most azt a feladatot kapnám, hogy egy ember arcát kellene 3D-s formában digitalizálni, én egyértelműen a FaceGen Modeller-t használnám. A Faceworx használata sok munkát igényel és nem mindig pontos, helyes modellt készít. A FaceGen Modeller használata pedig gyors, egyszerű, és a szükséges 3 db kép pedig egy perc alatt elkészíthető.

Amennyiben pedig egy bármilyen tárgy modellezése lenne a feladat, ha nem szeretnénk sok pénz áldozni a kiegészítő felszerelésre (David Laser Scanner), akkor a bemutatott 3D SOM tökéletes választás lenne.

OBJ megjelenítő program

A következőkben egy olyan egyszerű szoftvernek a fejlesztését írom le, mely a már ezelőtt is említett OBJ kiterjesztésű fájlokat képes megjeleníteni.

Az OBJ fájl formátum

Az OBJ [URL8] egy olyan karakteres fájlformátum, mely lehetővé teszi geometriai objektumok leírását – viszonylag – egyszerű és egységes formában. Ez a formátumot a Wavefront Technologies [URL12] kezdte elsőnek kifejleszteni a saját Advanced Visualizer animációs csomagjához, még az 1980-as években.

Egy ilyen OBJ fájlban minden egyes sor egy külön objektum, amely az azt felépítő csúcspont (vertex), kontrollpont (point), felület (surface) vagy egyéb elemnek feleltethető meg [URL13]. A teljes formátum nagyon komplex, sok olyan eszköz van benne, ami egy „egyszerű” objektum modellezéséhez feleslegesnek mondható. Ezért a kifejlesztendő szoftver (továbbiakban MyOBJViewer) csak egy olyan minimális, OBJ fájlokkal szembeni támogatást fog nyújtani, mely egy szimpla objektum megjelenítéséhez elegendő.

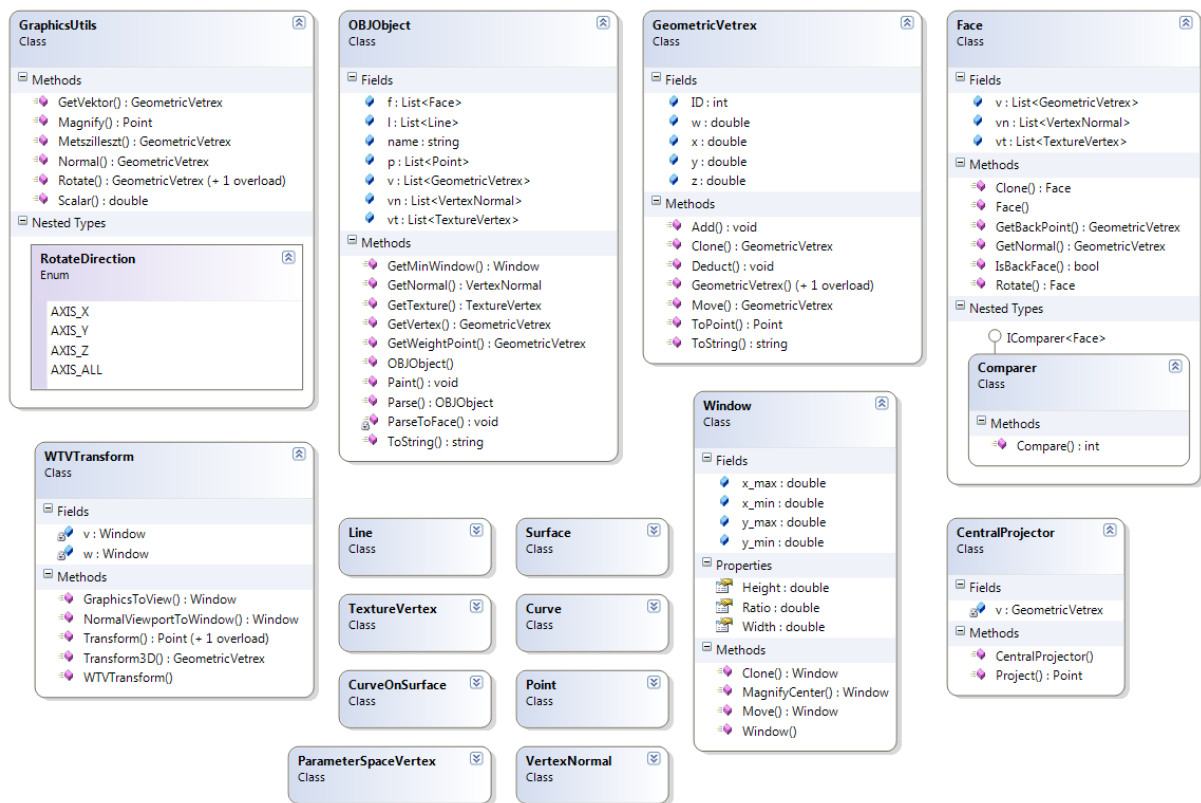
A szoftver fejlesztési folyamata

A szoftverrel szemben támasztott követelményeim:

- I. Általános leírás: A MyOBJViewer szoftver OBJ fájlok grafikus megjelenítésére szolgál.
- II. Általános követelmények:
 1. A szoftver képes legyen bármely háttértárolóról megnyitni egy fizikai (OBJ) állományt, és az abban reprezentált modellt 3 dimenziós (hatású) grafikaként megjeleníteni.
 2. A szoftverrel lehetőség legyen a modellt nagyítani, kicsinyíteni, illetve változtatni a szemszögöt (forgatni, dönteni a modellt).
- III. Rendszerkövetelmények:
 1. Operációs rendszer: Windows XP, Vista, 7 (.NET 3.5 Framework)
 2. Programozási nyelv: C# (.NET 3.5, MS Visual Studio 2008)

3. Célhardver: Nem szükséges figyelembe venni. Az adott operációs rendszert kielégítő konfiguráció.
4. Várható felhasználók száma: 1 fő.

A fenti követelmények alapján célszerűnek tartom egy Windows Form Application program fejlesztését. A későbbi fejlesztés végett az OBJ fájlformátum kezelését, feldolgozását végző komponenst egy külön függvény-könyvtárként fejlesztem (OBJCore.dll, 22. ábra).



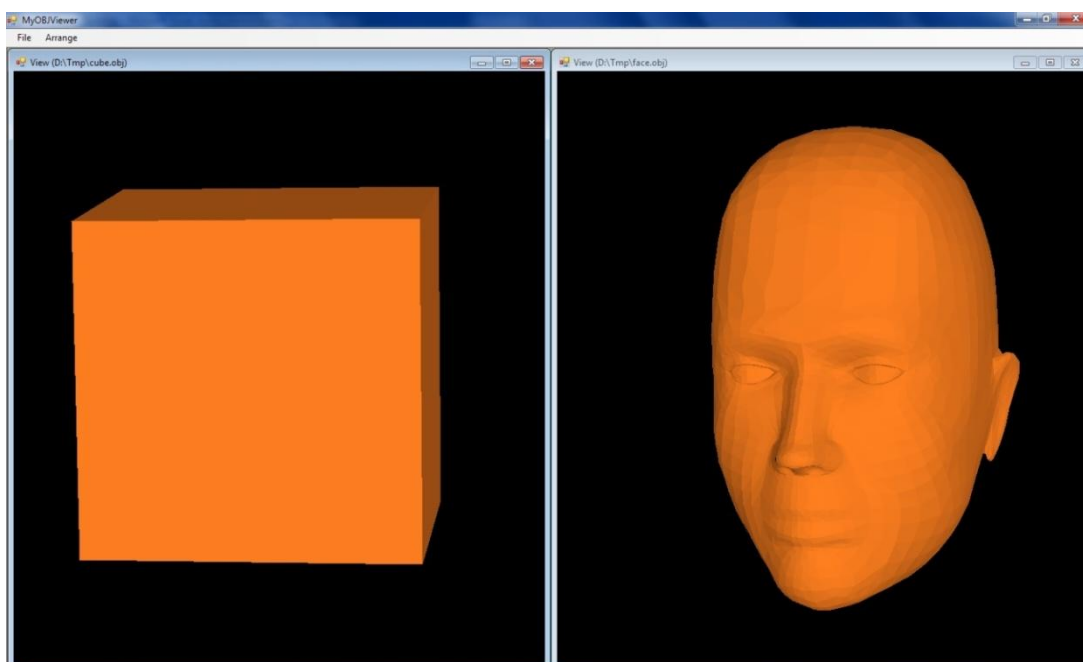
22. ábra, OBJCore.dll osztálydiagram, forrás: MyOBJViewer projekt

A függvénykönyvtár e verziója nem valósítja meg a teljes OBJ specifikációt, csak annak a poligonális részét (ez alatt a felületek poligonokkal történő közelítését értem). Szükségesnek láttam ebbe a függvénykönyvtárba beleintegrálni olyan eszközöket, mely az OBJ objektum (*OBJObject* osztály) megjelenítését segítik elő. Például a *CentralProjector* osztály egy példánya a modell centrális vetítésére szolgál, a *GraphUtils* osztály statikus metódusai pedig többek közt a forgatáshoz nyújtanak segítséget.

A program vázát, egy egyszerű *MDIForm* alkalmazás képezi. Azért választottam ezt a megoldást, mert így egyszerre több OBJ objektumot is meg tudunk jeleníteni, külön ablakokban, így akár a már előbb leírt két arcmodellező program eredményeit is össze tudjuk hasonlítani.

Mivel a függvénykönyvtár e verziójában még nem sikerült megvalósítani a textúrázást, ezért legszembetűnőbb hiányossága a programnak, hogy az objektumokat csak egy szín konstans árnyalásával képes megjeleníteni.

A 23. ábrán látható a program elkészült verziója.



23. ábra, *MyOBJViewer*, forrás: saját képernyőkép

A megjelenítő ablakban lehetőségünk van a modellt forgatni a bal egérgomb segítségével, közelíteni-távolítani az egér görgőjének segítségével, illetve áthelyezni a modellt a megjelenítési területen az egér jobb gombja segítségével.

A program egy következő változatában meg szeretném valósítani a textúrázott objektumok, illetve a paraméteres görbék és felületek támogatását.

Összefoglalás

Ha valaki manapság digitális fényképezéssel, vagy olyan modellezési eljárással szeretne foglalkozni, ahol a képalkotó eszköz valamilyen optikai elven működik, nélkülözhetetlen, hogy értse annak működését. Ehhez nagy segítség, ha tudjuk, hogy mi az a digitális fényképezőgép, milyen elven működik.

Azonban az eszközök ismerete még nem elegendő ahhoz, hogy valaki sikeresen alkalmazzon egy modellezési eljárást. Szükséges, hogy ismerje magát az eljárást, annak fázisait, alkalmazhatóságát, és esetlegesen hibát. Példának okáért, ha valaki modellezni (digitalizálni) szeretne egy elefántot, akkor nem szerencsés egy olyan eljárást választania, ahol a modellezendő tárgyat szinte stúdió körülmények között kell kezelnünk (David Laser Scanner).

Úgy gondolom, hogy mind a digitális fényképezésről, mind a modellezésről sikerült egy érthető és átfogó leírást adnom, mely viszonylag stabil alapul szolgálhat későbbi ismereteinkhez.

Miután már elsajátítottuk az alapismereteket erről a témáról, elkezdhetünk modellezni. A dolgozatban leírt két arcmodellező és a harmadik testmodellező program megfelelő eszközt ad akár egy kezdő, akár egy haladó felhasználó kezébe, mellyel kellő gyakorlatot szerezhethet elméleti tudása mellé.

Az egyes szoftverek elemzéskor csak a program szolgáltatásainak kis részét ismertettem. Egyrészt, mert a legtöbb szolgáltatás leírása megtalálható a fejlesztő cég weboldalán, másrészt terjedelemben többszöröse lett volna az aktuális leírásnak. Harmadrészt pedig a legtöbb funkció felesleges, csak kényelmi funkció, melyet egy átlagos felhasználó nem feltétlenül fog használni.

Az arcmodellező programokban elkészített projektjeim sikeresnek mondhatók, ugyanis csaknem helyes képet adtak a modellemről. Viszont a 3D SOM programban elkészített projektem nem sikerült teljesen, mivel a projekthez készített képek nem voltak elég jó minőségűek a megfelelő kontrasztelemezéshez.

Legvégül pedig, ha már túl vagyunk egy sikeres modellezési eljáráson, és előállt egy 3 dimenziós modellünk, tudnunk kell azt megjeleníteni.

Ehhez egy megoldás az általam készített MyOBJViewer nevű program. Sajnos idő szűke miatt nem nyújt teljes támogatást a program az OBJ fájlokhoz. Viszont egy olyan minimális OBJ támogatással bír a program, hogy az a modell egyszerű megjelenítését lehetővé teszi. Terveim között szerepel a program további bővítése, tökéletesítése.

Dolgozatomban bár csak az egész modellezési téma kis részét tudtam érinteni, úgy gondolom, hogy arról a kis részről érthető és részletes elemzést tudtam adni.

Köszönetnyilvánítás

Köszönettel tartozom témavezetőmnek, Dr. Tornai Róbert tanár úrnak, hogy mindenben támogatott a dolgozatom sikeres megírása érdekében.

Köszönöm barátomnak, Hörcsik Ákosnak, hogy felhasználhattam képeit az arcmodellező programok elemzése során.

És nem utolsó sorban köszönöm tanárainknak munkájukat, mellyel hozzájárultak tudásom gyarapodásához.

Irodalomjegyzék

- [1] Baki Ádám: Zoom in - A digitális fényképezés A-tól Z-ig + DVD melléklettel, SCOLAR Kft, Budapest, 2008, 6-9. oldal
- [2] FOTÓTECHNIKA, 2009. II. évfolyam 1.szám, Epizódok a fotográfia történetéből II. 42-45. oldal
- [3] Baki Péter: Fényképezés, Magyar Fotóművészek Szövetsége, 2007
- [4] Dr. Sevcsik Jenő, Gyakorlati fényképezés, Műszaki Könyvkiadó, Budapest, 1961, 7-8. oldal

- [URL1] <http://hu.wikipedia.org/wiki/F%C3%B3kuszt%C3%A1lvols%C3%A1g>
- [URL2] http://en.wikipedia.org/wiki/Frame_rate.
- [URL3] http://hu.wikipedia.org/wiki/3D_Modellez%C3%A9s
- [URL4] <http://www.david-laserscanner.com/index.php>
- [URL5] <http://www.creaform3d.com/en/handyscan3d/products/default.aspx>
- [URL6] <http://www.nagyformatumu.hu/hu/3d-szkenner>
- [URL7] http://www.looxis.com/en/k52.Welcome_Your-3D-Face-in-glass.htm
- [URL8] <http://en.wikipedia.org/wiki/Obj>
- [URL9] <http://www.facegen.com/modeller.htm>
- [URL10] <http://www.3dsom.com/index.html>
- [URL11] <http://www.3dsom.com/download/index.html>
- [URL12] http://en.wikipedia.org/wiki/Wavefront_Technologies
- [URL13] <http://local.wasp.uwa.edu.au/~pbourke/dataformats/obj/>