

GeoAR

Térvonatkozású információk 3D ábrázolása, és fúziója fényképekkel

TDK dolgozat

Készült: 2006. december 5.
Írta: Bolla Péter
Konzulens: Elek István

Absztrakt

Egy hegy csúcjáról körülnézve gyakran elgondolkozunk: melyik hegyeket, településeket látjuk? A térképeinkről le tudjuk olvasni az egyes földrajzi pontok neveit, és helyes tájolás után meg tudjuk határozni, merre keressük őket. A jellegzetes, feltűnő pontokat (hegycsúcsok, folyók, autóutak) így egyszerűen össze tudjuk párosítani a terepen és a térképen, és az így kialakult viszonyítási rendszerben többé-kevésbé meg tudjuk határozni a többi látható objektumot. Maradnak azonban olyan terepi információk, amelyek szemmel nem láthatóak, de érdekes lenne látni, hogyan viszonyulnak a terep többi részéhez. Ezek lehetnek akár erdőben húzódó utak, amelyeket a lombkorona miatt nem látunk, de lehet akár egy település határa is, amely a természetben nem is, csak virtuálisan létezik.

A GeoAR projekt ezen nem látható információk megjelenítésének problémáját járja körül: a projekt keretén belül térvonatkozású információk valamint kisebb-nagyobb területeket átfogó fényképek vagy egyéb képi felvételek egymásra vetítésének kérdéseivel foglalkozunk, valamint egy ezt megvalósító szoftver rendszer kifejlesztését célozzuk meg.

Ennek a dolgozatnak a keretében egy olyan szoftver termék kifejlesztését mutatjuk be, amely segítségével akárki elvégezheti a saját maga által készített fényképek, és saját maga által gyűjtött térinformatikai adatok egymásra vetítését. Minden lépésben azt tartjuk szem előtt, hogy rendszer használatához csak bárki által beszerezhető, hobbi eszközök legyenek szükségesek.

Első lépésben megvizsgáljuk, hogy a kiválasztott fejlesztési platform, a Java3D alkalmas-e a probléma megoldására. Ebben a fázisban elsősorban informatikai kérdésekre kell választ találni, figyelembe véve a feldolgozandó adatok minőségét, mennyiségét.

Második lépésben a megjelenítendő fényképek kalibrációját oldjuk meg. A Föld (vagy egyéb égitest) tetszőleges pontján, tetszőleges irányban készített képnek meg kell határozni a pontos pozícióját és orientációját egy megfelelő viszonyítási rendszerben, mert csak így tudjuk a virtuális adatokat megfelelően ráhelyezni a képre.

A harmadik lépés a térinformatikai adatok beolvasása a rendszerbe. Itt azt gondoljuk végig, hogy milyen típusú objektumok megjelenítésére alkalmas a módszer illetve a rendszer, és milyen módon érdemes megjeleníteni ezeket.

Negyedik lépésként összeállítjuk a szoftvert.

A dolgozat keretében végzett kutatás és a kifejlesztett szoftver alkalmas arra, hogy meghatározzuk, milyen irányban érdemes továbbfejleszteni a módszert, a rendszert.

1 Bevezető

Mai napra a számítástechnika, azon belül a 3D (3 dimenziós) ábrázolás olyan szintre fejlődött, hogy a valósággal szinte összehasonlítható minőségű képeket tudunk generálni, pusztán a látvány matematikai leírásával. A gyors fejlődést nagyrészt a játékipar generálta, ahol a elsődleges szempont a minél realisztikusabb ábrázolás, de a számítógépes vizualizáció minden szegmensében egyre elterjedtebben alkalmazzák eredményeit.

A térinformatikai alkalmazások is egyre gyakrabban használnak 3D megjelenítést. Így nem csak a síkban tudunk ábrázolni, hanem digitális domborzatmodellek felhasználásával valamely nézőpontból a valós domborzati viszonyoknak megfelelő 3D felszínre tudjuk kivetíteni az adatokat.

A 3D megjelenítés egyik érdekes alkalmazási területe a „virtuális valóság” (VR, *virtual reality*) megjelenítése. A VR rendszerekben a felhasználó fejre szerelt megjelenítő (HMD, *head mount display*), térbeli hangot reprodukálni képes hangrendszer, és a fej (nézőpont) mozgását követő érzékelők segítségével képes olyan látványt és hangzást előállítani, amely a felhasználó számára azt az élményt nyújtja, mintha valójában a látottak és hallottak alkotnák a valóságot.

A VR és a „valódi valóság” összemosása a „kiegészített valóság” (AR, *augmented reality*), amelyben a valódi világról készült képet vagy felvételt olyan információkkal egészítünk ki, amelyek közvetlenül a látványból nem derülnek ki.

A Geo3D is egy kiegészített valóság rendszer. A projekt alapvető célja, hogy nagy területeket átfogó tájképekre térinformatikai információt vetítsünk úgy, hogy az olyannak hasson, mint ha a feltüntetett információk a kép szerves részei lennének, mindeközben viszont a lehető legkevesebbet takarjon ki a valódi látványból.

2 Célkitűzés

Jelen dolgozat keretében egy olyan program létrehozása a cél, amely segítségével csak mindenki számára elérhető, hétköznapi eszközök felhasználásával készített fényképeket illetve rögzített térvonatkozású adatokat egymásra tudunk vetíteni. A hétköznapi képrögzítés eszköze a digitális fényképezőgép, amely ma már szinte minden háztartásban megtalálható; a térvonatkozású információk rögzítéséé pedig a kicsit kevésbé elterjedt, de könnyedén hozzáférhető eszköz, a kézi GPS műszer. Amennyiben valakinek ez utóbbi nem áll rendelkezésére, akkor is érdekes adatok birtokába juthat az interneten megtalálható nyilvános térinformatikai adatbázisok egyikén keresztül.

A mi szempontunkból a fényképezőknek egy tulajdonsága érdekes: a vízszintes látószöge. Ez határozza meg, hogy adott pontból a tér mekkora szeletét rögzíti a fényképezőgép. Míg felbontásban egyértelműen kijelenthető, hogy a nagyobb a jobb, addig látószögből nincs „ideális”, azt mindig a megjeleníteni kívánt terület mérete határozza meg. A mai kézi gépeken ez a zoom funkció, amely segítségével a fókusztávolságot (és ezzel a látószöget) állíthatjuk, általában fokozatmentes módon. Ez egyben előnye és hátránya is a dolognak. Az előnye az, hogy könnyedén az ideális méretű területről tudjuk a felvételt készíteni, hátránya viszont, hogy fokozatmentessége miatt nem számszerűsíthető a látószög. Az olcsóbb készülékeken még a végállások sem pontosan adottak, a köztes állások meg még a professzionális gépek optikáiról is csak ritkán olvashatóak le egzaktul. Emiatt megoldást kell találni a látószög meghatározására is.

A digitális fényképet hétköznapi használatban viszont leginkább a felbontása jellemzi. Egy, ma már kis felbontásúnak számító 2 megapixeles fényképezőgép 1600x1200 pixel oldalméretű képet tud készíteni (=1920000 pixel). Egy digitális fényképnek ha meg akarjuk határozni a felbontását (a műholdképeknél használatos értelemben), akkor azt csak egy-egy konkrét tárgytávolságra tudjuk megtenni az alábbi képlet szerint: $R = 2 * t * \tan(\alpha/2) / w$, ahol R a felbontás átlaga a képen, t a tárgytávolság, α a vízszintes látószög, w pedig a képszélesség pixelben. Vegyük egy fent említett 2 megapixeles gépet, amely marketingfelirata szerint 3x-os optikai zoomra képes. Ez megközelítőleg 18,72°-os vízszintes látószögnek felel meg. Ebből 1 km-es távolságra 20 cm-es felbontás adódik, 10 km-es távolságra pedig 2 m-es. Ezek az értékek is már elég jók, viszont fokozhatóak is, hiszen ma már általánosnak tekinthető az 5 megapixel, 2592x1944 pixel oldalmérettel (=5038848 pixel).

Természetesen nem csak a pixelek száma számít, nagyon lényeges még az objektív minősége, ugyanis egy rossz minőségű objektum esetén életlen, homályos képet kapunk, azaz hiába foglal el a valóság egy részlete több pixelt a képen, ha azok a pixelek mind azonos színűek, akkor nem rögzítünk több információt. Ezt azonban a köznapi használatban nem számszerűsítik.

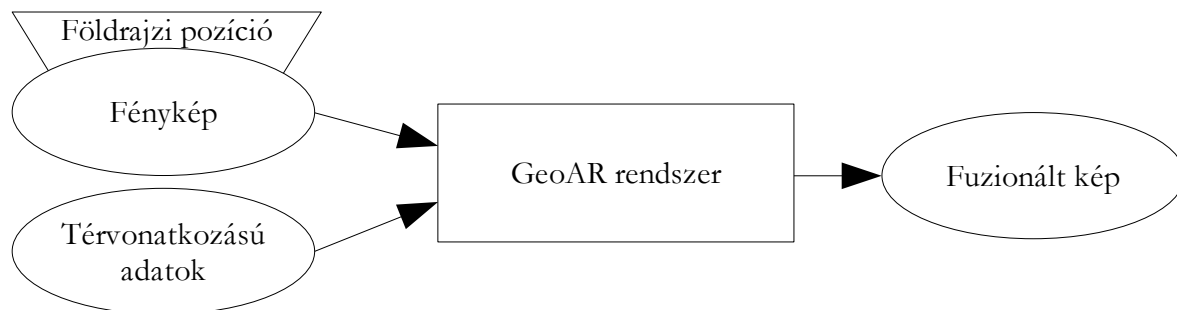
A fénykép készítésének helyét pontosan ismernünk kell. Ehhez az esetek többségében szükségünk van egy kézi GPS vevőre, amely segítségével kisebb-nagyobb pontossággal meg tudjuk határozni az aktuális földrajzi koordinátánkat. Bizonyos esetekben meg tudjuk határozni a pozíciónkat GPS nélkül is, ha éppenséggel egy háromszögelési alappont mellől, vagy egyéb ismert földrajzi koordinátájú ponton állva fényképezünk. Ebben az esetben akár pontosabb koordináták birtokába is kerülhetünk, hiszen a különböző jelenségek miatt a GPS rendszer segítségével átlagban csak ~15m-es pontossággal tudjuk a helyzetünket meghatározni.

Az ábrázolni kívánt térvonatkozású adatokat saját magunk is rögzíthetjük, vagy egyéb adatforrásokat használhatunk. Ha saját magunk rögzítjük, ismét GPS-re lesz szükségünk. Ilyen eset például, ha egy túra útvonalát szeretnénk megjeleníteni. A program bemeneteként alkalmazott adatok formátumául a GPX (GPS Exchange) formátumot választottuk. Ez nem a professzionális térinformatikai szoftverek formátuma, de a kitűzött célnak pont megfelelő. A legtöbb ingyenes GPS adatfeldolgozó program képes ennek a formátumnak írására, olvasására, és saját értelmező megvalósítása sem problémás.

Az adatokat nem csak GPS-szel rögzíthetjük, hanem az internetről is letölthetünk érdekes pontokat (POI, point of interest) és útvonalakat – például turistautakat, túraútvonalakat, – vagy akár valamelyik ingyenes GPS adatkezelő programmal saját magunk rajzolhatjuk meg őket.

3 Rendszerelemek

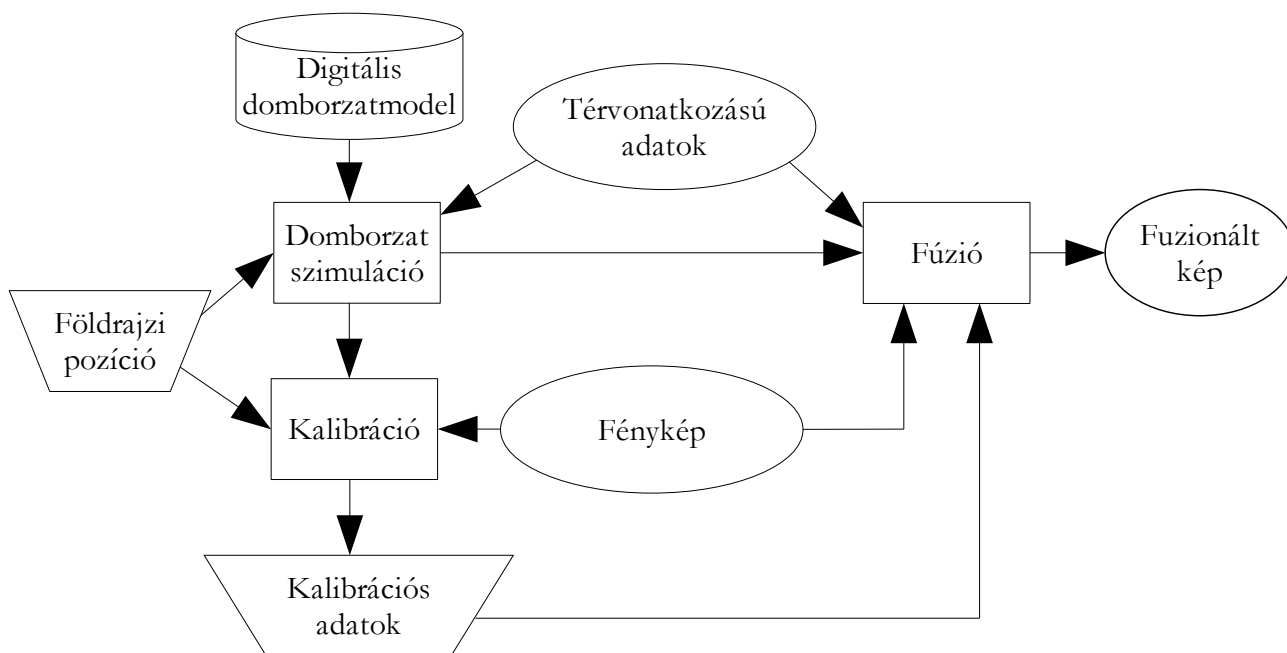
Az 1. ábrán látható a rendszer blokkvázlata. Bemeneti adatok a pozíció információval ellátott fénykép, és a megjeleníteni kívánt térvonatkozású adatok, kimeneti pedig a fuzionált kép, azaz az eredeti fénykép és térvonatkozású adatok egymásra vetítése.



1. ábra: A teljes rendszer blokkvázlata

Az alap feladat, hogy a térvonatkozású információkat 3D módon jelenítsük meg olyan módon, hogy úgy nézzen ki, mintha abból a pontból néznék, ahol a fénykép készült. Eddig a probléma egyszerű: a térvonatkozású adatok vonatkozási pontjait ismerjük (hiszen ezek által adott maga az információ), ezt valamilyen 3D megjelenítő rendszerbe betáplálva el is készülnénk a látvánnyal. Első probléma már itt jelentkezik: ez a módszer önmagában nem veszi figyelembe, a domborzatot, azaz hogy a hegyek kitakarják a mögöttük lévő völgyeket. Ennek figyelembevétele nélkül minden pont, minden útvonal teljes egészében látszódná, még akkor is, ha valójában egy hegy nem látható oldalán húzódna.

A következő probléma az így megkapott 3D-s kép és a fénykép összeillesztése. Nem elég ugyanis ismernünk a fénykép készítésének pontos koordinátáit, hanem szükségünk van a képtengely orientációjára is (azaz arra az irányra, amerre a fényképező „néz”, valamint a kép teteje felé mutató irányra.) Ezeknek a problémáknak a megoldására egy-egy további elemet tudunk felvenni a blokkvázlatba (2. ábra)



2. ábra: Részletes blokkdiagram

4 A Java3D keretrendszer

Világunk 3 dimenziós, egy kép viszont csak 2 dimenziós. A 3 dimenziós információkat ezért 2 dimenzióba kell vetíteni úgy, hogy az emberi látás folyamán az agyban 3 dimenziós hatást keltsen. Ilyen vetítési módszer a perspektív vetítés, amely a 3 dimenziós hatást azzal éri el, hogy a távolabb lévő objektumok kisebbnek látszanak. Minthogy egy fényképezőgép által készített fényképen is lényegében ugyanez történik, ezért a mi számunkra ez pont megfelelő eljárás.

A számítógépes grafikában a 3D ábrázolás igen elterjedt, így számos 3D keretrendszer közül válogathatunk, amelyek különböző technikákat is képviselnek. A megjelenítési módszereknek mindegyiknek megvan a maga felhasználási területe. Általában a realiztikus hatás és a feldolgozási sebesség a két ellentétes pólus, amely között meg kell találni az éppen megfelelő arányt. A mi esetünkben – mivel többek között nem használunk bonyolult megvilágítási módszereket és nincs szükségünk tükröződéskezelésre, – nem követelmény a fotorealistikus megjelenítés. Annál inkább lényeges a gyors megjelenítés, amelyre a kalibráció folyamán szükségünk lesz. A legtöbb számítógépes játék által is használt *scanline rendering* (soronkénti feldolgozás) módszer épp megfelelő a számunkra is. Ezzel a módszerrel a poligonokra illetve vonalakra bontott virtuális világot a képernyőn soronként dolgozza fel a rendszer, és minden pixelnek a színét csak az a poligon határozza meg, amelyhez tartozik.

A választás a Java3D keretrendszerre esett. A Java3D jelenetgráf alapú rendszer, amely magas szintű megközelítésnek számít. A jelenetgráf egy fa struktúra, amely gyökerében a virtuális világ egy nagy pontossággal megadott pontja áll. Csomópontjai transzformációk, megjelenítendő objektumok, nézőpontok, fényforrások, stb. Így jól elkülönülő részekre lehet bontan a virtuális teret, és egymástól függetlenül szépen le lehet írni az egyes objektumokat. (Alacsony szintű megoldás az, amikor közvetlenül poligonokat adunk át a megjelenítő rendszernek.)

A Java3D – mint a legtöbb 3D grafikus rendszer – a teret Descartes koordinátarendszerben írja le. A virtuális világot 3, derékszögű tengely által kifeszített tér alkotja. Alapértelmezésben nincsen kitüntetett irány a rendszerben. Hogy egy generált képen mi lesz a „felé”, „jobbra”, „balra” irányok, az mint a jelenetgráfban elhelyezett kamera orientációjától függ.

A virtuális világ egy nagy felbontású tér, ahol egy pont mindhárom koordinátáját 256 bites fixpontos értékkel jellemezhetjük. Ha ebben a rendszerben a virtuális 1 hosszú vektort a valóságban

1 m hosszú vektornak feleltetünk meg, akkor a virtuális térben a Planck-hossznál is kisebb illetve az univerzumnál is nagyobb távolságok nagy pontosságú definiálására is mód van. Egy ilyen nagy felbontású pontba kell elhelyezni a jelenetgráfunk gyökerét, ami köré fel kell építenünk a 3D modellünket. Mivel csak egy jelenetet építünk, ezért az egyszerűség kedvéért ezt az origóba helyezzük el.

A földi koordinátákat – a földrajzi szélesség, hosszúság és tengerszint feletti magasságadatokat – át kell transzformálnunk a virtuális világ XYZ terébe. Erre a földközéppontú, földhöz rögzített XYZ koordinátarendszert választottuk, mert ez a transzformáció általánosan ismert, elfogadott.

Ezzel jelentést adtunk az irányoknak: az X tengely az egyenlítő és a középmeridián metszéspontján, az Y tengely az egyenlítő és a keleti 90°. hosszúsági kör metszéspontján, a Z tengely pedig az északi sarkon halad át. Ez egy standard jobb sodrású rendszer.

A Java3D virtuális világában a nézőpontok viszont máshogy értelmezik az irányokat. Az X tengely a kép aljával és tetejével párhuzamos, és jobb irányba pozitív, az Y tengely a kép két oldalával párhuzamos és felfele pozitív, a Z tengely pedig a kép síkjára merőleges, és a kép középpontjából a néző felé mutat. Ezt figyelembe véve kell megalkotni a nézőpont transzformációját.

A program fejlesztése során kiderült a keretrendszer egy korlátja. Mivel úgy számolunk, hogy a virtuális világban 1 egység hossz az a valóságban 1 méter, ezért az objektumok az origótól megközelítőleg 6371000 egység távolságra kerülnének, koordinátaik ebben a nagyságrendben mozognának. Egyelőre ismeretlen, de a Java 3D rendszerben lévő okok miatt valószínűleg kerekítési hibák kumulálódtak, és az egyes objektumok egymáshoz viszonyítva pontatlanul jelentek meg. A megoldást erre az jelentette, hogy az földrajzi adatok XYZ adatokká transzformálása után az eredményt eltoltuk úgy, hogy a fénykép készítésének transzformált pontja kerüljön az origóba. Így minden koordináta több nagyságrenddel kisebb értéket vesz fel, és a hiba nem jelentkezik.

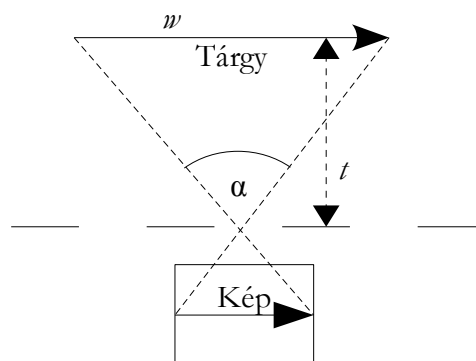
5 Felhasznált adatok

A program által felhasznált adatok az alábbiak: a fénykép, a fényképhez tartozó pozíció, a megjelenítendő térvonatkozású adatok valamint a digitális domborzatmodell.

5.1 A fénykép

A fénykép egy JPG formátumú digitális kép, tetszőleges oldalaránnyal és felbontással. A felbontásban természetesen a minél nagyobb, annál jobb elv érvényesül, de mivel a program közvetlenül nem dolgozza fel a képet, csak megjeleníti illetve rárajzol, ezért konkrét, számszerűsített értékek meghatározása nem lényeges. Egyetlen adat szükséges, amely a fényképezővel függ össze, ez a vízszintes látószög. Mivel ezt számszerűen nagy valószínűséggel nem tudjuk leolvasni a fényképezőről, ezért meg kell mérnünk.

A méréshez a 3. ábrán látható egyszerűsített modellt használjuk fel.



3. ábra: A látószög megmérése

A mérés menete: keresünk egy objektumot, amelynek ismert (vagy jól megmérhető) a szélessége (az ábrán: „Tárgy”, szélessége w). Ezt az objektumot beállítjuk a keresőben úgy, hogy az a kép

középvonalában legyen, és teljes egészében kitöltse azt. Ekkor le kell mérni a fényképező elvi képsíkja és a tárgy közötti távolságot. Ezt jelöljük az ábrán t -vel. w -ből és t -ből könnyedén meg tudjuk határozni a látószöget, amelyet α -val jelöljük az alábbi képlettel: $\tan\left(\frac{\alpha}{2}\right) = \frac{w/2}{t}$.

5.2 A pozíció adatok

A pozícióadatokat vagy kézi GPS műszerrel mérjük, vagy egyéb módon megállapított földrajzi koordinátákkal adott ponton készített kép esetén máshonnan ismerjük. A lényeg, hogy mindkét esetben a földrajzi szélesség, hosszúság és tengerszint feletti magasság adatokat ismerjük. Ezeket a programba számszerűen, WGS84 dátumon értelmezve kell megadni. Minthogy a rendszer az adatokat átszámítja XYZ rendszerbe, amelyben a különböző dátumok között egyszerű az átszámítás, ezért nem bonyolult kibővíteni a rendszert más dátumokon mért adatok fogadására.

5.3 Orientáció adatok

A fényképezőgép orientációja lényeges információ, azért szükséges, hogy a virtuális világban létrehozott objektumok és a való világ képe megfelelően illeszkedjenek egymásra. Egy megfelelően átalakított teodolit, amelyben a távcső helyére a fényképező kerülne, tökéletes megoldás lenne, hogy számszerű adatokat kapjunk az orientációról. Mivel ma hétköznapi használatban nem érhető el ilyen, vagy hasonló felszerelés, ezért ezeket az adatokat a rendszer nem is várja a felhasználótól. Más megoldást kell választani, és ez a manuális kalibráció, amely során a felhasználó a virtuális és a valódi világ képét egymással fedésbe hozza.

5.4 Térvonatkozású információk

A térvonatkozású információkat GPX fájlformátumban várja a program. A GPX eredeti célját tekintve GPS rendszerrel rögzített pontok és útvonalak tárolására kifejlesztett állomány. XML alapú, ebből következően könnyen készíthető hozzá feldolgozó modul.

Az egyes útpontok a földrajzi szélességükkel és hosszúságukkal adottak. Ezen felül minden pontoz megadható többek között a tengerszint feletti magassága, az útpont neve, rövid leírása és a pont felvételének időpontja. Ezek a legáltalánosabban használt attribútumok, amelyeket egy-egy ponthoz tárolni szoktak.

A rögzített útvonalak (*tracklog*) ilyen pontok listáival adottak. Egy-egy útvonalhoz tartozik egy név attribútum, valamint több útvonal szegmens, amelyek pedig útpontokból állnak. Ezekhez az útpontokhoz a földrajzi koordinátán kívül általában csak a tengerszint feletti magasságuk és a felvételi időpontjuk adott. Ábrázoláskor a szegmenseken belül az útpontokat össze szokás kötni, a szegmensek között viszont nem. A több szegmens jelenléte a készülék kikapcsolását vagy a műholdjel elvesztését jelentheti.

5.5 Digitális domborzatmodell

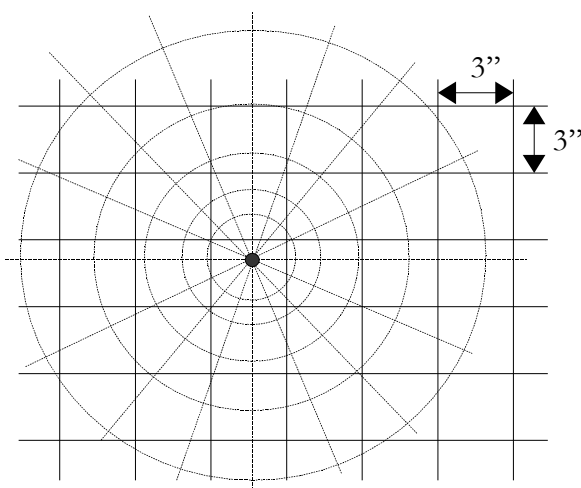
Elsőre nem feltétlenül világos, hogy miért is van szükségünk digitális domborzatmodellezésre. Ha ugyanis a rögzített vagy internetről megszerzett ábrázolandó útvonalak vagy útpontok magassági adatokat is tartalmaznak, akkor már kellő információ áll rendelkezésünkre, hogy azokat az XYZ térbe helyezve megfelelő módon a fényképre transzformáljuk. Amit nem vennénk így figyelembe, az a domborzati kitakarás, azaz hogy például egy hegy gerince mögé nem látunk be, és ha ott megy egy út, azt nem rajzolhatjuk rá a fényképre, mert a fényképen a tájnak nem a megfelelő része látszódna az adott helyen. Mivel a fénykép sokkal több információt nyújt mint az útvonal rajza önmagában, agyunk az alapján építené fel a 3D észlelést. Így viszont a hegy nem látható oldalán lévő, de berajzolt útvonalakat a látható oldalra helyezné, hibásan.

Ennek kiküszöbölésére szükséges a virtuális térben a domborzatot szimulálni, és így kiszámolni, hogy mely útpontok, és az útvonalak mely részei láthatóak, és melyek nem. A domborzat szimulációjához pedig szükségünk van valamilyen digitális domborzatmodellre. Erre a célra az ingyenesen hozzáférhető SRTM (*Shuttle Radar Topography Mission*) adatbázisra támaszkodunk. Az SRTM

az NGA (*National Geospatial-Intelligence Agency*, az amerikai katonai térképészeti intézet) és a NASA (*National Aeronautics and Space Administration*, az amerikai űrügynökség) közös programja volt, amelyben a föld szárazföldjeinek 80%-át felmérték, és ingyenesen elérhetővé tették. Ennek köszönhetően a 56° déli szélességtől a 60° északi szélességig minden szárazföldi területről 3" x 3"-es foktrapézonkénti átlagolásban van magassági adatunk. Ez Magyarország területén megközelítőleg 93m x 63m-es téglalaponkénti átlagolást jelent. Az adatok HGT fájlokban hozzáférhetőek, amelynek tartalma pontosan definiált. Egy fájlban egy 1° x 1°-os foktrapéz adatai vannak, 1201 sorban, soronként 1201 érték. Így minden foktrapéz között egy sor és egy oszlop átfedés van. A programnak mindig csak azokra a fájlokra van szüksége, amelyekből konkrétan adatokat akar használni. Így nem szükséges az összes fájl letöltése.

Ebből az adathalmazból kell a 3D megjelenítő számára az XYZ térben egy felületet létrehozni. Legegyszerűbb módszer az, ha minden egyes egzaktul adott 3" x 3"-as foktrapézt átranszformálunk XYZ-be, és közéjük egy négyszögletű felületet definiálunk. Ezeket a felületeket a 3D rendszer alpból tudja kezelni. Ennek a megoldásnak a nagy hátránya, hogy ha mind az 1200x1200 foktrapézt meg akarjuk jeleníteni, akkor túl nagy adathalmazt kapunk, amelynek a megjelenítése lassú. Ha le is vágnánk a megjelenítésben kicsi szerepet játszó négyszögeket, akkor is túl sok lenne még az adat, és a távolabbi részek feleslegesen lennének túlzottan finoman részletezve.

Ehelyett a magassági adatokat közvetlenül nem használjuk fel a megjelenítéskor, hanem ezek alapján matematikai módon definiálunk egy felületet, amelyet egy nekünk jobban megfelelő módon újra mintavételezünk. (4. ábra)



4. ábra: magassági adatok újramintavételezése

A 4. ábrán látható elrendezés az újramintavételezés módját szemlélteti. Eredetileg egy négyzetrács-hálónk van, amelyet folytonos vonallal jelölünk. A négyzetrács metszéspontjaiban adott magassági adatok alapján egy matematikai modellt hozunk létre, amely segítségével tetszőleges pontban meg tudjuk becsülni a magasságot. Legegyszerűbb eset ha lineáris Bézier felületet definiálunk. Ebben az esetben minden pont magassága csak a hozzá legközelebb eső 4 pont magasságától függ, és lineárisan változik. Ebben az esetben az egyes négyszögek megtörve csatlakoznak egymáshoz, a felület nem lesz teljesen folytonos. Magasabb szintű felületek alkalmazásával ezt ki tudnánk küszöbölni, de akkor további csomópontok megadására lenne szükségünk. Ez megoldható lenne: a 4 legközelebbi ismert pont és az azok szomszédai segítségével ki tudnánk számolni a másodfokú felülethez szükséges további 5 vagy harmadfokúhoz szükséges további 12 csomópontot. Mivel azonban az újramintavételezés folyamán ismételtelen elveszne ez a folytonosság, ezért a matematikai felületen ez nem követelmény.

Ebből a matematikai modellből minden pontnak kiszámítható a felületi normálisa is, amely a megvilágítási értékek kiszámolása miatt szükséges.

Az újramintavételezés során ebből a matematikai felületből számítjuk ki tetszőleges pont magasságát. Egy olyan felületet hozunk létre, amelynél az ismert magassági pontok a nézőpont körüli koncentrikus körök és sugárirányú egyenesek metszéspontjaiban vannak, a megjelenített poligonokat

pedig a két sugár és két koncentrikus kör által kijelölt négy pont közé illesztjük be. A perspektív vetítés tulajdonságai miatt a szétnyíló sugarak a 2 dimenzióba vetített képen párhuzamosnak látszanak. Ezzel azt érjük el, hogy a közelebbi és a távolabbi részeken is látszólag ugyanannyi magassági pontunk ismert.

A program fejlesztése során külön figyelmet fordítottunk arra, hogy könnyedén lecserélhető legyen az SRTM adatbázis egy nagyobb pontosságot biztosító adatbázisra. Ilyen Magyarország területére ingyenesen nem elérhető, de létezik (DDM-10, DDM-50)

6 Kalibráció

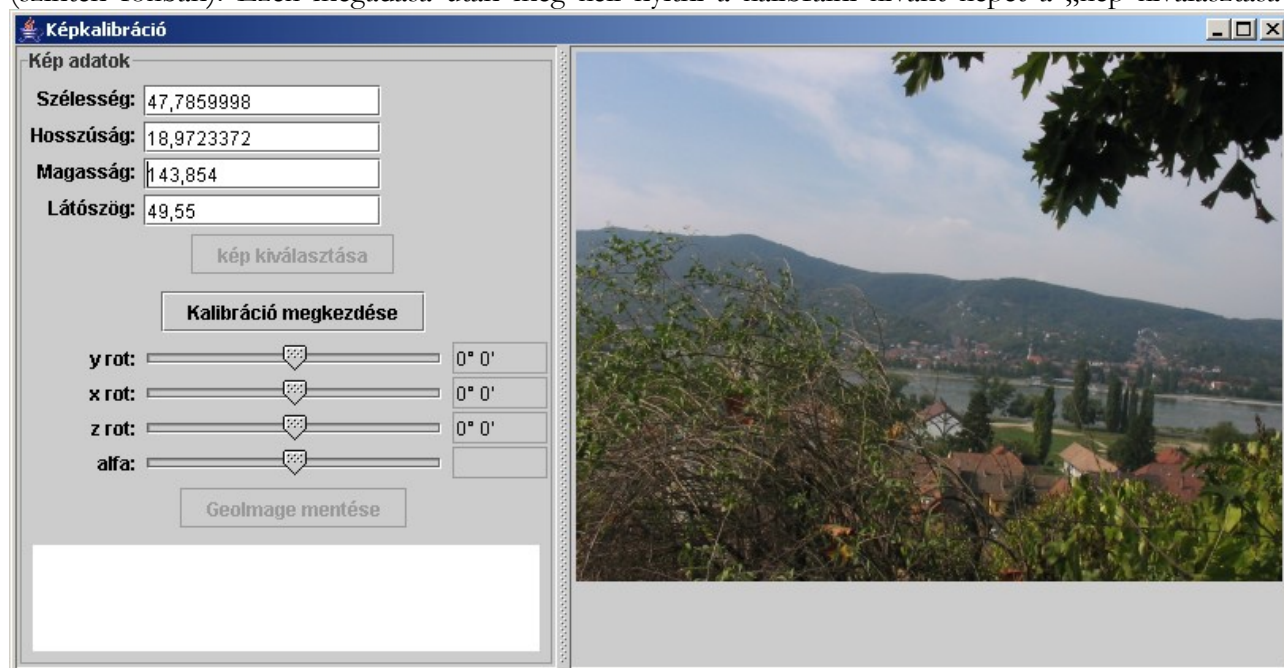
A virtuális világban a fényképező helyét két transzformációval írjuk le. Az első a keretet (koordinátarendszert) átviszi a nézőpont helyére, ahol a fénykép készült, (ezt az adatot olvastuk le a GPS-ről a fénykép készítésének pozíciójában,) valamint beállítja, hogy a fényképező egy jól meghatározott irányba nézzen. A kalibráció során ehhez az irányhoz képesti elforgatásokat állítunk be, amelyekkel a nézőpont tetszőleges orientációját el tudjuk érni.

Az alap irány olyan, hogy a földrajzi északi pólus felé néz a kamera, a felfele irány pedig a „szokványos” felfele iránynak felel meg, azaz az ellipszoid felületére merőleges, és a föld középpontjától nézve kifelé mutat.

Ezt az irányt transzformáljuk 3 elforgatással. Elsőként az Y tengely körüli forgatást hajtjuk végre. Ez olyan, mint amikor a fényképezőt az állványon a vízszintes sík mentén körbe forgatjuk. Értéke az irányszög. A második elforgatás az X tengely mentén történik, ez az előző példánál maradva annak felel meg, amikor a fényképezőt fel-le állítjuk. Ez a magassági szög. A harmadik elforgatás Z tengely körüli. Ez az optikai tengely körüli elforgatásnak felel meg. Ha precízen vízszintbe helyezhető állvánnyal rendelkezünk, amelyre a fényképezőt is fel lehet szerelni úgy, hogy a kép alja és teteje a vízszintes síkkal párhuzamos legyen, akkor erre a harmadik elforgatásra nincs szükségünk. Ha kézben tartva készítettük a képeket, akkor viszont szinte bizonyos, hogy mindhárom elforgatásra szükségünk lesz.

6.1 A kalibráció menete

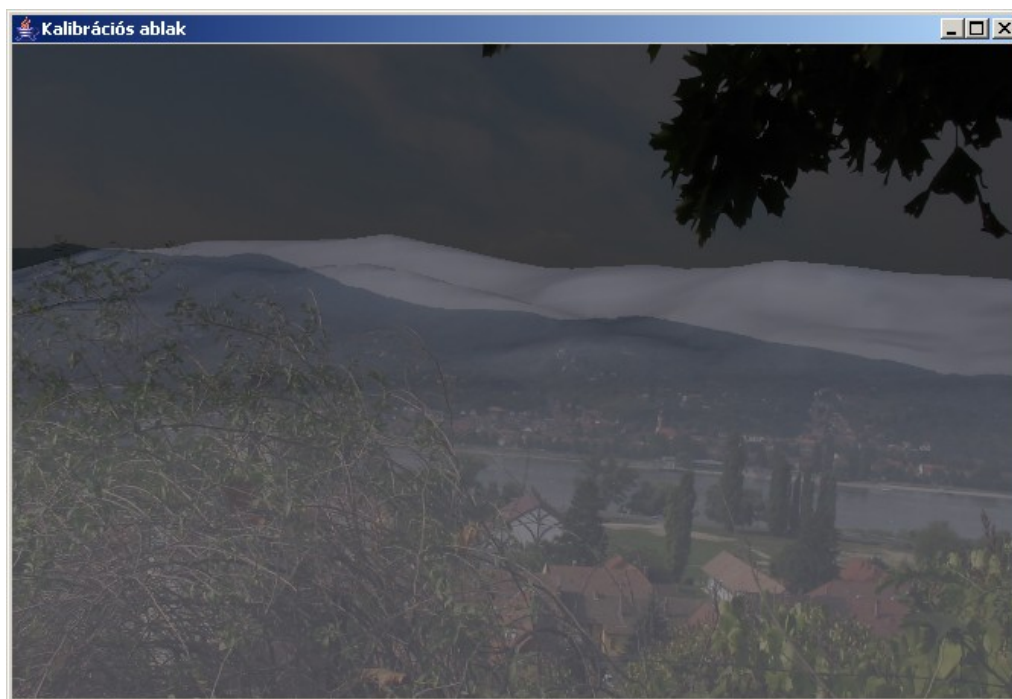
A kalibráció a kalibrációs alkalmazással történik. (5. ábra) A program elindítása után meg kell adni a földrajzi koordinátákat (fokban), a tengerszint feletti magasságot (méterben) és a látószöget (szintén fokban). Ezek megadása után meg kell nyitni a kalibrálni kívánt képet a „kép kiválasztása”



5. ábra: A kalibrációs program vezérlőképernyője

gombra történő kattintással. Amikor kiválasztottunk egy képet, akkor az az ablak bal felén meg is jelenik.

Ezután a „kalibráció megkezdése” gombra kattintva kezdhethetjük meg magát a beállítási folyamatot. Egy új ablak nyílik meg, amelyben a 3D domborzatmodell jelenik meg, amelyre átlátszósági arány szerint rávetül a kiválasztott fénykép. (6. ábra)



6. ábra: A kalibráció alapja: a domborzatmodell

Mivel a virtuális világban lévő „szem” látószöge megegyezik a fényképezéskor használt látószöggel, ezért ha a fénykép az ablakot szélességében pontosan kitölti, akkor pontos orientáció beállítás esetén a generált digitális domborzatmodell és a fényképen látható domborzat egybeesnek. A kalibráció során ezt a megfelelő orientációt kell beállítanunk. A jobbra és balra gombokkal az Y körüli forgatást állítjuk, a fel és le gombokkal az X körülit, a *Delete* és *Page Down* gombokkal pedig a Z tengely körülit. Alapból minden leütés 1° elfordulást eredményez, de ez nem biztosít elég nagy pontosságot, ezért ha az *Alt* kiegészítő billentyűt is nyomva tartjuk, $1'$ -et tudunk léptetni. Így már elegendő pontosság érhető el. A fénykép átlátszóságát a vezérlőablak „alfa” csúszkájával tudjuk állítani.

A kalibráció alapja tehát a domborzati összeegyeztetés. Először állítsuk az átlátszóságot 100%-osra. Így csak a domborzatmodellt látjuk. Keressük ki azt a részt, amelyik a képen látható domborzathoz hasonlít. Ekkor az átlátszóságot vegyük vissza, 50-70% közöttre úgy, hogy a fényképen és a generált képen is jól látszódjon a domborzat kontúrja. (A generált képen a domborzat szürke, és világosabb, mint az ég. A rendes képen várhatóan a domborzat sötétebb mint az ég. Ez az ellentétes kontraszt biztosítja, hogy a domborzat kontúrja jól látható legyen.) Ezután illesszük pontosan egymásra a domborzat kontúráját. Elsősorban a csúcsokra és völgyekre érdemes koncentrálni. Figyeljünk arra, hogy a domborzatmodell nem tartalmazza teljes egészében a fák lombkoronája által képviselt magasságot, a képen viszont egy erdővel borított területen mi azt látjuk. Emiatt a csúcsok egymásra illesztésénél a generált domborzati képet lehet, hogy kicsit lejjebb kell állítanunk, mint ahogy a fényképen látszik. Problémát okozhat még az is, hogy a domborzatmodellen a hirtelen kiemelkedő csúcsokat az átlagolás eltüntetheti.

Elképzelhető, hogy a korábban leírt módon meghatározott látószög nem teljesen pontos érték. Ennek a tünete lehet például, ha a képen az egyik csúcsot összeillesztjük a generált képpel, és egy következő csúcs jó irányban van, de nem esik egybe a generált kép csúcsával. Ha a fényképen ilyen értelemben túl „sűrűn” vannak a csúcsok, akkor a látószöget növelni kell, ha viszont a generált képen vannak túl sűrűn, akkor csökkenteni kell a látószöget.

Amikor a lehető legpontosabban elvégeztük a kalibrálást, akkor a „GeoImage mentése” gombra kattintva elmenthetjük az adatokat. A kimenet egy XML fájl, amely tartalmazza a fényképhez tartozó összes lényeges adatot: a fájl nevét, földrajzi pozícióját, orientációját és a látószöget. Ebből a fájlból később könnyedén ki tudjuk nyerni az adatokat, hogy a képre a megfelelő módon vetíthessük az információt.

7 A kép generálása

A kalibrált kép és a térvonatkozású adatok egymásra vetítése az ImageGenerator alkalmazással történik. Az program parancssoros, nincsen felhasználói felülete. Paraméterezése az alábbi:

```
ImageGenerator -i <inputXml> -o <outputImage> -d <inputData>
```

Az inputXml fájl a kalibráló programmal elmentett GeoImage adatfájl. Ez tartalmazza a fénykép elérési útvonalát, és a hozzá tartozó pozíció, orientáció és látószög adatokat. Az outputImage paraméter a kimeneti fájl nevét határozza meg. A kimeneti fájl a bemenetivel megegyező méretű JPG állomány lesz. Az inputData egy GPX fájl, amely a megjelenítendő térvonatkozású adatokat tartalmazza.

A program lefutásával a kimeneti fájl tartalmazza az összevetített képet.

A képen az útvonalakat piros vonal jelöli, az útpontok ábrázolására viszont több lehetőségünk van: ábrázolhatjuk gömbként, vagy árbocrúdként. A 7. ábrán két mintaképet láthatunk, az egyiken a pontok az egyik módszerrel, a másikon a másik módszerrel kerültek ábrázolásra. Különbség még a két kép között, hogy eltérően vannak kalibrálva. Feltűnhet, hogy a felső képen az út a gerincen halad, míg az alsón kicsit lejjebb. Az alsón viszont a nagymarosi rév kikötőjében felvett pont pontosan a helyén van, a felsőben viszont nem.



7. ábra: Eltérően kalibrált, különböző jelet használó képek

Nem tökéletes a kép bal oldalán a hegyre felhúzódó turistaút nyomvonala sem. Annak a völgyben kéne mennie, ehelyett viszont a hegy oldalában fut.

8 Értékelés

Az előbb említett pontatlanságokból arra lehet következtetni, hogy a rendszer valamely pontja nem tökéletes. A kapott végeredmény számos hibával terhelt. Sajnos a TDK dolgozat keretében már nincs módunk a problémák feltárására. Több probléma elképzelhető:

- A digitális domborzatmodell pontatlanságából adódhat az eredmény pontatlansága.
- A GPS által rögzített útvonaladatok pontatlanságából adódhat az eredmény pontatlansága.
- A kalibráció a felvázolt eszközökkel és módon nem megvalósítható elég pontosan.
- A 3D rendszer és a fényképezőgépben létrejövő vetítés a vártnál lényegesebb eltéréseket mutat.

E kérdések vizsgálatával lehet a probléma felgöngyölítéséhez hozzákezdeni. Ezeken kívül számos továbbfejlesztést és vizsgálatot lehet a rendszeren végrehajtani:

- Az útpontok és az útvonalak jelölésére további számos megoldást ki lehet találni. Ezeknek a használhatóságát lehet vizsgálni.
- Az útpontok és útvonalak feliratozásának kérdését lehet vizsgálni.
- Automatikus orientációmeghatározás kidolgozása számszerű méréssel (teodolitra szerelhető fényképező) vagy matematikai módon (képen szereplő ismert pontok megadásával, és vektorműveletekkel)
- Mozgóképek feldolgozása és kiegészítése.
- Nézőpont mozgatása.
- Egy vagy több helyről készített több kép egyidejű feldolgozása.

Irodalomjegyzék

[1] 3D computer graphics. *Wikipedia, The Free Encyclopedia*. 2006 december 3, 04:48 UTC.
Wikimedia Foundation, Inc. 2006 december 5;
http://en.wikipedia.org/w/index.php?title=3D_computer_graphics&oldid=91727888

[2] Java3D API documentation;
<http://download.java.net/media/java3d/javadoc/1.4.0/index.html>

[3] Introduction to the Java 3D API
<http://download.java.net/media/java3d/javadoc/1.4.0/javafx/media/j3d/doc-files/intro.html>

[4] Bézier curves and Bézier surfaces, Piet Verplancken , 1998 december
<http://home.tiscali.be/piet.verplancken3/bezier/bezier.html>

[5] Geodetic Datum Overview, Peter H. Dana
<http://www.colorado.edu/geography/gcraft/notes/datum/datum.html>

[6] Shuttle Radar Topography Mission, NASA
<http://www2.jpl.nasa.gov/srtm/>