

KORSZERŰ GEOINFORMATIKAI MÓDSZEREK AZ ERDÉSZETBEN

Egy geoinformációs rendszer fejlesztésének tudományos eredményei

DOKTORI (PhD) ÉRTEKEZÉS

Czímber Kornél

Nyugat-Magyarországi Egyetem

Erdőmérnöki Kar, Sopron

Erdészeti tudományok doktori program, E/3. alprogram

Témavezető: Dr. Bácsatyai László

Sopron

2002

Korszerű geoinformatikai módszerek az erdészetben
Egy geoinformációs rendszer fejlesztésének tudományos eredményei

Értekezés doktori (PhD) fokozat elnyerése érdekében

Készült a Nyugat-Magyarországi Egyetem Erdészeti Tudományok
doktori program E/3. alprogramja keretében

Témavezető: Dr. Bácsatyai László

Elfogadásra javaslom (igen / nem)

.....
alíírás

A jelölt a doktori szigorlaton % -ot ért el,

Sopron

.....
a Szigorlati Bizottság elnöke

Az értekezést bírálóként elfogadásra javaslom (igen / nem)

Első bíráló: Dr. (igen / nem)

.....
alíírás

Második bíráló: Dr. (igen / nem)

.....
alíírás

A jelölt az értekezés nyilvános vitáján.....% - ot ért el

Sopron

.....
a Bírálóbizottság elnöke

A doktori (PhD) oklevél minősítése.....

.....
Az EDT elnöke

KIVONAT

Korszerű geoinformatikai módszerek az erdőszetben

Egy geoinformációs rendszer fejlesztésének tudományos eredményei

A dolgozat egy földrajzi információs rendszer fejlesztésének tudományos eredményeit, korszerű eljárásait és technológiáit ismerteti. Az eredmények a raszteres és a vektoros adatok köré épülnek. A raszteres eljárások bemutatnak egy algoritmikus képfeldolgozót, egy új raszter formátumot, egy új interpolációs módszert, illetve javított láthatósági, hidrológiai és vektorizálási algoritmusokat. A raszteres részben több fejezet foglalkozik a digitális fotogrammetria témaköreivel, úgymint a sugárnyaláb kiegyenlítéssel, az ortofotó készítés új lehetőségeivel, a térbeli kiértékelés szoftveres változatával, valamint az automatikus felületmodell előállításával. A vektoros eljárások ismertetnek egy memória alapú adatbázis-kezelőt, egy teljesen új térbeli indexelési módszert, és egy hatékony valós idejű topológikus térképezőt. Önálló fejezet foglalkozik az erdőszeti digitális térképekkel. Az utolsó rész a háromszög alapú felületmodellezés új módszereit mutatja be.

ABSTRACT

Modern Geoinformation Methods in Forestry

Scientific Achievements of a Geoinformation System's Development

This paper describes the scientific achievements, modern algorithms and technologies of a Geoinformation System's Development. The achievements are built around the Raster and Vector data. The Raster procedures present an algorithm based Image Processor, a new raster format, a new interpolation method, and improved viewshed, hydrological and raster-to-vector procedures. In the raster part more chapters deal with Digital Photogrammetry, such as bundle block adjustment, new possibilities of Orthophoto generation, software version of spatial data acquisition, and automatic Digital Elevation Model extraction. The Vector procedures describe a main-memory based database manager, a brand-new spatial indexing method, and a real-time topological mapping tool. An independent chapter investigates the Hungarian Digital Forest Maps. The last part of this paper introduces new methods of the triangle network based surface modeling.

Tartalomjegyzék:

1. BEVEZETŐ.....	1
1.1. TÉMAVÁLASZTÁS.....	1
1.2. VISSZATEKINTÉS.....	1
1.3. NÉHÁNY GONDOLAT EREJÉIG.....	2
1.4. SEMMI SEM TÖKÉLETES.....	4
1.5. KUTATÁSI MÓDSZEREK.....	5
1.6. KÖSZÖNET.....	5
2. GEOINFORMATIKAI ÁTTEKINTŐ.....	6
2.1. ALAPFOGALMAK.....	6
2.2. MODELLEZÉS.....	7
2.3. ADATMODELLEK ÉS DIMENZIÓK.....	9
2.4. RASZTERES ADATMODELL.....	10
2.5. VEKTOROS ADATMODELL.....	12
2.6. MEGJELENÍTÉS.....	14
3. RASZTERES ELJÁRÁSOK.....	16
3.1. ALGORITMIKUS KÉPFELDOLGOZÁS.....	16
3.2. RASZTER FORMÁTUM.....	20
3.3. RASZTERES ELEMZÉS.....	24
3.3.1. <i>Piramis interpoláció.....</i>	<i>24</i>
3.3.2. <i>Láthatósági vizsgálatok.....</i>	<i>25</i>
3.3.3. <i>Hidrológiai elemzések.....</i>	<i>26</i>
3.3.4. <i>Árvíz- és gátmodellezés.....</i>	<i>30</i>
3.3.5. <i>Terjedési modell.....</i>	<i>30</i>
3.4. KÖZÉPPONTOS VETÍTÉSŰ FELVÉTELEK TÁJÉKOZÁSA.....	32
3.4.1. <i>Képi koordináta rendszer.....</i>	<i>32</i>
3.4.2. <i>A tájékozás alapegyenletei.....</i>	<i>33</i>
3.4.3. <i>Az alapegyenletek megoldása.....</i>	<i>36</i>
3.4.4. <i>Sugárnyaláb kiegyenlítés.....</i>	<i>38</i>
3.4.5. <i>Kapcsolópontok automatikus meghatározása.....</i>	<i>40</i>
3.4.6. <i>Tömbkiegyenlítés DigiTerra környezetben.....</i>	<i>42</i>
3.5. ORTOFOTÓ KÉSZÍTÉS.....	44
3.6. TÉRBELI KIÉRTÉKELÉS.....	47
3.7. AUTOMATIKUS FELÜLETMODELL ELŐÁLLÍTÁS.....	49
3.8. RASZTER-VEKTOR ÁTALAKÍTÁS.....	53
4. VEKTOROS ELJÁRÁSOK.....	55
4.1. ADATBÁZIS TERVEZÉS.....	56
4.2. RELÁCIÓS ADATBÁZIS.....	57
4.3. TÉRBELI INDEXELÉS.....	60
4.4. VALÓS IDEJŰ TOPOLOGIA.....	64
4.5. TERÜLETEK ÉPÍTÉSE.....	68
4.6. ERDÉSZETI DIGITÁLIS TÉRKÉPEK.....	71
4.7. TIN FELÜLETMODELLEZÉS.....	75
5. ÖSSZEFOGLALÁS.....	84
5.1. ELŐZMÉNYEK, CÉLKITŰZÉSEK.....	84
5.2. KUTATÁSI MÓDSZEREK.....	84
5.3. TUDOMÁNYOS EREDMÉNYEK.....	84
5.4. TÉZISEK.....	87
5.5. EREDMÉNYEK HASZNOSÍTÁSA, TOVÁBBI TEENDŐK.....	87
6. IRODALOMJEGYZÉK.....	88
6.1. NYOMTATOTT IRODALOM.....	88
6.2. ELEKTRONIKUS IRODALOM.....	89
6.3. KUTATÁSSAL KAPCSOLATOS ELŐADÁSOK, POSZTEREK.....	91

Ábrajegyzék:

1-1. Ábra: DigiTerra Map geoinformatikai program	3
2-1. Ábra: Raszteres adatmodellek tárolása.....	9
2-2. Ábra: Raszteres adatmodell felépítése.....	11
2-3. Ábra: Raszteres adatmodell tematikus kódjainak értelmezése	11
2-4. Ábra: Egy erdőrézlet objektum relációs kapcsolatainak ábrázolása	12
2-5. Ábra: Topológia - térbeli kapcsolatok egy vektoros példán.....	13
2-6. Ábra: Topológikus építkezés közben jelentkező hibák	13
3-1. Ábra: Ugyanazon terület hat részletének algoritmikus képfeldolgozása	19
3-2. Ábra: A párbeszédpanel mögött egy teljes képfeldolgozó modul húzódik meg.....	19
3-3. Ábra: Blokk és piramis technika	21
3-4. Ábra: Láthatóság vizsgálat: bal oldalon egy felülnézeti kép, jobb oldalon egy függőleges metszet a nézőpont és egy kerületi pixel között	25
3-5. Ábra: Domborzatmodell a vízösszefolyási térképpel	29
3-6. Ábra: Vízyűjtő területek és a vízösszefolyási térkép.....	29
3-7. Ábra: Perspektív leképezés	34
3-8. Ábra: Két sorból és két oszlopból álló tömb normálegyenleteinek együttható mátrixa.....	39
3-9. Ábra: Egy digitális légifelvétel és a felvételből érdeklődési operátorokkal előállított két kép	40
3-10. Ábra: Tervező nézet és a kiegyenlítés menetét mutató ablak	43
3-11. Ábra: Digitális ortofotó készítés két felvétel esetén	44
3-12. Ábra: Digitális ortofotó előtérben a digitális erdészeti térképpel	45
3-13. Ábra: Epipoláris geometria	47
3-14. Ábra: Térbeli kiértékelés a DigiTerra Map programban	48
3-15. Ábra: Automatikus felületmodell előállítás	50
3-16. Ábra: Két tájékozott felvétel átfedő részéről automatikusan előállított felületmodell.....	52
3-17. Ábra: Az automatikusan kinyert felületmodellre húzott digitális ortofotó	52
3-18. Ábra: A raszter-vektor átalakítás típusai	53
4-1. Ábra: Logikai adatmodell DigiTerra Map környezetben	57
4-2. Ábra: Bináris fa indexelés	60
4-3. Ábra: Raszter- és régiófa indexelés	62
4-4. Ábra: Két vonalszakasz metszésének alapesetei	65
4-5. Ábra: Egy vonallánc topológiája	66
4-6. Ábra: Vonalszerkesztés DigiTerra környezetben. Az ábrán piros négyzetek jelölik a végponthibákat.....	67
4-7. Ábra: Belső szigetek kikötése és a zárt terület körbejárása	68
4-8. Ábra: Vonal-terület topológia a DigiTerra Map alatt.....	70
4-9. Ábra: Digitális erdészeti térkép részlete. Bal oldalon látható a térképet felépítő fedvények listája.	74
4-10. Ábra: A négy-él és a háromszög adatstruktúra	77
4-11. Ábra: Bézier-háromszögek, baricentrikus koordináták és a rekurzív bontás.....	79
4-12. Ábra: A DigiTerra Map a háromszög topológia alapján valós időben rajzolja a szintvonalakat, a feliratokat, a háromszögeket és a Thiessen-poligonokat.....	81
4-13. Ábra: TIN alapján készített, magassági színezésű terepmodell perspektív képe.....	82
4-14. Ábra: A kitétség alapján valós időben színezett felületmodell perspektív képe.....	82
4-15. Ábra: Térbeli megjelenítési lehetőségek a DigiTerra Map programban.....	83

Táblázatjegyzék:

2-1. Táblázat: Szabályos és szabálytalan adatmodellek dimenziói	10
2-2. Táblázat: A szabályos és a szabálytalan adatmodellek összehasonlítása.....	10
2-3. Táblázat: Néhány egy vektoros állományon elvégezhető művelet	14
2-4. Táblázat: Néhány két vektoros állomány között elvégezhető művelet	14
2-5. Táblázat: A geometria megjelenítési lehetőségei	15
3-1. Táblázat: Vízrajzi elemek osztályozási táblázata	28
3-2. Táblázat: Az ismeretlenek száma a tájékozásban	39
3-3. Táblázat: DigiTerra Map tömbkiegyenlítés moduljának paneljei	42
4-1. Táblázat: PONT adattípus logikai modellje	72
4-2. Táblázat: VONAL adattípus logikai modellje	72
4-3. Táblázat: TERÜLET adattípus logikai modellje	73

1. Bevezető

1.1. Témaválasztás

Dolgozatom egy **geoinformatikai rendszer fejlesztéséről** szól. A fejlesztés nyolc éve kezdődött. A disszertációban a hangsúlyt a tudományos eredményeket jelentő **algoritmusokra, szoftver technológiákra** helyezem. Ugyanakkor figyelemmel kísérhető disszertációmban az algoritmusokat felvonultató szoftverek fejlődése, valamint a szoftverekkel elért kutatási és gyakorlati eredmények.

A geoinformatika erdészeti alkalmazása Magyarországon az 1990-es évek elejére datálható. Az Állami Erdészeti Szolgálatnál *ETIR* néven először *Arc/Info* majd *Microstation* alatt folyt a fejlesztés. A Soproni Egyetem Erdőmérnöki Karán szintén folytak geoinformatikai kutatások. Mindegyik esetben maradtak megoldatlan kérdések. Egyik rendszer sem jutott el az országos gyakorlati alkalmazás szintjére.

Az erdészeti térképezés és az erdőállományok adatbázisának hatékony kezelése viszont egyre inkább egy **geoinformatikai megoldásra várt**. Kezdetben kiegészítő szoftverek fejlesztését terveztem, amelyek mind egy önálló algoritmus köré épültek. Később az egyes szoftverek összekapcsolása után körvonalazódott egy általános geoinformatikai rendszer, mely képes kiszolgálni a magyar erdészeti szakma igényeit. A geoinformációs rendszer elkészítése, mint gyakorlati célkitűzés, egyben tudományos célkitűzés is volt. A rendszerbe új algoritmusok, a magyar erdészeti igényekhez igazított technológiák épültek be.

Az évek során sokszor nekem szegeztek a kérdést, hogy mi volt a motiváció, miért vágtam bele egy geoinformatikai rendszer fejlesztésébe. A válasz röviden a **gyakorlati igények, a megismerés és a kihívás**.

1.2. Visszatekintés

Érdeklődésem 1993-ban fordult a geoinformatika irányába. Nagyszerű lehetőségeket láttam az **informatika** és a **geotudományok** párosításában. A geotudományok esetében a korszerű adatgyűjtés, a hatékony elemzési módszerek, az informatika részéről pedig a valós környezetünkről gyűjtött nagymennyiségű adatok feldolgozása érdekelt. Először a megjelenítés fogott meg: látványos, térben ábrázolt terepmodellek, tematikus térképek, színes űrfelvételek és részletekben gazdag légifelvételek.

Később a módszertan és az **algoritmusok** felé fordultam. Egyrészt a **kihívás**, másrészt az algoritmusok **megértése** vezetett arra, hogy geoinformatikai eljárásokat programozzak. Ha lehetőségem van kipróbálni egy új szoftvert, az első dolog, hogy megpróbáljam megérteni a szoftver működését, milyen adatszerkezeteket használ, milyen algoritmusokkal dolgozik.

Az eljárásokból kisebb programok épültek fel. Első ilyen program a *DIGITERRA* névre hallgató terepmodellező szoftver volt, melyet diplomatervem részeként készítettem (*Czimer 1994*) (*A diplomaterv a III. Térinformatika a Felsőoktatásban Szimpóziumon díjat is nyert*). A szoftver különlegessége, hogy egy tetszőleges pontthalmazból rendezés után valós időben állít elő tetszőleges sűrűségű felületmodellt. A felületmodell a környező pontok súlyozásából állt elő.

1994 után doktorandusz hallgatóként igyekeztem megismerni a geoinformatika teljes spektrumát. Továbbfejlesztettem a **felületmodellező** programot, a közeli pontok súlyozása helyett szabálytalan háromszöghálót alkalmaztam. Az új program a *TIN* nevet kapta. 1995-ben elkészítettem egy képernyőn **digitalizáló** szoftvert *DTOPO* névvel, mely később a digitális erdészeti térképek előállításának eszköze lett.

1996-ban számos algoritmus beérett, és így megszületethetett a **digitális fotogrammetria** program *STEREO* néven, mely egyszerű számítógép konfiguráción is térbeli kiértékelést biztosított. Ugyanebben az évben egy nemzetközi pályázat kapcsán (*Márkus, I., Czimmer és mások 1996*) egy **képfeldolgozó** szoftver alapjait is elkészítettem *IMAGE* néven. A program a távérzékelte digitális felvételek feldolgozásához szükséges valamennyi funkciót tartalmazta. A programot számos hazai és külföldi konferencián mutattam be sikerrel (*Czimmer 1996, Czimmer 1997*). A szoftvert időközben több modullal bővítettem. Ennek köszönhető, hogy a mai napig használják egyetemek és kutatóintézetek. Ez az első magyar fejlesztésű program, mely digitális **ortofotót** állított elő.

1996-os év más tekintetben is mérföldkő. Ebben az évben két volt évfolyamtársammal megalakítottuk a *DigiTerra* társaságot. Ugyanebben az évben alakot öltött a **vektoros térképező és elemző** szoftver, melyet 1997 tavaszán nagy érdeklődés mellett mutattunk be a szakközösségnek. A programot *MAP* névre kereszteltük.

1997-től elindult az 1:10'000 méretarányú **erdészeti üzemi térképek** digitalizálása. Jelenleg az országos lefedettség 60%-os. 1999-re elkészült a Digitális Erdészeti Térképek formátum leírása is *DET* rövidítéssel (*Állami Erdészeti Szolgálat 1999*), melynek kidolgozásában én is közreműködtem.

Időközben a *Windows* térhódítása nyilvánvalóvá vált, ezért az eddig kifejlesztett szoftvereket egy közös grafikus környezetbe kezdtük átültetni. A program felépítésének köszönhetően egy integrált, általánosan használható, hatékony térinformatikai eszköz született. Az évek során a program újabb modulokkal egészült ki. Részben a geoinformatika, részben az erdőgazdálkodás területéről épültek be újabb elemek. Az erdészeti szakmai funkciók (*erdőművelés, fahasználat, csemetekert, vadászat*) megjelenését az indokolta, hogy az adatfeldolgozás tárgya ugyanaz az objektum, nevezetesen az erdőrészlet, mint a geoinformatikában. Az új funkciókat évente megrendezett fórumokon, konferenciákon ismertettük.

A programot jelenleg az Állami Erdészeti Szolgálat központja és irodái, kisebb-nagyobb geodéziai cégek (*FÖMI, Geodézia Rt, megyei földhivatalok*), tizenöt Erdészeti Részvénytársaság, egyetemek és kutatóintézetek használják. A telepített programok száma meghaladja a 250-et. A Soproni Egyetem oktatói és hallgatói ingyen használhatják a szoftvert.

2002-ben már a **DigiTerra MAP** névre keresztelt program harmadik verzióját fejlesztjük. Ez a verzió a geoinformatika új irányzatainak jegyében születik: általános adatbázis elérés, programozhatóság, Internet kapcsolat, többnyelvű és alakítható grafikus környezet ...

1.3. Néhány gondolat erejéig

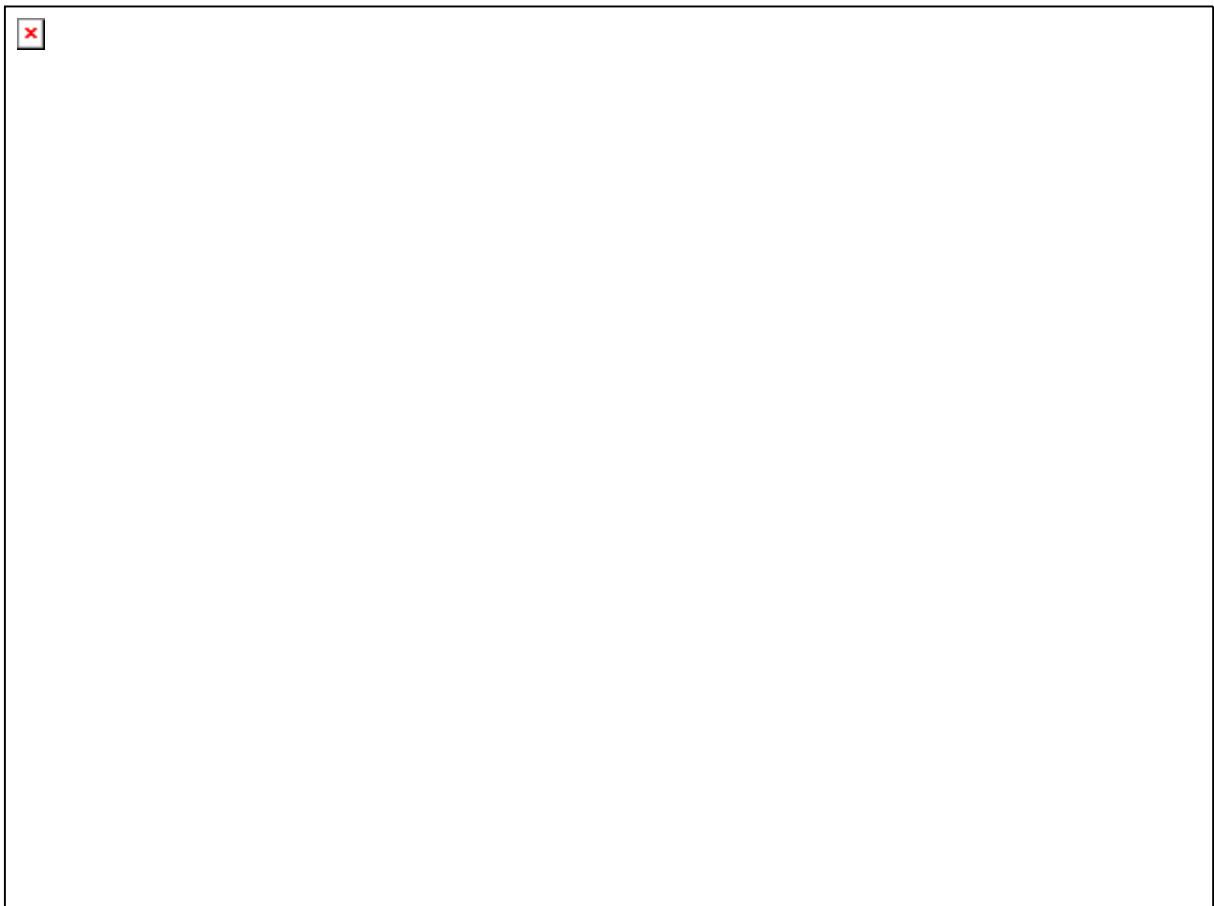
Van még egy dolog, ami motivált a szoftverfejlesztésben. Ez pedig a geoinformatikai programok összetettsége és terjedelme. Megfigyelhető, hogy a *Windows* világban a programok méretének növekedése lépést tart a háttértárolók kapacitásának fejlődésével. A szoftverek

funkcionalitása és hatékonysága viszont elmarad ettől a tendenciától. Az is elgondolkodtató, hogy az újabb verziók egy korszerűbb számítógépen sokszor a korábbi verzióval azonos sebességgel futnak. Ez elsősorban a megnövekedett memóriaigénynek, a robusztus és redundáns kódnak tulajdonítható.

A számítógépes berkekben ismert *Moore* törvénye, mely kimondja, hogy adott áron elérhető számítógépes teljesítmény néhány év alatt megkétszereződik. Ismert viszont *Grosch* választörvénye is, mely szerint hiába fejlődik a hardver, a szoftverek úgyis minden erőforrást (vagy még többet) kihasználnak (*Platt 2001*).

A szoftvercégek a **robusztus** jelzővel illetik programjaikat. Miért ez a jelző: talán, mert egy robusztus programot nehezebb ledönteni a lábáról. Sajnos a fizikai világból vett törvényszerűség nem állja meg a helyét a szoftverek terén. Egy szoftver azért robusztus, mert több programkódot tartalmaz, amely statisztikailag több hibával terhelt. A válasz valószínűleg egyszerűbb és nem a robusztusság és a bővülő funkciók a magyarázat, hanem a hardver- és szoftvergyártó cégek közötti piaci összefüggések és érdekek.

Szerény véleményem, hogy egy program tudjon kevesebb, de világos és sokoldalúan paraméterezhető funkciót. A funkciók megvalósítása mögött egy tömör és hatékony kód húzódjon meg. A nagyobb programok objektumokból, komponensekből, modulokból épüljenek fel. A *DigiTerra Map* szoftver (1-1. Ábra) funkcióit tekintve felveszi a versenyt a nemzetközi geoinformatikai szoftverekkel, ellenben a programkód mérete mindössze 2 MB.



1-1. Ábra: *DigiTerra Map* geoinformatikai program

A *DigiTerra Map* program a következő modulokat tartalmazza: *adatbázis kezelő, jelentéskészítő, topológikus térképező, raszteres elemző, képfeldolgozó, digitális fotogrammetria, felületmodellező, térbeli megjelenítő és erdészeti szakmai modulok*. A nagyfokú **integráltság** szembeötlő. Más geoinformatikai szoftverek az előbbi funkciókból mindössze néhányat tartalmaznak. A *DigiTerra Map* fejlesztésekor bizonyítani akartam, hogy egyetlen szoftver képes a geoinformatikai feladatok széles spektrumát megoldani.

A program mellett szól még a magyar nyelvű környezet is. Nagyon kevés geoinformatikai rendszer rendelkezik magyar nyelvű felhasználói felülettel és dokumentációval. Még kevesebb azon programok száma, amely támogatja a speciális magyar igényeket: *kartográfiai igények, adatcsere formátum, jogszabályok, hazai vetületek, kódszótárak*.

A fejlesztés során sok esetben hoztam létre olyan szoftvertechnológiákat, amelyek röviddel a létrehozás előtt vagy után párhuzamosan más geoinformatikai szoftverben is megjelentek. Ez arra utal, hogy sikerült felzárkóznunk a nemzetközi szoftverfejlesztés **színvonalához**. Természetesen most már két tárprogramozóval dolgozom a *DigiTerra* cégnél. A szoftvertervezést és az alapkód kidolgozását viszont még mindig én végzem.

A geoinformatikai rendszerek jelenlegi fejlődését az adatbázis-kezelő rendszerek és a számítógépes hálózatok, az Internet fejlődése határozza meg. A fejlődésben és az új geoinformatikai szabványok kidolgozásában jelentős szerepet vállal az **OpenGIS** konzorcium (*OpenGIS 1998, OpenGIS 2001*). A geoinformatikai világ egyre inkább visszakanyarodik az **adatbázis-kezelés** világa felé. Korábban a szoftverek speciális adatfájlokban tárolták a földrajzi adatokat. A jelenlegi adatbázis-kezelők viszont lehetővé teszik a vektoros és a raszteres adatok tárolását az adatbázisban és biztosítanak bizonyos szintű térbeli lekérdezéseket is (*Sheklar és mások 1998*). Az adatbázis-kezelő rendszerek (*Oracle, Microsoft SQL, IBM DB2, Informix*) elősegítik szabványok kialakulását és terjedését, az adatok kompromisszumok nélküli átadását és központosítását. A központosítás csökkenti a redundanciát és elősegíti az ügyfél-kiszolgáló (*kliens-szerver*) rendszerek kiépítését.

A **jövő** geoinformatikai rendszere egy kliens böngésző-elemző program lesz, amelybe az Interneten keresztül a szerverről töltődnek le a komponensek (*Java, ActiveX, .Net*) és az adatok. Az adattárolás, adatelőhívás, bizonyos esetekben az elemzés funkcióját is átveszik a szerveren futó adatkezelő rendszerek. Ez a fejlődési irány kedvez a kis fejlesztő cégeknek, akik egyszerűen kezelhető, legkevésbé robusztus, komponens alapú szoftvereket készítenek.

1.4. Semmi sem tökéletes

Természetes, hogy egy három fős fejlesztőcsapat nem tud versenyezni egy háromszáz fős fejlesztő gárdával. Jelenleg a *DigiTerra* programok mögött mintegy 140'000 programsor áll, amely több mint **kétezer A4-es oldalnak** felel meg. A számok ellenére a szoftverünknek mégis vannak hiányosságai. Néhányat a teljesség igénye nélkül megemlítenék. A jelenlegi verzió nem tartalmazza, de az új verzióban már tervezzük a nemzetközi alapfelületek és vetületek támogatását. A programozhatóságra és alakíthatóságra az előbb elmondottak érvényesek.

A programunk a szükséges fájlformátumokat támogatja, de korántsem annyit, mint egy nemzetközi térinformatikai szoftver. Úgy tűnik időközben ez a kérdés is megoldódik. A jövő geoinformatikai rendszerei mind egy adatbázisból fognak dolgozni. A hozzáférés szabványos felületeken keresztül történik majd.

Néhány kritikát is megemlítek, amely a cégünk felé irányul. Sokszor elhangzott, hogy „*túl kicsik vagyunk*”. Cégünk immár 6 éve prosperál. A felhasználói igényekre sokkal gyorsabban, rugalmasabban tudunk válaszolni, mint a nagy cégek. A céget az erdészeti szakma igényei hívták életre, és amíg a szakma mellettünk áll, addig ki is szolgáljuk.

A programstabilitás is egy fontos kérdés, főleg *Windows* környezetben. Jelenlegi verziónk eljutott az integráltság olyan szintjére, amely már a stabilitás rovására megy. Az új verziót már a felmerült problémák ismeretében fejlesztjük. Az új fejlesztői környezetek (*Visual Basic, JavaScript, Java, C++, C#, .NET*) már leveszik a programozó válláról a programok legkritikusabb részét a memória kezelést, az objektumok létrehozását, bővítését és törlését. A fejlesztői környezetek közül talán a legígéretesebb a *Microsoft .NET* környezet (*Platt 2001*), amely forráskód és platform független megoldást kínál. A környezet egy új operációs rendszerként is felfogható, ami biztonságosan futtatja az Internetről letölthető és komponensekből felépülő programokat. A *.NET* környezet *Windows 2000* és *XP* operációs rendszerek alatt már elérhető.

1.5. Kutatási módszerek

A kutatási téma meghatározása a geoinformatika adott területének kiválasztását jelentette. Gyakori volt az igények szerinti témaválasztás is. Ezután következett a nyomtatott és az elektronikus szakirodalom tanulmányozása. Utóbbihoz nagy segítséget nyújtott a *CiteSeer - Scientific Literature* (<http://citeseer.nj.nec.com/cs>) digitális könyvtára. A szakirodalom alapos vizsgálatát az algoritmus tervezése, programozása és tesztelése követte. A tesztelés eredménye sokszor a program, néha viszont az algoritmusterv javítását kívánta meg. Az utolsó fázis az algoritmusok összedolgozása, az algoritmusokból eljárások, az eljárásokból programmodulok és végül a programok felépítése. A programmodulokat már konkrét gyakorlati alkalmazásokon tudtam tesztelni.

1.6. Köszönet

Aki valaha is programozott tudja, hogy a szoftverfejlesztés nagyon sok időt igényel: programtervezés, kódolás, véget nem érő tesztelés és javítás. Nagyon nehéz egy olyan programot írni, amelyet nem csak a programozó, hanem mások is használni tudnak. Mégis a programozás a rendszerfejlesztésnek csak az egyik fele. Az új technológiák, algoritmusok kidolgozása csaknem ugyanannyi időt igényelt. Ezért a köszönet mindenkit megillet, aki a rendszer megszületéséhez közvetve-közvetlenül hozzájárult.

Hálával tartozom kis családomnak, a *DigiTerra Mérnöki Irodának*, és a *Soproni Egyetem* oktatóinak és hallgatóinak, kiemelve a *Földmérési és Távérzékelési Tanszék* munkatársait. A program első számú tesztelője mai napig is a Tanszék. A szakmai és gyakorlati eredmények, az évek során összegyűlt értékes tapasztalatok mind a *DigiTerra Mérnöki Iroda* munkáját és kitartását dicsérik.

Köszönetet mondok az erdész, a földmérő és a távérzékelési szakterület művelőinek, mindazoknak, akik az erdészeti geoinformatika és a *DigiTerra* szoftverek elindulását, fejlődését és országos kiteljesedését támogatták. Külön köszönet illeti dr. Bácsatyai Lászlót, aki a pályán elindított, patronált és doktori témavezetőm volt.

2. Geoinformatikai áttekintő

A disszertáció fő tárgyát képező algoritmusok és technológiák tárgyalása előtt következzenek a geoinformatika rövid ismertetése. Ennek célja elsősorban a szakterület és az általam használt fogalmak tisztázása.

2.1. Alapfogalmak

A geoinformatika alapfogalmait az **informatika** alapfogalmaiból célszerű levezetni. A következő felsorolásban a fogalmakat egymásra épülésük sorrendjében tárgyalom:

- egyed** a valós világ véges számú tulajdonságértékekkel leírható tárgya, személye, eseménye, jelensége, amely az információs rendszer szempontjából lényeggel bír. Szinonima: *entitás*.
- egyedtípus** egyedek azonos tulajdonságokkal jellemezhető osztályai. Szinonima: *entitástípus*.
- objektum**..... az egyed számítógépes környezetben történő konkrét ábrázolása.
- objektumtípus** objektumok azonos tulajdonságokkal jellemezhető osztályai.
- adat**..... értelmezhető és feldolgozható objektív tény, ismeret, fogalom, mely egy objektum adott tulajdonságának leírására szolgál.
- adatmodell**..... egy információs rendszerben szereplő objektumtípusok felsorolása, az objektumtípusok tulajdonságainak és kapcsolatainak leírása.
- adatmező**..... objektumtípusok egy tulajdonságát tároló adattárolási egység.
- adatrekord**..... egy objektum véges számú tulajdonságait tároló adattárolási egység.
- adattábla** azonos típusú objektumokat tároló adattárolási egység.
- adatbázis**..... véges számú objektum előfordulások adatmodell szerint szervezett együttese.
- információ**..... az adat értelmezésével előállított új, vagy újszerű ismeret, jelentés, híranyag, tájékoztatás.
- informatika**..... információ kezelésének elméleti és gyakorlati kérdéseivel foglalkozó tudományág.
- információs rendszer** ... az információ kezelésére hivatott rendszer. A rendszer funkciói az adatok gyűjtése, tárolása, megjelenítése, rendszerezése, elemzése.

A **geoinformatikai** fogalmak származtatása a fenti fogalmak felhasználásával a **geo-** vagy **földrajzi** előtaggal történik. Az előtag az ismeretszerzés tárgyát **földrajzi környezetünkre** szűkíti. Földrajzi környezetünk a Föld felszíne, a felszínen vagy alatta lévő tereptárgyak, jelenségek összessége, amely az ismeretszerzés szempontjából lényeggel bír. Földrajzi

környezetünk képzése tehát egy lényegkiemelő, egyszerűsítő absztrakciós folyamat eredménye.

Ezek után a **földrajzi objektum** a Föld felszínén vagy alatta található tárgyak, jelenségek számítógépes ábrázolását jelenti. A **földrajzi információ** a Föld felszínéről, a felszínen vagy alatta található földrajzi objektumokról gyűjtött ismeretanyag. A **geoinformatika** pedig a földrajzi információ kezelésének tudománya.

Magyarországon rövidege miatt elterjedt a **tér-** előtaggal történő fogalomképzés: **téradat**, **térinformatika**. Én ezt nem tartom szerencsésnek, mert sok esetben a képzett szavak félreérthetők és a tér- előtag nem minden esetben utal földrajzi környezetünkre (*CAD/CAM, csillagászat stb. is térbeli adatokkal dolgozik*). Rövidege és lényegre utalása miatt a **geo-**előtag használatát szorgalmazom.

2.2. Modellezés

A geoinformatika lényege, hogy **modelleket** alkossunk környezetünkről, amelyek alkalmasak a nyilvántartás, az elemzés, a szimuláció és a döntéstámogatás bonyolult problémáinak megoldására. A modellezés tömören a **valós világ** csökkentett információ készlettel történő leírása (*Horváth 1989*).

A valós világ leírása egy háromlépcsős absztrakciós folyamat eredménye (*Detrekői, Szabó 1995*). Az első lépésben a valós világot egy **elméleti modellel** helyettesítjük, amelyben meghatározzuk a valós világ egyedeit, személyeit, tárgyait, jelenségeit, eseményeit, amelyeket a végső modellben szerepeltetni kívánunk. Második lépésben meghatározzuk az elméleti modell egyedeinek leírásához szükséges jellemzőket és a közöttük lévő összefüggéseket, azaz létrehozuk az objektumok **logikai modelljét**. A harmadik, befejező lépésben előállítjuk a **fizikai modellt**, mely a logikai modell számítógépes környezetben történő leképezését és feltöltését jelenti. A modell hatékonysága a modell eredetiségével és egyszerűségével mérhető.

A számítógépes modellben szereplő objektumok jellemzésére szolgáló **tulajdonságok**:

- 1. **helyzeti adatok** - geometria
- 2. **leíró adatok** - attribútumok
 - 2.a. azonosító adatok (*sorszám, helység*)
 - 2.b. csoportosító adatok (*osztály - csoport - típus*)
 - 2.c. kapcsoló adatok (*reláció és térbeli kapcsolat*)
 - 2.d. szakadatok (*szakterület*)
 - 2.e. metaadatok (*adat az adatról*)

Egy földrajzi információs modell létrehozása során az elméleti modellben felsorolt földrajzi **egyedeket**, vagy **entitásokat**, **geometriai elemekkel és attribútumokkal írjuk le**. A földrajzi egyedek számítógépes környezetben történő ábrázolásai a **földrajzi objektumok**.

Földrajzi objektumok geometriai jellemzésére általában a következő elemeket használjuk: **pont, vonallánc, háromszög, négyszög, sokszög**. Összetett objektumok leírása az előbbi elemek többszörözésével történik. Ívek, görbe elemek közelítése a töréspontok számának növelésével, interpolációval érhető el. Felületek ábrázolása háromszögek vagy négyszögek összekapcsolásával lehetséges. Belső szigetek, gyűrűk kialakítása sokszögek egymásba

ágyazásával oldható meg. A geometriai elemek jellemző pontjait koordinátákkal adjuk meg. A többnyire két-, ritkábban háromdimenziós koordinátákat valamilyen előre megválasztott vonatkozási rendszerben értelmezzük.

Az **attribútumok** a földrajzi objektumok leírásához használt szöveges és numerikus tulajdonságértékek összessége. Az attribútumok típusai az azonosító- és kapcsoló adatok, a szakadatok és minőségi adatok.

Az azonosító adatok az objektum **egyedi azonosítására** szolgálnak. Ez a legtöbb esetben egyetlen adatmezőt jelent, amelyben egy növekvő vagy előre kialakított sorszám kap helyet (*lineáris sor, irányítószám*). Az azonosító adatmezőt szokás **elsődleges kulcsnak** is nevezni. Az elsődleges kulcsra hivatkozó adatmezőket **kapcsoló kulcsoknak** hívjuk. Előfordul, hogy az azonosításra több adatmező értékeinek összessége szolgál (*hely – tag – részlet – alrészlet, község – fekvés – helyrajzi szám – alátörés – épületjel – lakásszám*). Ilyenkor kompozit kulcsról beszélünk. Osztott adatbázisoknál gyakori, hogy az azonosító adatok között megjelenik az adat létrehozásának helyére utaló kód is. Ezzel garantálható az azonosító kulcs egyedisége, és megoldható az adatok későbbi összesítése, feldolgozása. Régebben gyakori volt digitális térképeken a földrajzi objektumok **geokódolása**. A geokód egy földrajzi pozícióval kiegészített azonosító, mely az objektum geometriája közelében vagy belsejében helyezkedik el, és utal az objektum leíró adataira.

A leíró adatok következő kategóriája az objektumok **csoportosításáról** gondoskodik. Helytakarékoság és az egységes kezelés miatt sokszor az eltérő objektumtípusokat egy adattáblában tároljuk. Ilyenkor szükségessé válik az objektum típusára utaló adatmezők bevezetése. Nagyszámú objektumtípus összevonásakor további hierarchikus csoportok kialakítása is indokolt lehet. Egy lehetséges hierarchikus bontás a következő: *osztály* → *csoport* → *típus*. Egy osztály több csoportot, míg egy csoport több típust foglal magába.

Az attribútumok harmadik csoportja az objektumok **kapcsolatait** írja le. Az objektumok között általában relációs és térbeli kapcsolatokat definiálunk. A definíció az adatmodell része. **Relációs kapcsolat** az objektumok tulajdonságértékeinek összehasonlításán alapul. A fontosabb relációs típusok a következők: *egyenlő, eltérő, kisebb, kisebb-egyenlő, nagyobb, nagyobb-egyenlő, hasonló*. A **térbeli kapcsolat** az objektumok geometriáján alapul. A térbeli kapcsolat lehetséges esetei: *tartalmazás, átfedés, határosság, metszés, közelség, elkülönülés*. Bizonyos adatmodellek esetében a térbeli kapcsolatok magától értetődőek, sokszor viszont a térbeli kapcsolatok feltárása és digitális tárolása az egyik legnehezebb feladat. A relációs és térbeli kapcsolatok kialakíthatók azonos és eltérő típusú objektumok között is.

Az attribútumok között a következő csoport a **szakadatoké**. A leíró adatok e csoportját mindig a geoinformatikai rendszer létrehozásának célja határozza meg. Az adatmezők a szakterület adatait tárolják, annak a szakterületnek az adatmezőit, amely a geoinformatikai rendszert használja.

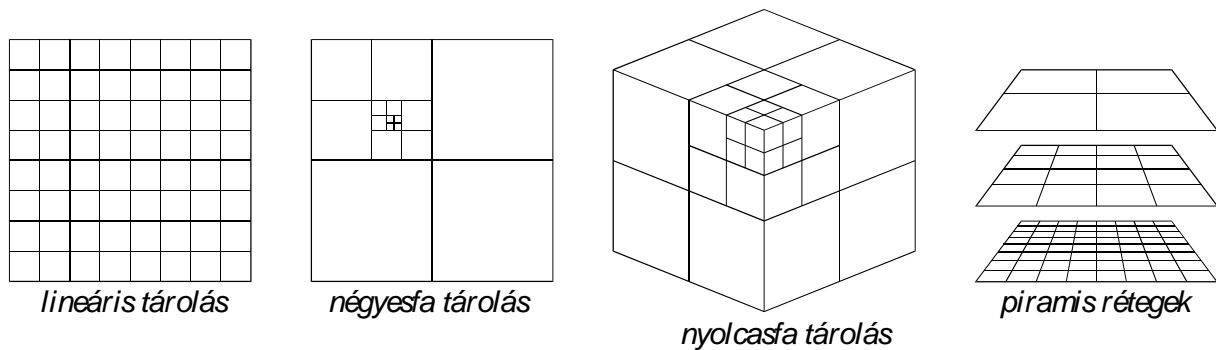
A **metaadatok** a leíró adatok utolsó csoportját képezik. A metaadatok kapcsolódhatnak közvetlenül az objektumokhoz, de az ismétlések elkerülése miatt sokszor nagyobb adattárolási egységnél rögzítik. A metaadatok az adatmodellt, az adatbázist, a táblákat, a mezőket vagy a rekordokat jellemezhetik. A jellemzés kiterjed az adatbázis céljára, az adatforrásokra, a források vonatkozási rendszereire, az adatok keletkezésére és aktualitására, a geometriai és tematikus pontosságukra. Nem elhanyagolhatók a szerzői jogokra és a korlátozásokra vonatkozó leírások sem.

2.3. Adatmodellek és dimenziók

Egy földrajzi információs rendszerben szereplő objektumok elsődleges jellemzésére a geometria szolgál. A geometria speciális kezelése meghatározza az alkalmazott **adatmodell** típusát. Az adatmodellek egy lehetséges csoportosítását adja a modellben résztvevő geometriai elemek **szabályos**, illetve **szabálytalan** elhelyezkedése és kiterjedése (Czimer 1997).

Szabályos geometriai elemek

Ezek az adatmodellek egyértelműen leírják a szabályos geometriai elemek térbeli elhelyezkedését, egymáshoz való viszonyát és az elemi alkotók formáját, méreteit. Az ilyen adatmodelleket **tesszelációs modelleknek** is szokás nevezni, amelyek a teret szabályos geometriai elemekre bontják. Dimenzióit tekintve általában mindig két- vagy háromdimenziósak. Leggyakoribb változata a **raszteres adatmodell**, amely a területet elemi téglalapokra (*pixel*), míg a teret elemi téglatestekre (*voxel*) bontja. A raszteres adatmodellben az elemi alkotók általában lineárisan helyezkednek el. Gyakori azonban a rekurzív (*négyesfa*, *nyolcasfa*) és hierarchikus (*piramisrétegek*) kialakítás is (2-1. Ábra).



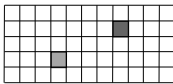

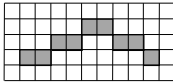
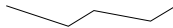
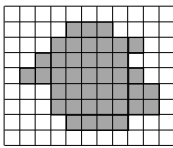
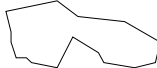
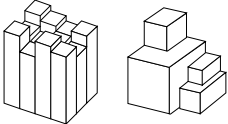
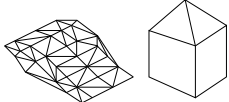
2-1. Ábra: Raszteres adatmodellek tárolása

Előfordulnak szabályos háromszög és hatszög felbontások is, de ezek grafikus megjelenítése nem igazodik a korszerű raszteres megjelenítőkhöz. A tesszelációs adatmodelleknél a térbeli kapcsolatok egyértelműek, ezeket nem kell külön definiálni. Fontos megemlíteni, hogy a raszteres adatmodellek alkalmasak a szabálytalan geometriai elemek csoportosítására, **térbeli indexelésére** is.

Szabálytalan geometriai elemek

Az adatmodellek másik nagy csoportját azok a modellek alkotják, ahol a geometriai elemek jellemző pontjai szabálytalanul helyezkednek el. A pontok, vonalláncok, háromszögek, négyszögek, sokszögek jellemző pontjainak ábrázolása koordinátákkal, vektorokkal történik. Innen származik az adatmodell elnevezése is: **vektoros adatmodell**. A vektoros adatmodellnél nagy hangsúlyt kell fektetni a térbeli kapcsolatok megfelelő kialakítására.

Mind a szabályos, mind a szabálytalan adatmodellek között számos altípust lehet elkülöníteni. A csoportosítás történhet a **geometriai elem kiterjedésének dimenziója** szerinti szerint (2-1. Táblázat).

A kiterjedés dimenziója	Szabályos adatmodell < raszter >	Szabálytalan adatmodell < vektor >
0D - nincs (<i>pixel, pont</i>)		
1D - lineáris (<i>vonallánc</i>)		
2D - sík (<i>négyszög, sokszög</i>)		
3D - térbeli (<i>felület, test</i>)		

2-1. Táblázat: Szabályos és szabálytalan adatmodellek dimenziói

Mindegyik adatmodell-változat rendelkezhet egy további dimenzióval is, az **idővel**. Ezek az adatmodellek alkalmasak az időbeli változások rögzítésére, különböző időponthoz tartozó állapotok együttes elemzésére, korábbi változatok visszaállítására. A megfelelő adatmodell kiválasztása, ismerve az adatmodellek előnyeit és hátrányait (2-2. Táblázat), mindig hosszas mérlegelés eredménye. Egy korszerű geoinformációs rendszer támogatja mindkét adatmodellt és az adatmodellek közötti kétirányú konverziót is.

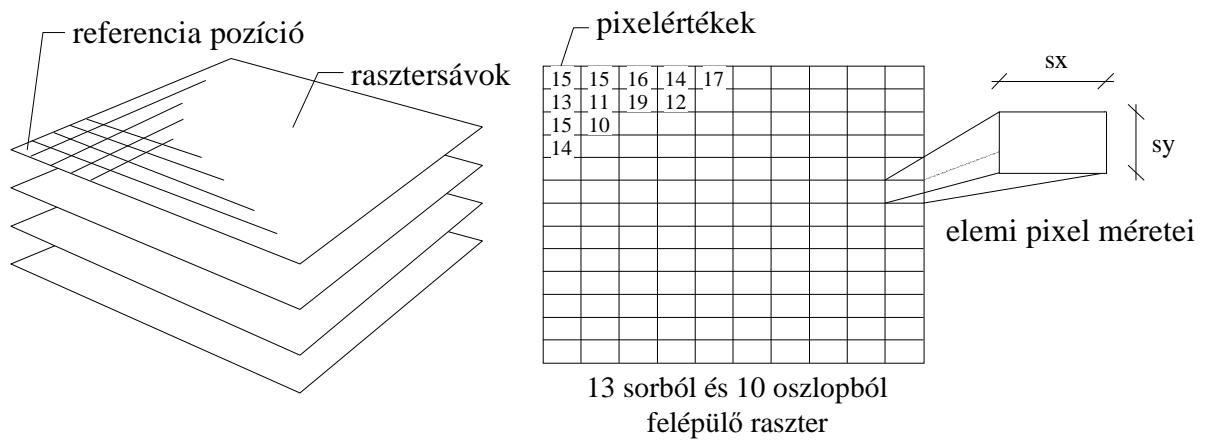
Karakterisztika	Szabályos adatmodellek < raszter >	Szabálytalan adatmodellek < vektor >
Előállítás	Többnyire egyszerű	Munkaigényes
Geometriai pontosság	Kevésbé pontos	Pontos
Tárolás típusa	Mátrix	Szekvenciális
Tároláshoz szükséges hely	Nagy	Kicsi
Kereső algoritmusok	Gyors	Lassú
Térbeli kapcsolatok	Egyszerű	Bonyolult
Térbeli elemzések	Egyszerű	Bonyolult
Térbeli mintavételezés	Jó	Változó
Információ visszaadás	Részletes és egyenletes	Lényegi és egyenlőtlen
Elévülési idő	Rövid	Hosszabb
Aktualizálás	Egyszerű	Bonyolult

2-2. Táblázat: A szabályos és a szabálytalan adatmodellek összehasonlítása

2.4. Raszteres adatmodell

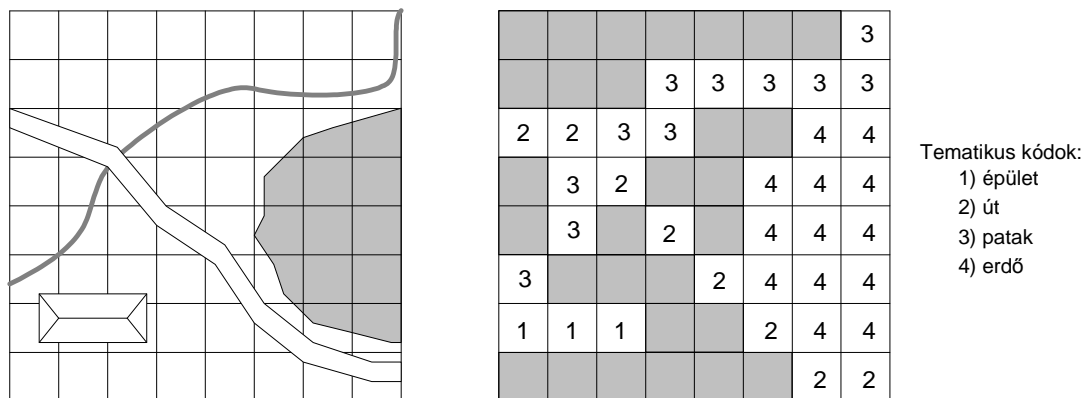
A raszteres adatmodell elemi téglalapokból, **pixelekből** épül fel. Az elemi pixelek sorokba és oszlopokba rendeződnek. Az így kialakult mátrixok alkotják a **raszter sávjaikat**. A raszteres adatállomány általában több rasztersávot tartalmaz.

A raszter geometriáját az elemi téglalapok szélessége és magassága, a sorok és oszlopok száma, valamint a raszter egy kiválasztott (általában bal-felső) pontjának földrajzi koordinátái szolgáltatják (2-2. Ábra). A méret és pozícióadatok a raszter **georeferencia** adatai.



2-2. Ábra: Raszteres adatmodell felépítése

A raszter leíró adatait az egyes pixelekben tárolt numerikus értékek alkotják. Az értékek a raszter típusától függően sokfélék lehetnek: *képinzítás értékek, magasságok, kitettség adatok, hőmérséklet adatok, csapadékmennyiségek, terepi mérések eredményei, tematikus kódok stb.* Utóbbi kód a pixel által lefedett terület tartalmi, minőségi jellemzésére szolgál (2-3. Ábra). Raszteres adatmodell esetében speciális szerephez jut az **üres cella**, mely az adott raszterrel le nem fedett területet jelöli.



2-3. Ábra: Raszteres adatmodell tematikus kódjainak értelmezése

Raszteres adatok előállítását korszerű és hatékony eszközök végzik: *digitális kamerák, távérzékelő műholdas felvevők, lapolvasók.* Az adatok származhatnak más adatforrásokból is. A vektor-raszter átalakítás, a raszteres térbeli interpoláció és térbeli elemzések eredményei mind raszteres állományok.

A raszteres elemzések hatékonysága a **végeselem módszerben** rejlik. A raszter a vizsgált területet véges számú elemi pixelre bontja. A pixelek esetében az elemző funkciók könnyen megkomponálhatók. A raszter felépítésének köszönhetően a rasztercella elérése, a térbeli kapcsolatok, a szomszédsági viszonyok egyértelműek. A raszteres elemző funkciókat működési hatókörük alapján különítjük el:

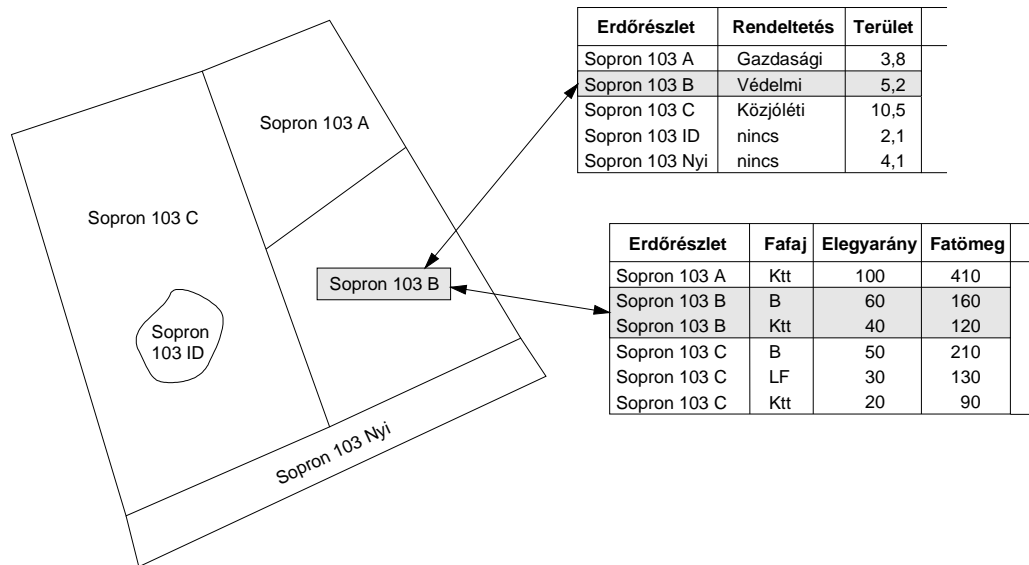
- **lokális** funkciók – hatóköre a pixel maga (*sávok közötti matematikai műveletek ...*),
- **fokális** funkciók – hatóköre a pixel közvetlen környezete (*élesítés, simítás, lejtés ...*),
- **zonális** funkciók – hatóköre az azonos értékű pixelek összessége (*statisztika ...*),
- **globális** funkciók – hatóköre a raszter egésze (*statisztika, láthatóság, terjedés ...*).

Az elemző funkciók jelentős része raszteres terepmodellel értelmezhető, ahol az egyes pixelértékek a terepfelszín magasságát írják le.

2.5. Vektoros adatmodell

A vektoros adatmodellt felépítő adatmezők két nagy csoportja a **geometria és az attribútumok**. A geometria és az attribútumok tárolódhatnak külön adattáblákban vagy együttesen. A geometriai elemeket szabálytalan elhelyezkedésű, koordinátákkal, vektorral adott pontok építik fel.

A földrajzi elemeket objektumtípus szerint általában külön adattáblákban tároljuk. Nagyszámú objektumtípus esetén az eltérő típusú objektumok összevonhatók. Ilyenkor az objektum típusát külön adatmező tárolja. Gyakori, hogy az objektumhoz az azonosító adatok alapján más objektumok is kapcsolódnak. Egy erdőrészlet geometriájához az azonosító adatokon kívül kapcsolhatók a részletadatok, az erdősítési adatok, a gazdálkodási adatok, valamint több fafajsort és megjegyzés adatai is (2-4. Ábra). Az előbbi típusú adatok mind más-más adattáblában helyezkednek el. Az objektumok előhívása relációs kapcsolatok segítségével történik.



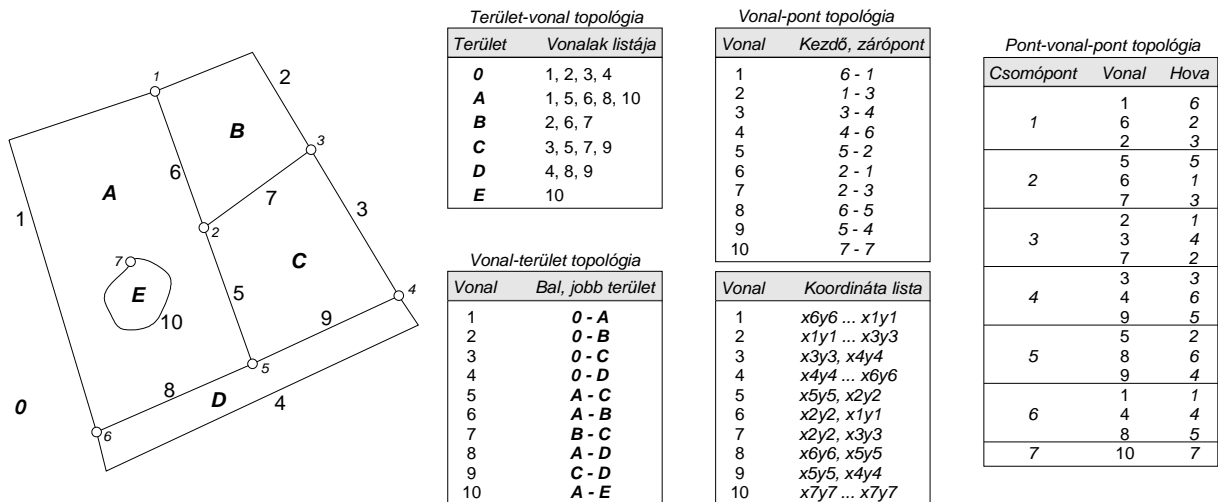
2-4. Ábra: Egy erdőrészlet objektum relációs kapcsolatainak ábrázolása

Vektoros adatmodell esetében az attribútum adatok között nagy szerephez jutnak a térbeli kapcsolatokat definiáló adatmezők. A térbeli kapcsolatokat **topológiának** nevezzük. A topológia egyrészt elősegíti a térbeli elemzéseket, másrészt biztosítja az ellentmondás mentes vektoros adatállományok létrehozását.

A topológikus adatszerzés elve a következő. Az adatábrázolás alapegysége a koordinátaival adott pont. A pontokat vonalláncok kötik össze. A vonalak egymást nem metszik, csak a

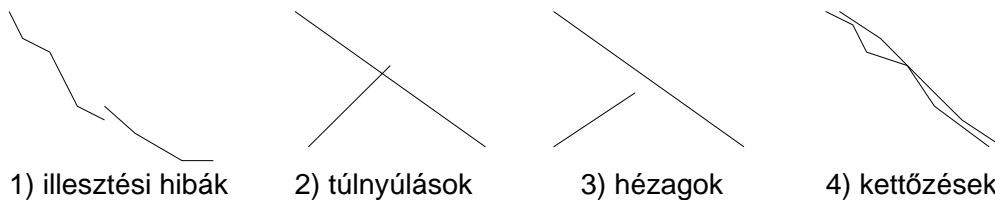
csomópontokban találkoznak. A vonalláncok által körbezárt térrészek területeket, sokszögeket alkotnak.

Az előbbi építkezést követve létrejön a pont-vonal, vonal-terület topológia. A kapcsolatok mindkét oldalon tárolódnak: a pont mely vonalon található, a vonalakat mely pontok építik fel, a vonal mely területeket választja ketté, a területet mely vonalak építik fel. A kapcsolatok az azonos típusú elemek között is létrejönnek: vonal-vonal kapcsolat, terület-terület szomszédosság. Összetett kapcsolatok is felállíthatók: pont-vonal-pont topológia (2-5. Ábra).



2-5. Ábra: Topológia - térbeli kapcsolatok egy vektoros példán

Topológikus vektoros adatállományok építése közben számos hiba léphet fel. Az egyszerű hibák gyakran csak később derülnek ki: például a hézagok területek összekapcsolását eredményezik, amely csak a területépítés után derül ki. Fontos tehát a topológiai hibák mielőbbi, akár szerkesztés közbeni felderítése (2-6. Ábra).



2-6. Ábra: Topológikus építkezés közben jelentkező hibák

A vektoros adatmodellek elemzése általában sok időt és számítást igényel. A számítások a megfelelő topológia kialakításával és a térbeli indexeléssel gyorsíthatók. A térbeli indexelés a geometriai elemek gyors előhívását teszi lehetővé. A műveleteket szokás csoportosítani a bemeneti adatállományok száma szerint.

Egy vektoros állományon elvégezhető térbeli műveletek sokfélék lehetnek (2-3. Táblázat). Számos művelet a leíró adatok alapján dolgozik (*kiemelés, összevonás*), néhány művelet a geometriát módosítja (*javítás, generalizálás, merőlegesítés, összekapcsolás, transzformálás*), további elemzések új geometriai alakzatokat állítanak elő (*védőzóna, súlypont, befoglaló téglalap, konvex burkoló, legrövidebb út keresés*). Két vektoros állomány között értelmezhető műveletek leggyakrabban halmazműveletekre vezethetők vissza (*unió, különbség, metszet*).

Gyakori, hogy ezek a műveletek töréspontokat módosítanak, új metszéspontokat hoznak létre (2-4. Táblázat).

Művelet	Forrás	Eredmény
Kiemelés az attribútumok alapján a geometriai elemek kiemelése		
Eliminálás az attribútumok alapján a felesleges határolóvonalak törlése		
Generalizálás a töréspontok egy megadott távolságra lehetnek egymástól		
Pufferzóna a geometriai elemek körüli fix vagy változó távolságra lévő zóna		
Transzformálás a geometriai elemek transzformálása egy másik koord.rendszerbe		


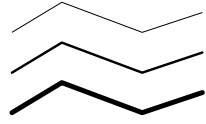
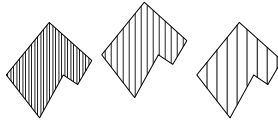

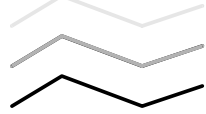
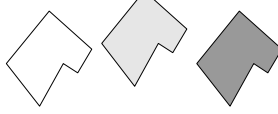

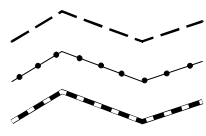
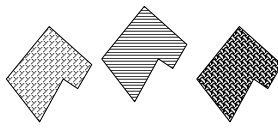
2-3. Táblázat: Néhány egy vektoros állományon elvégezhető művelet

Művelet	Források	Eredmény
Egyesítés - unió a geometriai elemek átfedésével új geometriai elemek képzése		
Kivonás terület elemek kivonásával új területek képzése		
Metszet képzés területelemek közös részéből új területek képzése		
Aktualizálás a geometriai elemek lecserélése az újakra		
Térkép illesztés különböző forrásból származó geometriai elemek illesztése		

2-4. Táblázat: Néhány két vektoros állomány között elvégezhető művelet

2.6. Megjelenítés

A földrajzi objektumok leíró adatainak megjelenítése táblázatos vagy listás formában lehetséges. Az adattáblák geometriai adatait rendszerint grafikusán ábrázoljuk. A megjelenítés egy **térképnézeten** történik. A térképnézet vonatkozási rendszerét előre meg kell választanunk. A térképnézetek több rétegből, **tematikákból** épülnek fel. Egy tematika az adatbázis objektumainak egy kiválasztott részhalmazát jelenti. A geometria megjelenítése sokféle lehet: *pontok*, *szimbólumok*, *vonalak*, *poligonok*, *feliratok*, *kartodiagramok*, *képek*. A megjelenítés három alapvető paramétere a **méret**, a **szín** és a rajzelem **típusa** (2-5. Táblázat).

	Pont	Vonal	Terület
méret			
szín			
típus			

2-5. Táblázat: A geometria megjelenítési lehetőségei

A megjelenítés paramétereinek hozzárendelése a rajzelemhez a következő módokon történhet:

- a leíró adatok csoportjaihoz **tematikus osztályokat** rendelünk. A megjelenítés paramétereit osztályonként módosíthatjuk. A tematikus osztályok listája képezi az adott tematika **jelmagyarázatát**.
- a szín, a méret és a típus jellemzőket a leíró adatok között tároljuk. A megjelenítésért felelős adatmezők alkotják a **kartográfiai adatbázist**.
- a szín, a méret és a típus jellemzőket a lekérdezés során kifejezések eredménye adja. Ezeket a lekérdezéseket kartográfiai lekérdezéseknek nevezzük.

A megjelenítéssel szemben támasztott további követelmények:

- méretarány függő megjelenítés, méretarány független feliratozás
- feliratok, kartodiagramok, képek átfedés mentes megjelenítése
- alapfelületek, vetületi rendszerek támogatása
- koordináta feliratok, örkereszt, őrháló, címek, aláírások, jelmagyarázatok kirajzolása
- raszteres és vektoros adatmodellek együttes megjelenítése
- felületmodell segítségével háromdimenziós kép készítése
- nyomdai színre bontás támogatása

3. Raszteres eljárások

A geoinformatikai rendszerekben a raszteres adatok megjelenítése, feldolgozása egyre nagyobb szerephez jut. A raszteres adatokat a következők jellemzik: gyors elemzések, az adatok előállítása viszonylag olcsó, az állományok naprakészek és nagy területről szolgáltatnak homogén adatokat (*Csornai, Dalia 1991*).

A raszteres adatok elérhetősége sokat változott az utóbbi időkben. A földrajzi információt felhasználók igénylik az aktuális és részletes raszteres állományokat. Számos geoinformatikai eljárás kizárólag távérzékelte raszteres adatokra támaszkodik (*felszínborítás térképezése, változás detektálás, természeti erőforrások és természeti károk felmérése*).

A fentiek alapján ezért én is fontosnak tartom nem csak a raszteres adatok megjelenítését, hanem a raszteres elemzések támogatását is egy geoinformatikai rendszerben. A *DigiTerra* program raszteres moduljának fejlesztése során számos szoftvertechnológia és új eljárás született. Ezek közül mutatnak be néhányat a következő fejezetek.

3.1. Algoritmikus képfeldolgozás

Az algoritmikus képfeldolgozás egy szoftvertechnológia. Lényege, hogy mindig a forrásképpen dolgozunk és az **eredményt algoritmusok sorozata állítja elő** valós időben. A valós idejű feldolgozást a gyors számítógépek és az algoritmusok gépi-, vagy gépközel kódra fordítása teszi lehetővé. A módszer előnyei:

- nincsenek köztes, ideiglenes adatállományok,
- az eredmény azonnal látható ezért számos változat kipróbálható,
- különböző típusú és felbontású felvételek egyidejű feldolgozása,
- hiba esetén nem kell a feldolgozási folyamatot újratekdenünk.

Az algoritmikus képfeldolgozás kifejlesztését ugyanaz a tény motiválta, mint a képtömörítési algoritmusok kidolgozását, egyrészt a helytakarékoság, másrészt a számítógép merevlemezen tárolt adatok olvasásának és írásának sebessége nem fejlődik olyan mértékben, mint a feldolgozó egység (*processzor*) teljesítménye. A *DigiTerra* programon kívül tudomásom szerint csak az *Er-Mapper* szoftver tartalmaz algoritmikus képfeldolgozást (*Er-Mapper 1998*).

Az algoritmikus **képfeldolgozás soronként** történik egy megfelelő vonatkozási rendszerben befoglaló koordinátaival adott téglalap alakú területen. A soronkénti feldolgozás gyors, minimális memóriaigényű és nagy méretű raszteres állományok elemzésére is használható. A feldolgozás alkalmazható átnézeti kép, nagyított kép megjelenítésére, de a teljes állomány feldolgozására is. *DigiTerra* környezetben a feldolgozás lépései a következők:

- Forrás megnyitása

A névvel azonosított adatállomány megnyitása után a következők történnek: az állomány típusának azonosítása (*BMP, PCX, LAN, TIFF, ERS, BIL, RAS ...*), a rasztersorok beolvasásához szükséges jellemzők meghatározása, a georeferencia adatok betöltése. A program fejlesztése során a különböző raszteres állományok azonosítása, megnyitása, betöltése sok problémát okozott. Ennek oka elsősorban az állományok szegényes leírásában keresendő. Az egyik legproblémásabb adatformátum

a nagyfokú szabadsága miatt a TIFF formátum. A legkönnyebben a BMP, LAN és BIL formátumú fájlokat lehet értelmezni.

- Sávok kiválasztása

A modul korlátlan rasztersávot tud kezelni egyszerre. Lehetőség van multi- és hiperspektrális felvételek feldolgozására is. A sáv orientált megközelítés lehetővé teszi eltérő időpontból származó és különböző terepi felbontású raszterek együttes feldolgozását. Összehasonlítás céljából megemlítem, hogy más rendszerekben az eltérő forrásból származó, vagy eltérő felbontású felvételek összedolgozása mindig problémás és ideiglenes állományok sorát igényli.

- Képsorok betöltése

A befoglaló koordinátaival adott téglalap alakú területre a betöltés soronként történik. A betöltés előtt meg kell határozni a képsor adatállományban elfoglalt pozícióját. A képsor betöltését minden érintett sávra el kell végezni.

- Képsorok kibontása

Néhány állomány esetében a feldolgozás előtt szükséges a tömörített rasztersorok kibontása (*PCX, TIFF, JPEG*). A kicsomagolás csak egy kitérőt jelent és nem szakítja meg a feldolgozás folyamatát.

- Fokális funkciók

Az eredmény pixel a forrás pixel és meghatározott közelségű környezete alapján készül el. Fokális funkciók esetében szükséges a korábban beolvasott néhány sor megtartása, illetve néhány sorral előre olvasás is a szomszédos pixelek elérése érdekében. Fokális funkciók például a következők: *simítás, élesítés, lejtés, kitettség, magasan – közepesen – alacsonyan vágó szűrők, statisztikák stb.*

- Mintavétel

A betöltött képsor befoglaló koordinátái általában nem egyeznek meg a kívánt terület befoglaló koordinátaival, ezért szükséges a képsor mintavétele. Kicsinyített kép előállításakor az adatállományban sorokat, illetve pixeleket át kell ugrani, míg nagyított képnél a beolvasott sorokat, illetve pixeleket többszörözni kell.

- Lokális funkciók

Az eredmény pixel az egyes sávok között értelmezett algebrai vagy logikai kifejezés eredménye. A lokális funkciókat gép közeli kódra fordított kifejezések és függvények hajtják végre. Ilyen funkciók az *összeadás, hányados, átkódolás, kiemelés, stb.*

- Régiók és üres cella

Az eredmény raszter nem csak téglalap alakú lehet. Sok esetben szükséges, hogy a raszter egy szabálytalan, töréspontjaival adott sokszögön belül jelenjen meg. További igény az üres cellák támogatása. Az üres cella értéke minden sáv esetében megadható. Az üres cella nem vesz részt az eredmény pixel színértékének kialakításában és kirajzolásakor a cella alatti terület érintetlen marad.

- Képosztályozás

A képosztályozást a tanulóterületek kijelölése, a tanulóterületek alapján az osztályok felállítása és az osztálystatisztikák kiszámítása előzi meg. Ha az egyes osztályok statisztikai rendelkezésre állnak (*átlag, szórás, kovariancia*), akkor a pixelek osztályozása elvégezhető. A támogatott osztályozási eljárások a következők: *átlagtól való minimális távolság, legnagyobb valószínűség, korreláció vizsgálat, spektrális szög térképező*.

- Globális funkciók

Globális funkciók a fokálishoz hasonlóan kivételes kezelést igényelnek. Ebben az esetben szükséges a teljes raszter vagy raszterblokkok betöltése. Ilyen funkciók például: *láthatósági viszonyok, hidrológiai elemzések, képszegmentálás, statisztika számítások*.

- Kontraszt és fényerő beállítás, színkeverés

Távérzékelte felvételek a légkör miatt kontrasztszegények. Az algoritmikus képfeldolgozás lehetőséget ad a kontrasztgazdag kép való idejű előállítására. A kontraszton kívül lehetőség van a fényerő és az átviteli görbe beállítására is. A kontraszt beállítható a kép egészén, a kép aktuálisan megjelenített részletén és lokálisan is egy pixel adott környezetén belül.

- Statisztika

Az előállított képről minden esetben statisztika készül (*pixelszám, átlag, szórás, kovariancia, minimum, maximum*), mely számos további eljárás bemeneti adatát képezi (*például: főkomponens transzformáció, standardizálás, osztályozás ...*).

Ezek után az algoritmikus képfeldolgozás elérése a következő három utasítással történik. Az utasítások elrejtik a felhasználó elől a képfeldolgozás komplexitását:

```
StartRasterRead(R, x1, y1, x2, y2, w, h);
```

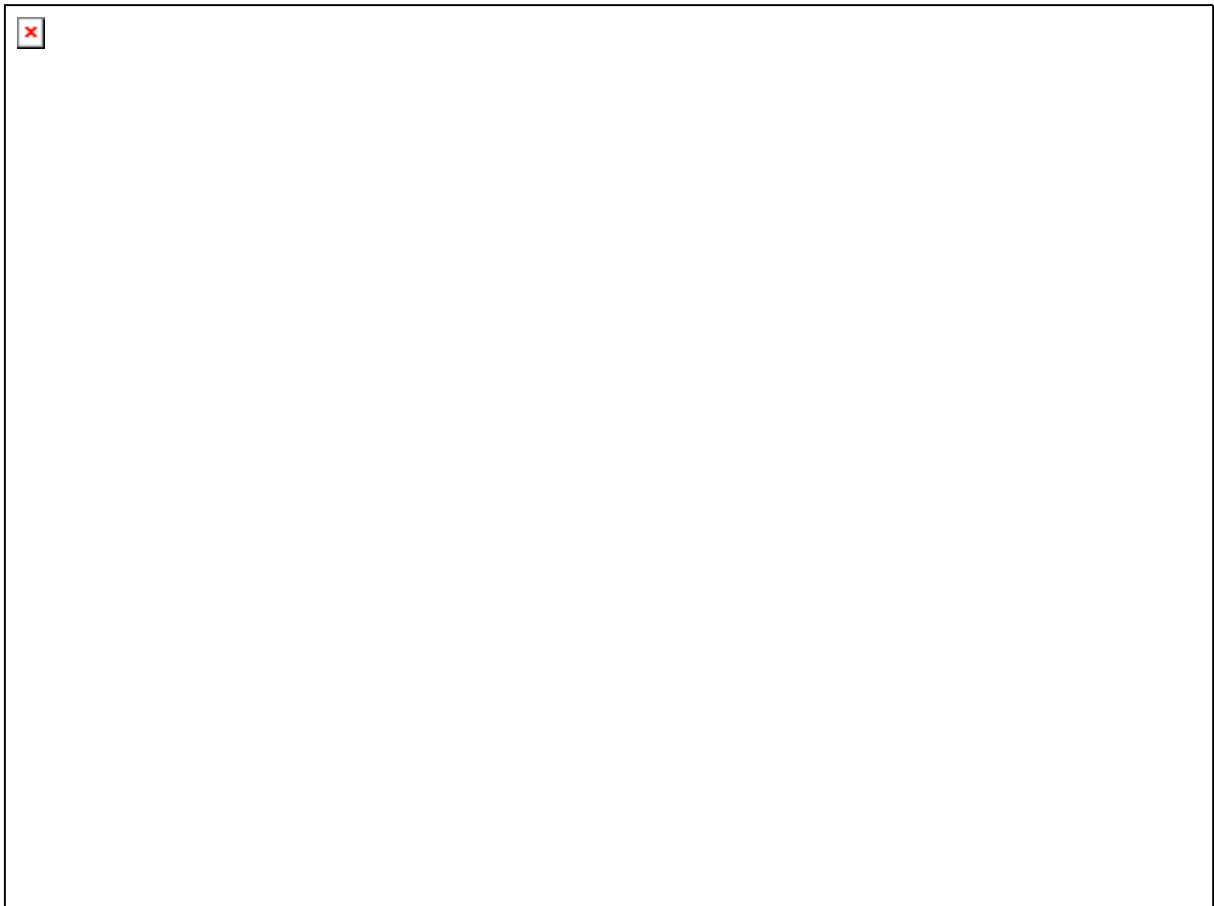
R raszter megnyitása olvasásra. A betölteni kívánt terület befoglaló koordinátáit az *x1*, *y1* és *x2*, *y2* koordináták írják le. Az eredmény raszter mérete *w* szélességgel és *h* magassággal adható meg.

```
DoRasterRead(R);
```

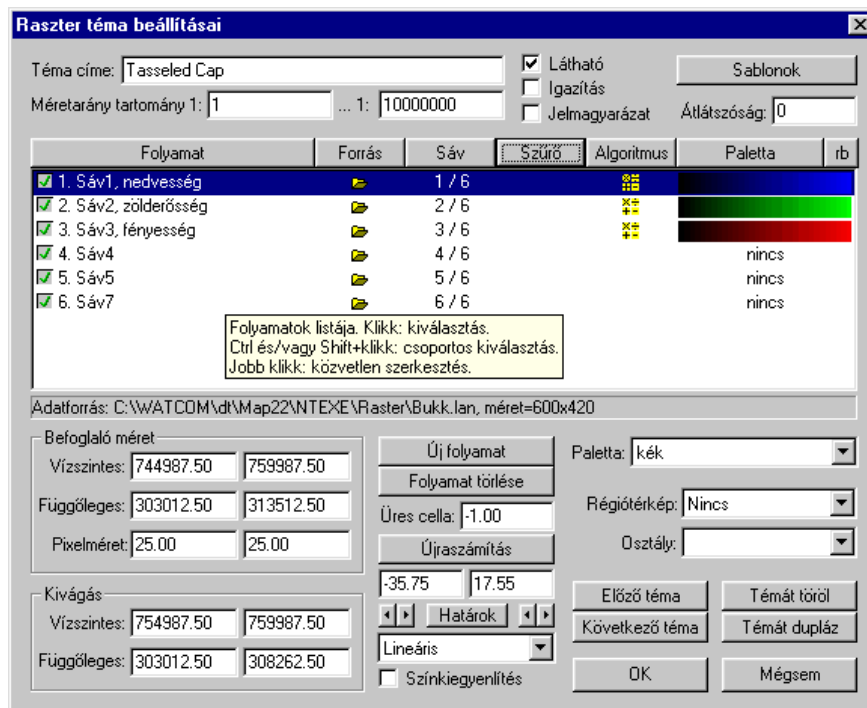
R raszter következő sorának olvasása és feldolgozása. Az eljárás az előbbi jelölés alapján *w* darab pixelt olvas be és dolgoz fel. Az eljárást az előbbi jelölés alapján *h* alkalommal kell meghívni, hogy a befoglaló méretre a raszter előálljon.

```
EndRasterRead(R);
```

R raszter lezárása, beolvasáshoz szükséges tárterület felszabadítása.



3-1. Ábra: Ugyanazon terület hat részletének algoritmikus képfeldolgozása



3-2. Ábra: A párbeszédpanel mögött egy teljes képfeldolgozó modul húzódik meg

A *DigiTerra* algoritmikus képfeldolgozó modulja intuitív és hatékony környezetet nyújt a digitális felvételek elemzéséhez (3-1. *Ábra*). Az algoritmikus képfeldolgozó képes helyettesíteni egy bonyolult, több menüpontból álló képfeldolgozó programot is (3-2. *Ábra*). Több mint 100 beépített függvénye, több mint 50 raszteres szűrője és több mint 50 előre elkészített képfeldolgozó algoritmus mellett a felhasználó újakat is definiálhat. A függvények, szűrők és algoritmusok részletes ismertetésére azonban itt nincs lehetőségem.

3.2. Raszter formátum

Egy korszerű digitális képfeldolgozó szoftvert fel kell készíteni a raszteres adattípusok széles skálájának fogadására. Jelenleg nem tudok olyan digitális képformátumról, amely a következő összes igényt kielégítené:

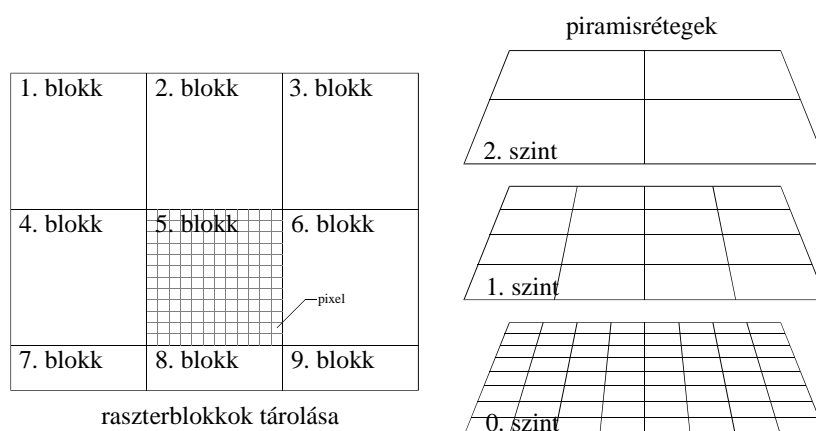
- 1, 4, 8, 16, 32 és 64 bites előjeles egész számok tárolása cellánként
- 1, 4, 8, 16, 32 és 64 bites előjeltelen egész számok tárolása cellánként
- 32 és 64 bites valós számok tárolása cellánként (*terepmodell*)
- korlátlan számú rasztersáv támogatása (*hiperspektrális felvételek*)
- georeferencia adatok tárolása az adatállományban
- statisztikai és megjelenítési adatok tárolása az adatállományban
- blokkok vagy sorok közvetlen elérése (*gyorsítás*)
- átnézeti kép és piramisrétegek támogatása (*gyorsítás*)
- adatvesztés nélküli tömörítési módszerek támogatása (*helytakarékoság*)
- adatvesztéssel járó tömörítési módszerek támogatása (*helytakarékoság*)
- metaadatok tárolása az adatállományban

Az általam kidolgozott raszteres formátum az előbbi igények teljesítését célozza meg. A formátum *DigiTerra Raszter* névre hallgat. Az adatállományok neve *RAS* kiterjesztésre végződik. A raszteres állomány jellemzői a bináris kódolású adatfájlban és nem egy külön szövegfájlban tárolódnak. A raszteres állomány jellemzésére a következő tulajdonságok szolgálnak:

- | | |
|-------------------|--|
| • Szélesség | pixeloszlopok száma |
| • Magasság | pixelsorok száma |
| • Sávszám | rasztersávok száma |
| • Cellatípus | bitszélesség, előjeles típus, valós típus, bővíthetőség |
| • Befoglaló méret | raszter bal-alsó és jobb-felső sarokpontjának koordinátái |
| • Alapfelület | raszteres állomány vonatkozási rendszerének alapfelülete |
| • Vetület | raszteres állomány vonatkozási rendszerének vetületi rendszere |
| • Blokkméret | pixeladatok tárolása blokkméret nagyságú négyzetekben |
| • Piramisréteg | piramisrétegek száma |
| • Tömörítés | tömörítés típusa |
| • Kvantálás | adatsorok bitszélességének csökkentése |
| • Statisztika | rasztersávok statisztikai adatai (<i>átlag, szórás, kovariancia</i>) |
| • Metaadat | az adatállomány leírása, jellemzése, sávok leírása, sávkiosztás |

Bővebb magyarázatot igényel a blokk és piramisréteg kialakítás, valamint a tömörítés. A blokkok és piramisrétegek bevezetését a nagy méretű raszteres állományok hatékony kezelése indokolja. A gyors képfeldolgozáshoz mindkét módszerre szükség van. A blokk technika a kép részleteinek, míg a piramis technika a kép egészének elérését optimalizálja.

A **blokk technika** lényege, hogy a képet nagyobb, általában $64 \cdot 64$, $128 \cdot 128$ vagy $256 \cdot 256$ méretű blokkokra daraboljuk és a pixeleket a blokkon belül már sorokba és oszlopokba rendezve tároljuk (3-3. Ábra). Hagyományos tárolás esetén a kép egy részletnek betöltése néhány rasztersor kiválasztását és a soron belül az érintett sorrészlet betöltését jelenti. Ez számos adatfájlon belüli pozícionálást jelent. A háttértárolón a pozícionálás sokkal több időt vesz igénybe, mint az olvasás. A blokkok bevezetése ezeket a pozícionálásokat minimalizálja. Egy képrészlet megjelenítése néhány blokk beolvasásával megoldható. A pozícionálás ilyenkor csak a blokk beolvasása után szükséges. Ha a hagyományos tárolású raszteres állományból h sort akarunk beolvasni, akkor ez $h \cdot \text{sávszám}$ pozícionálást jelent. A blokk orientált tárolás esetén ez mindössze $h / \text{blokkméret}$ pozícionálást igényel. Ez előbbiek számokkal megvilágítva: 1800 pozícionálás áll szemben 5 pozícionálással. Az előbbi fejtegetés után kijelenthető, hogy a képrészletek betöltése, a raszter 1:1 vagy nagyobb méretarányú elérése a blokk technika bevezetésével jelentősen gyorsítható.



3-3. Ábra: Blokk és piramis technika

Míg a blokk technika előnyös a kép részleteinek előhívásában, nagyon hátrányos az átnézeti képek megjelenítésében. Ha a kép csak blokkokat tárol, mint például néhány TIFF változat, akkor a kép egészének megjelenítéséhez a teljes rasztert be kell olvasni. Az átnézeti képek előállítására sorok és oszlopok elhagyásával történik, amely a **piramis technikával** gyorsítható (3-3. Ábra). A piramis technika lényege, hogy az átnézeti képek több lépésben készülnek el. Minden egyes átnézeti kép az előző képből származtatható minden második sor, illetve minden második oszlop elhagyásával vagy két-két sor, illetve oszlop átlagolásával. Az utóbbi módszer valamivel számításigényesebb, de szebb eredményt ad. Az átnézeti kép szélessége és magassága az eredeti kép szélességének és magasságának a fele. Mivel fél sort nem lehet tárolni, a felezés az egész aritmetika szabályai szerint történik. A felezés elvileg addig folytatódik, amíg a szélesség és a magasság nagyobb, mint nulla. A gyakorlatban azonban $32 \cdot 32$ méret alá nem célszerű menni. A határérték számítás alapján a piramisirétegek mérete az eredeti képméret harmada. A többlet helyigény az átnézeti kép megjelenítésének gyorsaságban megtérül. Nézzük a következő példát:

Ha a képméret $1040 \cdot 850$ pixel, akkor az átnézeti képek mérete:

1. Réteg $520 \cdot 425$ pixel,
2. Réteg $260 \cdot 212$ pixel,
3. Réteg $130 \cdot 106$ pixel,
4. Réteg $65 \cdot 53$ pixel,
5. Réteg $32 \cdot 26$ pixel.

A *DigiTerra Raster* formátumban a két előbbi technikát ötvöztem, illetve a piramis technika helytakarékos változatát is megvalósítottam. A **blokk és piramis technika ötvözése** abban rejlik, hogy az egyes piramisrétegeket is a blokk technikával tároljuk. Az előnyök $4000 \cdot 4000$ vagy nagyobb méretű raszteres képeknél jelentkeznek. A **helytakarékos piramis technika** lényege, hogy a kép mérete nem növekszik a piramisrétegek méretével, ugyanakkor a speciális tárolásnak köszönhetően a piramisrétegek mégis elérhetők (*egy 1.2Gb méretű raszter esetén a piramisrétegek mérete 300Mb*). A speciális tárolás azt jelenti, hogy egy piramisréteg a szükséges pixeladatok $\frac{3}{4}$ részét tárolja és $\frac{1}{4}$ részét a fölötte elhelyezkedő piramisrétegből veszi. A blokk technika és a helytakarékos piramis technika rendkívül előnyösen ötvözhető a diszkrét *Wavelet* transzformációval, az általam is használt veszteséges tömörítési technikával.

Piramis technikát sok képformátum támogat (ECW, IMG, RRE). Blokk technikát alkalmaz a TIFF 6.0 formátum. Piramis és blokktechnikát együttesen nagyon kevés, általában csak *Wavelet* tömörítésű képformátumok alkalmaznak. Ezek a formátumok viszont csak 1 vagy 3 sávós és 8 bites raszteres adatábrázolást támogatnak. Az általam kidolgozott helytakarékos piramis technikával a szakirodalomban eddig még nem találkoztam.

A raszteres **tömörítési algoritmusok** fejlesztését és alkalmazását az előző fejezetben már kifejtett tény indukálja. Bizonyos raszteres adatok esetében megkívánjuk a veszteségmentes tömörítést, ilyenek például a raszteres tematikus térképek. Raszteres felületmodellek, digitális légi- és űrfelvételek veszteséges tömörítése viszont előnyös.

A *DigiTerra Raster* formátumban a veszteségmentes tömörítés első lépése a differencia kép előállítás. A differencia kép úgy áll elő, hogy minden egyes pixel digitális értéke helyett az aktuális és a tőle balra lévő pixel digitális értékeinek előjeles különbségét tároljuk. Az első oszlop pixeleit változatlanul hagyjuk. A különbség értékek már jóval kisebb bitszélességen ábrázolhatók. Sok esetben a különbség értékek azonosak, legtöbbször nullák. A különbségek és az ismétlések kódolására speciális *Huffmann* táblázatot alakítottam ki. A *Huffmann* táblázatban a gyakori értékeket rövidebb, míg a ritkébbakat hosszabb bitsorozat írja le.

A raszteres adatok veszteséges tömörítése jelenleg a **diszkrét Wavelet transzformációval** a leghatékonyabb. Egy diszkrét transzformáció együtthatóit *Daubechies* (1988) definiálta. A diszkrét *Wavelet* transzformáció lényege, hogy a digitális értékeket, mint jelsorozatokat két azonos méretű részre különítjük el. Ez a két rész a lényegi és a különbség rész, vagy más szóval integrál és differenciál rész. A szétválasztott jelsorozat lényegi részén az átalakítás ismét elvégezhető mindaddig, amíg a jelsorozat hossza egynél nagyobb. Az egymás utáni felezés miatt a jelsorozat méretét a szakirodalom kettő hatványában jelöli meg, de tetszőleges méretű jelsorozatot is feldolgozhatunk, ha a páratlan hosszúságú jelsorozat utolsó pixelét elhagyjuk. A diszkrét *Wavelet* transzformáció munkaképletei alapján (*Závoti 1996*) a következő C programvázlatot dolgoztam ki:

```
Wavelet(v, L, M, inc) // v vektor, L méret, M momentum, inc növelés
{ l=1<<L; h0=daub+M*(M-1); g0=daub+M*(M+1)-1; // méret, együtthatók
  for(i=0; i<L; i++) // Szintek
  { for(j=0,a=buf,b=v; j<l; j++,a++,b+=inc) *a=*b; // segédbuffer
    l/=2; // a vektort két felé választjuk
    for(j=0,a=v,b=v+l*inc; j<l; j++,a+=inc,b+=inc) // vektor elemek
    { *a=*b=0; c=2*j; p=buf+c; h=h0; g=g0; // indexek
      for(k=0; k<M; k++) // összesítés
      { *a += *h++ * *p; *b += *g-- * *p; c++; p++; // integrál
        *a += *h++ * *p; *b -= *g-- * *p; c++; p++; // differenciál
        if(c==2*1) { c=0; p=buf; } // index nullázása
      }
    }
  }
}
```

```

    }
  } // end for vektorelemek
} // end for szintek
}

```

Ezek után az inverz diszkrét Wavelet transzformáció a következő:

```

iWavelet(v, L, M, inc) // v vektor, L méret, M momentum, inc növelés
{ l=1; h0=daub+M*(M-1); g0=daub+M*(M+1)-1; // méret, együttthatók
  for(i=0; i<L; i++) // Szintek
  { for(j=0,a=buf,b=v; j<2*l; j++,a++,b+=inc) *a = *b; // segédbuffer
    for(j=0,a=v,b=v+inc; j<l; j++,a+=2*inc,b+=2*inc) // vektor elemek
    { *a=*b=0; c=j; p=buf+c; h=h0; g=g0; // indexek
      for(k=0; k<M; k++) // összesítés
      { *a += *h++ * *p + *g-- * *(p+1); // integrál
        *b += *h++ * *p - *g-- * *(p+1); // differenciál
        if(c==0) { c=l-1; p=buf+c; } else { c--; p--; } // index jav.
      }
    } // end for vektorelemek
    l*=2; // a vektor a kétszeresére növekszik
  } // end for szintek
}

```

Képek esetében a jelsorozat két dimenziós. Az átalakítást a **kép minden során és oszlopán** tetszőleges sorrendben el kell végeznünk. A C program hatékonyságának köszönhető, hogy az előbbi programkódok alkalmasak egy és kétdimenziós jelsorozatok transzformálására is. Az átalakított kép különbség részében az értékek nullához közeli lesznek. A tömörítés lényege, hogy az átalakított kép digitális értékeit egy előre megadott számmal osztjuk és a tört részt elhagyjuk, majd az így kapott egész értékeket tömörítjük. Az osztást **kvantálásnak** nevezzük, célja a bitszélesség csökkentése. Az átalakított értékek, mint bitsorozatok tömörítése legegyszerűbben egy *Huffmann* táblázattal és egy ismétlő kóddal történhet. Az ismétlő kód megadja, hogy az aktuális digitális értéket hányszor kell ismételni.

A diszkrét *Wavelet* transzformációval 1:10 ... 1:25 tömörítési arányt tudunk elérni. Ez a tömörítési arány még viszonylag jó képminőséget biztosít. A transzformáció sajátossága, hogy a tömörítés előállítja az egyes piramisrétegeket is. Ezért a kép kibontását csak a szükséges szintig kell elvégeznünk. A legfelső szinten a raszteres kép átnézeti képét találjuk.

3.3. Raszteres elemzés

A raszteres elemző eljárások olyan összetett földrajzi elemzéseket takarnak, melyben a lokális, fokális és globális raszter funkciók sorozata állítja elő a kívánt eredményt. A korábbi fejezetben említett algoritmikus képfeldolgozó önmagában is végtelen variációjú elemzés elvégzésére ad lehetőséget. Itt elsősorban az általam kidolgozott, a raszteres funkciók közül a globális funkciók csoportjába tartozó néhány eljárást kívánok ismertetni. Az eljárások mindegyike kapcsolatban áll a raszteres terepmodellel, vagy előállítja azt, vagy kiindulási adatforrásként használja fel.

3.3.1. Piramis interpoláció

Gyakran előfordul, hogy háromdimenziós pontokból, vagy szkennelt térképekről automatikusan vektorizált szintvonalakból kell előállítani raszteres felületmodellt. Mindkét esetben a **bemeneti adatok nagy száma** (*több millió pont*) miatt a korábbi térbeli interpolációs módszerek alacsony hatékonyságúnak bizonyulnak. A hagyományos módszerek alatt a következő raszteres interpolátorokat értem: *távolság hatványával súlyozás*, *Krigelés*, *polinomiális regresszió*. Főleg a szintsíkba felemelt vektorizált szintvonalak interpolálása időigényes ezekkel az interpolátorokkal. Ebben az esetben a feladat akár százmillió pont betöltése, indexelése és interpolálása. A hagyományos módszerek időigénye általában $\sigma(n^2)$ és $\sigma(n^3)$ között van, ahol n a pontok száma. Egy lehetséges alternatíva a szabálytalan háromszöglefedés, melynek időigénye $\sigma(n \cdot \log n)$, de itt az eredményt raszterizálni kell, ami viszont ront a hatékonyságon. Nagy adatsűrűség esetén a háromszöglefedés tárígénye meghaladhatja az eredmény raszter méretét is.

Mikor több szkennelt szintvonalas térképszelvény alapján kellett felületmodellt előállítani, gondolkodtam el egy kizárólag raszteres alapokon futó algoritmuson. Az eljárás lényege, hogy a **bemeneti adatokat** közvetlenül az **eredmény raszteren** helyezi el és az interpoláció is az eredmény raszteren történik. Pont esetében a pont magasságát a pont koordinátái alapján kijelölt rasztercellába kell írni. Vonal esetében a vonal kezdőpontjának és végpontjának koordinátái alapján kijelölt cellák között kell egyenest húzni és az érintett cellákba az egyenes magasságát elhelyezni.

A bemeneti adatok feldolgozása után egy hézagos raszter áll elő sok üres pixellel. Az interpolátor feladata, hogy az üres pixelekhez, amelyekhez nem tartozik magasság, értéket rendeljen a közeli nem üres pixelek alapján. Az üres pixelek kitöltése **piramisrétegek** segítségével történik. Egy magasabb piramisréteg pixelének képzése az alatta lévő réteg $2 \cdot 2$ pixelének átlagolásával történik. Az átlagolásban nem vesznek részt az üres pixelek, amelyekhez nem tartozik magasság. Ha volt legalább egy bemeneti magasság, akkor a sorozatos összevonás után elérkezünk egy olyan piramisréteghez, amelyben már nem lesznek üres pixelek.

Ezután megkezdődik a **lebontási szakasz**, amely során az alacsonyabb piramisrétegek üres pixeleinek értékét a fölötte levő piramisréteg átlagolt pixeleiből lokális interpolátorokkal határozzuk meg. A **lokális interpolátor** lehet bilineáris vagy bikubikus. A bilineáris interpolátor a felső réteg $2 \cdot 2$ pixelére fektetett elsőfokú, míg a bikubikus interpolátor a felső réteg $4 \cdot 4$ pixelére fektetett harmadfokú felülettel határozza meg az aktuális réteg pixelértékét. A lebontási szakasz akkor ér véget, mikor elérjük a legalsó szintet.

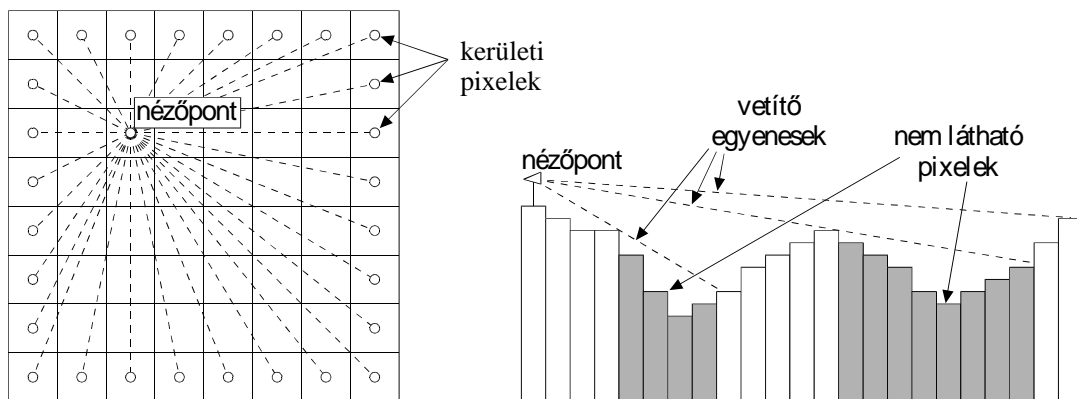
A piramis interpoláció elsősorban nagy sűrűségű és szabálytalan elhelyezkedésű ponthalmaz interpolációjára használható. Kevés pont esetén néhány hagyományos eljárás szebb eredményt produkál, sok pont esetén viszont az egyes módszerekkel készített raszterek közötti különbség minimális. A piramis interpoláció gyors, a mai számítógépeken pár másodperc alatt képes egy $5000 \cdot 5000$ méretű raszteres állományt előállítani. Kijelenthető, hogy az adatok betöltése és az eredmény raszter mentése több időt vesz igénybe, mint maga az interpoláció.

3.3.2. Láthatósági vizsgálatok

A láthatósági vizsgálat célja, hogy a raszteres terepmodellel megjelöljük azokat a pixeleket, amelyek egy vagy több nézőpontból láthatók. Az eljárás szempontjából fontos a nézőpont magasságának megadása. Sokszor a magasságot a terepfelszín fölötti relatív magassággal definiáljuk. A hagyományos eljárások minden egyes pixelt megvizsgálják, hogy valamelyik nézőpontról látható-e vagy sem. Az eljárás műveletigénye $\sigma(n \cdot s^3)$, ahol n a nézőpontok száma, s pedig a raszter egy sorában és egy oszlopában szereplő pixelek számának átlaga.

Az általam kidolgozott módszer lényege, hogy minden egyes pixel helyett csak a raszter területén lévő pixeleket vizsgálja. Az eljárás műveletigénye az előbbi jelölések alapján $\sigma(n \cdot s^2)$. Az algoritmus legfelső ciklusa a nézőpontok felsorolása. A következő ciklus a kerületi pixeleket veszi sorra. A tényleges vizsgálat a legbelső ciklusban történik, melyen a nézőpontot és a kerületi pixelt egy vetítő egyenessel kötjük össze.

A vetítő egyenest egységvektor segítségével írjuk fel. Az egyenesen a nézőpontból kiindulva egységvektor távolságokra történnek a vizsgálatok. Ha az egyenes aktuális pontja a terep fölött helyezkedik el, akkor ez a pixel az adott nézőpontról nem látható. Ellenkező esetben a pixel a nézőpontról látható, és az egységvektor magassági összetevőjét újra kell számolni. Ha a nézőpont a terepfelszín fölött helyezkedik el, akkor a vetítő egyenes meredeksége a nézőponttól a kerületi pixelig tartó vizsgálat során folyamatosan emelkedni fog. Minden bizonnyal az algoritmus könnyebben megérthető a 3-4. Ábra segítségével:



3-4. Ábra: Láthatóság vizsgálat: bal oldalon egy felülnézeti kép, jobb oldalon egy függőleges metszet a nézőpont és egy kerületi pixel között

A nézőpontból kiinduló sugaras pásztázáshoz hasonló algoritmust a szakirodalomban is olvashatunk. A kerületi pixelek pásztázása, a vetítőegyenest folyamatos emelése a számítások minimalizálása érdekében viszont saját szoftvertechnológia. A láthatósági vizsgálat kommunikációs hálózatok telepítésében, összelátási vizsgálatokban, és, mint a későbbiekben látni fogjuk, ortofotó készítésben jut nagy szerephez.

3.3.3. Hidrológiai elemzések

A hidrológiai elemzés elsődleges célja, hogy a vizsgált terület raszteres terepmodelljén meghatározzuk a **vízlefolási irányokat**. A lefolási viszonyok ismeretében elkülöníthetők a **vízgyűjtő területek** és meghatározható a **vízösszefolyási térkép**. A vízösszefolyási térkép segítségével felkereshetők a vízrajzi elemek, úgymint patakok, folyók, tavak, és meghatározható azok vízhozama. Az elemzés minden egyes eredményéről külön statisztikák készíthetők, például vízgyűjtő területek nagysága, összefolyt vízmennyiség, vízrajzi elemek méretei stb. Amint látható a hidrológiai elemzés szerteágazó, de korántsem teljes körű. Számos veszteséggel (*intercepció, párolgás, vízgyűjtők*), illetve alternatív úttal (*karsztvíz, rétegvíz, talajvíz*) kell számolni. Nagyon nehéz minden paramétert figyelembe vevő modellt felállítani, viszont a következőkben ismertetett eljárás jó közelítéssel tárja fel egy terület hidrológiai viszonyait. Az ismertetést egy saját szoftvertechnológia indokolja, mely az algoritmusokat egy elemi eljárásra vezeti vissza.

A hidrológiai elemzés első lépése tehát a vízlefolási irányok meghatározása. A kiindulási alap a terület raszteres terepmodellje. A **raszteres terepmodell** minden egyes pixele az általa lefedett terület egység átlagmagasságát hordozza. Egy adott pixel lefolási iránya ideális esetben a pixel közvetlen környezetében lévő pixelek magasságaiból határozható meg. A szakirodalom két módszert ismer. Az **első módszer** az egyszerűbb, mely a lefolási irányt a pixel nyolc közvetlen szomszédja közül abban jelöli meg, amelyik a legalacsonyabb. Ebben az esetben a pixel teljes vízmennyisége átadódik a lefolás irányába. A **második módszer** először meghatározza a szomszédos pixelek alapján a parciális deriváltakat, majd azokból számítja ki a lejtés nagyságát és irányát. Itt az átadandó vízmennyiség a lejtés nagyságának függvénye. A lejtés iránya egyértelműen meghatározható a parciális deriváltakból, viszont ez az irány egy lefolási sávot jelöl ki, amely több szomszédos cellát is érintet. A szakirodalomban megismert módszerek leggyakrabban egy táblázatot készítenek, amely megadja, hogy melyik szomszédos cellának mekkora vízmennyiséget kell átadni adott parciális deriváltak mellett.

A parciális deriváltakat numerikus módszerrel kell meghatározni. Az előző és a következő pixelértékek különbségének és a pixelek távolságának hányadosa adja a parciális deriváltakat. A durva hibákat három sor, illetve három oszlop parciális deriváltjainak átlagolásával lehet kiszűrni:

$$dx = \frac{\sum_{i=-1}^1 (r_{x+1,y+i} - r_{x-1,y+i})}{6 \cdot sx} \quad (3-1)$$

$$dy = \frac{\sum_{i=-1}^1 (r_{x+i,y+1} - r_{x+i,y-1})}{6 \cdot sy}$$

ahol: dx, dy a parciális deriváltak nagysága x és y irányban
 sx, sy a pixel mérete x és y irányban
 $r_{x,y}$ a raszter x. oszlopban és y. sorban lévő pixel magassága

Ezek után az $r_{x,y}$ pixel középpontjában a raszteres terepmodellre merőleges vektor a következő: $\underline{v}(-dx, -dy, 1)$. A lejtés nagysága és iránya a \underline{v} vektorból könnyen meghatározható:

$$\begin{aligned} \operatorname{tg} \alpha &= \sqrt{dx^2 + dy^2} \\ \lambda &= 90^\circ - \operatorname{atan2}(-dy, -dx) \end{aligned} \quad (3-2)$$

ahol: dx, dy a parciális deriváltak nagysága x és y irányban
 α a lejtő vízszintessel bezárt szöge
 λ a lejtés irányszöge, 0° =észak, 90° =kelet, 180° =dél, 270° =nyugat

(A képletben szereplő az $\operatorname{atan2}(y,x)$ függvény számítógépes környezetben implementált függvény, mely a két argumentum hányadosának arcus tangensét számolja ki. Használata azért előnyös, mert a függvény figyel az előjelekre, a nulla értékekre és az x,y koordináták polárszögét adja eredményül.)

A lefolyási irány meghatározása bizonyos esetekben nehézségekbe ütközik. Kijelenthető, hogy a legtöbb raszteres terepmodell nem a vizsgált terület hidrológiai elemeinek pontos leírására hivatott, ezért a modellben számos ellentmondás található. Az első ilyen ellentmondás a terület **lokális minimumpontjai**. Ezek a pontok elméletileg a vízgyűjtő területek kiinduló pontjai, a gyakorlatban viszont csak mélyedések, amelyeket, ha megtölt a víz, már nem viselkednek vízelnyelő pontként. A második probléma, hogy a raszteres terepmodellek magasságai egész aritmetikát használnak, tehát csak méteres vagy deciméteres magasságok különíthetők el. Ezért a modellben számos olyan pixel található, amelynek nincs alacsonyabb, hanem számos azonos magasságú szomszédja van. Az ilyen pixelek esetében a **lefolyási irány többértelmű**.

A lefolyási irány meghatározásakor az első feladat a területen belüli lokális minimumpontok és **lefolyástalan területek feltöltése**. A feltöltés egy iteratív folyamat. Minden egyes lépés előtt meghatározzuk a lefolyási irányokat, és ha van lefolyástalan vagy több lefolyással rendelkező pixel, akkor megkezdődik a feltöltés. A feltöltés kiinduló pontja maga a lefolyástalan pixel. A kiinduló pontból egy *Elöntés* algoritmussal kell meghatározni azt a pixelt, amelyik a kiinduló pixel magasságától eltér és a legkisebb.

Az **Elöntés** (*FloodFill*) algoritmus számítógépes környezetben jól ismert iteratív eljárás. Lényege, hogy egy pontból kiindulva elöntés szerűen sorolunk be újabb és újabb pixeleket, amelyek egy adott feltételnek megfelelnek.

A legkisebb pixel meghatározása után a lefolyástalan pixelt és a pixellel azonos magasságú környező pixeleket a kikeresett legkisebb pixel magasságára kell emelni. Az iteráció végén a raszteres terepmodell belsejében nem található lokális minimumpont, csak a terepmodell szélein. A feltöltött és az eredeti modell különbsége megadja a feltöltés mértékét, amely számos további elemzés tárgya lehet.

A feltöltés után a **lefolyási irány** még nem tökéletes, ugyanis vannak olyan pixelek, amelyek esetében a lefolyási irány többértelmű, de ezek nem lefolyástalan területek, hanem **teraszok és platók** részei. A többértelműség az ún. *Felfolyási* algoritmussal oldható fel. Az algoritmus furcsa neve pontos leírást takar. Mivel a víz lefolyása bizonyos pixeleken többértelmű, ezért a raszteres terepmodell szélein elhelyezkedő lokális minimumpontokból kell a lefolyást pixelről-pixelre fordított irányban követni. A követés ismét az *Elöntés* algoritmussal határozható meg hatékonyan. A minimum ponthoz, illetve a már besorolt pixelekhez iteratív hozzá kell kapcsolni mindazokat a pixeleket, amelyek az adott pixelnél magasabbak vagy azonos magasságúak. A hozzákapcsolásnál fontos a sorrend, először a főirányokat, majd az

átlós irányokat kell vizsgálni. Ha nem ebben a sorrendben kapcsoljuk a pixeleket, akkor a lefolyási vonalak nem a legrövidebb lefolyási irányokat fogják követni.

A lefolyási irány kissé bonyolult meghatározása után már valamivel egyszerűbb feladat a **vízösszefolyási térkép** előállítása (3-5. *Ábra*). A vízösszefolyás az egyes pixeleken átfolyt összes víz mennyiségét jelenti. A módszer meglehetősen számításigényes. Minden egyes pixelt le kell vezetnünk a lokális minimumpontokig. A levezetés irányát az előbbieken meghatározott lefolyási irányok szolgáltatják. Levezetés közben az érintett pixelek számlálójához hozzá kell adnunk az egységnyi vízmennyiséget. Miután minden egyes pixelt megvizsgáltunk, a számlálók pontosan a pixeleken átfolyt víz mennyiségét fogják visszaadni. Mindkét lefolyási irány számítási módszer esetén használható ez az eljárás. Az összefolyási térkép előállítása után már egyszerűen meghatározhatók a vízrajzi elemek és a vízhozamok. A vízrajzi elemek egy lehetséges elkülönítése a 3-1. *Táblázattal* történhet.

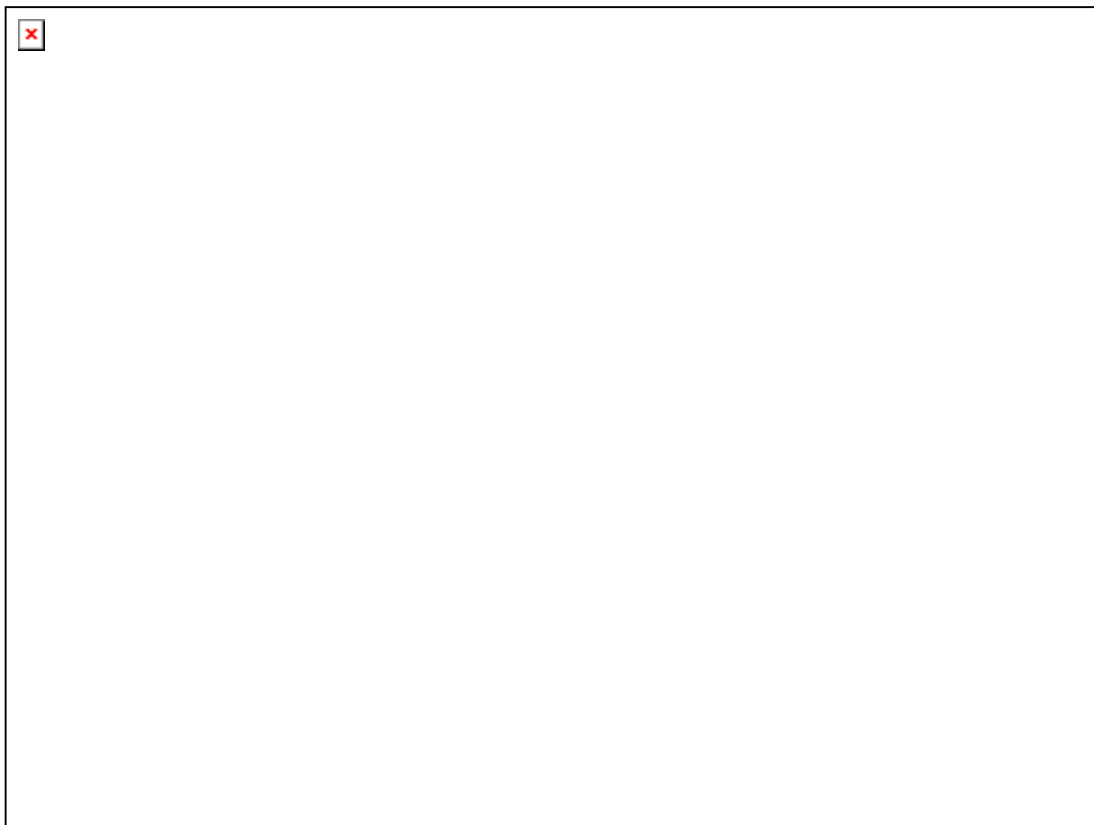
Átfolyt vízmennyiség	Vízrajzi kategória
0-50 egység	Átlátszó kategória
50-100 egység	Időszakos vízfolyás
100-500 egység	Patak
500-2000 egység	Folyó
2000- egység	Folyam

3-1. *Táblázat: Vízrajzi elemek osztályozási táblázata*

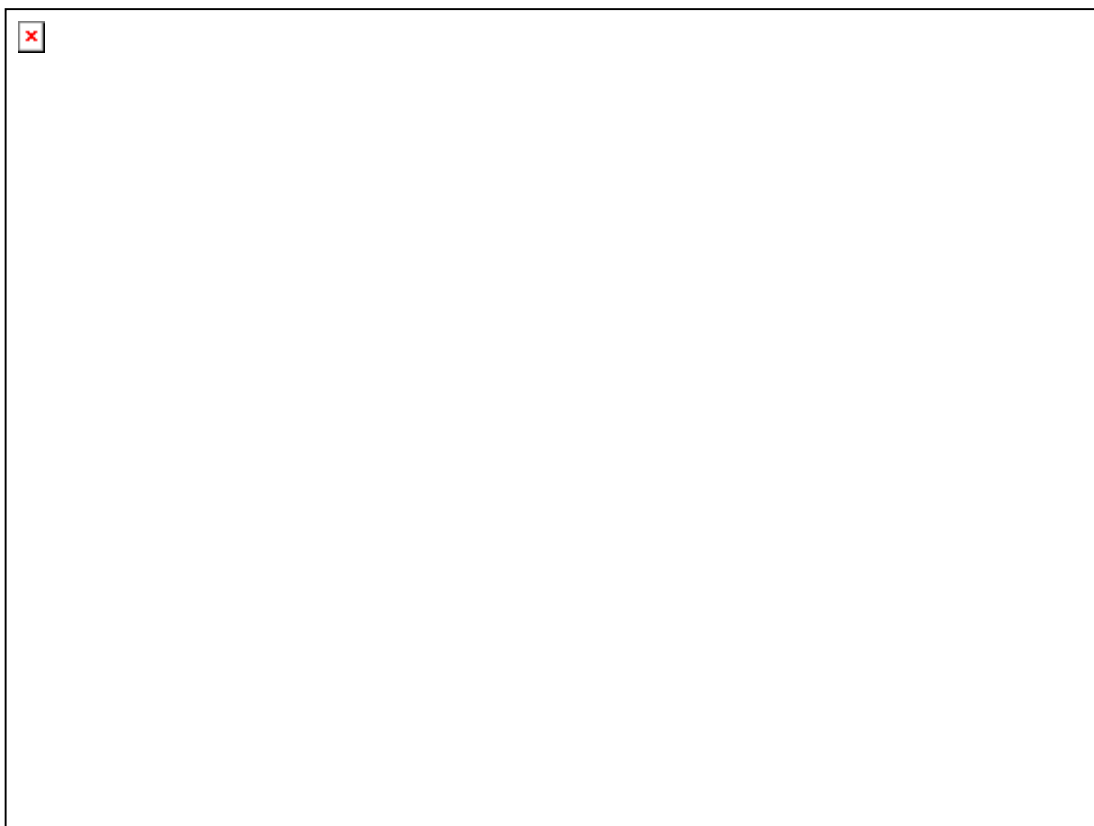
A kategóriákba sorolt raszteres vízrajzi elemeket raszter-vektor átalakítással (*vékonyítás, pixelösszevonás*) vektoros geometriai elemekre lehet konvertálni. A vízhozamokat a pixeleken tárolt vízmennyiségek adják.

A **vízgyűjtő területek** szintén a lefolyási irányokból határozhatók meg (3-6. *Ábra*). Az algoritmus a *Felfolyási* algoritmussal azonos. A kiinduló pontok itt is a terület lokális minimumpontjai. Az egyetlen különbség, hogy a lokális minimumpontokat egyedi sorszámmal látjuk el és a besorolt pixelekhez nem a lefolyási irányt, hanem a sorszámot rendeljük. Lehetőség van a vízgyűjtőterületek darabolására a vízfolyam bal-, illetve jobb oldala alapján, továbbá a vízfolyamok összefolyási pontjaiban. A vízgyűjtő területek meghatározása után számítható a vízgyűjtő terület nagysága, vízhozama, domborzati viszonyai stb.

A szakirodalom számos megoldást ismertet, amelyben a hidrológiai viszonyokat a terepmodellből kiindulva határozzák meg (*Arge és mások 1994, Tarboton 1996, 2001*). Saját munkámban a szoftvertechnológiai módszer kíván említést. A vízösszefolyási algoritmus kivételével minden algoritmust az *Elöntés* algoritmusra vezettem vissza. Ezáltal az eljárás átláthatóbb, hatékonyabb és tömörebb lett.



3-5. Ábra: Domborzatmodell a vízösszefolyási térképpel



3-6. Ábra: Vízyűjtő területek és a vízösszefolyási térkép

3.3.4. Árvíz- és gátmodellezés

Egy terület hidrológiai viszonyainak elemzésénél fontos szerephez jut a víz lefolyásának tanulmányozása mellett a terület **előntésének és gátrendszerének modellezése**. Az előntött területek nagyságának meghatározása, a gátak teherbírásának számítása fontos feladat a megfelelő védekezés kidolgozásához. A védekezés új gátak emelésében, a meglévő gátak javításában, magasításában, új hullámterek kijelölésében és ideiglenes tározók kialakításában nyilvánulhat meg. A védekezés mellett a modell az árterek, mint élőhelyek térképezésében is nagy szerephez jut.

Fontos megemlíteni, hogy az árvíz és gátmodellezés precíz, legalább **deciméter pontosságú raszteres terepmodellt** igényel. A raszteres terepmodell előállítására térbeli mérések interpolációjával történik. A térbeli mérések forrásai a következők lehetnek: digitalizált szakági térképek, terepi felmérések (*GPS mérés vagy tahimetria*), fotogrammetriai kiértékelések.

Az **előntési modell** az előbbieken ismertetett *Elöntés* algoritmussal hatékonyan megvalósítható. Az előntés kiindulópontja az előntött területen elhelyezett forráspont, amely tartalmazza az előntési szint magasságát. A forráspontból kiindulva a szomszédos pixeleket akkor kell az előntött területhez sorolni, ha a szomszédos pixel magasabb, mint az aktuális pixel, de alacsonyabb a forráspont magasságánál.

A **gátmodellezés** szorosan kapcsolódik az előntési modellhez. A gátrendszer térbeli elhelyezkedése és magassága befolyásolja az előntést. A gátszakaszok kijelölése után a gátrendszer egyes keresztmetszetei vizsgálhatók. A keresztmetszet a gát teherbírásának számításához biztosít adatokat. Az előntési- és a gátmodell kitűnő eszköznek bizonyul a völgyzáró gátak tervezésében is.

Ha nagyobb területet vizsgálunk, akkor lehetőség van egy forráspont helyett több eltérő magasságú forráspont elhelyezésére is. Ebben az esetben a *Elöntés* algoritmus kompetitív változatát kell alkalmaznunk. Ezzel a módszerrel egy teljes folyószakasz előntése és gátrendszere modellezhető. A folyószakaszon a folyó vízmagasságának mérési pontjaiban célszerű a forráspontokat elhelyezni.

Egy árhullám levonulása az árhullám teljes víztömegének mennyiségi modellezésével a következőkben ismertetett terjedési modellel lehetséges. A terjedési modell viszont jóval számítást igényel.

3.3.5. Terjedési modell

A **terjedési modell** lényege, hogy valamely térbeli eloszlású jelenség vagy esemény időbeli terjedését modellezzük. A terjedési modell segítségével erdőtűz-, levegő- és vízszennyezés, esetleg virágpor (*pollen*) terjedést tudunk modellezni. A terjedési modell hatékony megvalósítása a raszteres adatmodellel történhet. A raszteres adatmodellnél a térbeli kapcsolatok, a szomszédos pixelek felkeresése egyértelmű. A raszter elemi alkotóin, a pixeleken belül a bonyolult összefüggések viszonylag egyszerűen megfogalmazhatók. A terjedési modellre egy elméleti algoritmust dolgoztam ki, melyet gyakorlati később alkalmazások követhetnek.

A terjedés modellezésének első lépése a **források** elhelyezése. A források lehetnek véges vagy végtelen kibocsátásúak. A források helyzetét pixelekhez kötjük. Előfordul, hogy forrásokat nem helyezünk el, hanem egy konkrét szennyezési foltot mérünk fel, hogy annak terjedését vizsgáljuk meg.

A források elhelyezése után a következő feladat a **terjedési irányok és sebességek** megadása. Az irányok és sebességek együttes megadása egy táblázatban történik. A táblázatban az egyes főirányokhoz vagy adott szögterományokhoz hozzárendeljük a terjedési sebességeket. A táblázatok elkészülhetnek globálisan, a terület bizonyos régióira és pixelenként is. Bizonyos esetekben (*például folyómeder*) egy a raszterrel azonos méretű terjedési maszkot is definiálunk, amely a terjedési tartományt korlátozza.

A források, terjedési irányok, terjedési sebességek ismeretében a terjedési modell egy egyszerű, látványos, de számításigényes **iterációs eljárás**. Minden egyes iterációs lépésben az aktuális raszteres rétegből állítunk elő egy újat. A számításokat pixelenként kell elvégezni. Az előállítás figyelembe veszi az aktuális és a szomszédos pixelekben tárolt terjedési értékeket, a terjedési irányokat és sebességeket, valamint a terjedési maszkot. A szomszédos pixelek vizsgálata 3x3, 5x5, 7x7 stb. méretű ablakokra korlátozódik. Az új terjedési érték előállítása általában konvolúció, átlagolás, maximum vagy minimum keresés eredménye.

Erdőtűz esetében a forrás a tűz keletkezésének forrása, vagy egy konkrét tűzeset felmérése lehet. A terjedési érték a tűz károsításának mértékét tárolja. A terjedési irányt az uralkodó szélirány határozza meg. A terjedési sebességet a szélesebbesség, a domborzati viszonyok és a vegetáció befolyásolja. Erős szélben, lejtős területen, száraz vegetációban a tűz felgyorsul. A terjedési maszkban az erdei utakat, tűzpásztákat, vízfelületeket tüntetjük fel. A szakirodalom számos erdőtűz modellezéssel kapcsolatos munkát említ.

Vízszennyezésnél a forrás lehet egy pontszerű szennyező forrás, de itt is elképzelhető egy konkrét szennyezés felmérése. A terjedési érték ilyenkor a szennyezés koncentrációja. A terjedési irány és a sebesség folyónál egybeesik a folyó terjedésével. Tavaknál a szennyezés minden irányban terjedhet, talán az uralkodó szélirányban egy kicsit erősebb mértékben. A terjedési maszk egybeesik a folyómederrel, illetve a tómederrel. A vízszennyezési modellhez hasonlóan **árhullámot** is lehet modellezni. Ebben az esetben a terjedési érték nem koncentráció, hanem a vízoszlop magassága.

Légszennyezés esetén a források lehetnek például gyárkémények. A terjedési érték a szennyezés koncentrációja. A terjedési sebesség és irány az uralkodó szél függvénye. Terjedési maszkot általában nem adunk meg. A **pollenterjedés** nagyon hasonlít a légszennyezés modellezéséhez. Egyetlen eltérés, hogy a forrás egy nagyobb kiterjedésű vegetációs egység.

3.4. Középpontos vetítésű felvételek tájékozása

A távérzékelés egyre nagyobb szerephez jut a földrajzi adatgyűjtés területén. A raszteres adatok jelentős részét is távérzékelési módszerekkel állítják elő. A távérzékelte felvételek túlnyomó többsége **perspektív, középpontos vetítéssel** készül. Amennyiben a távérzékelte raszteres adatokat egy geoinformatikai rendszerben kívánjuk felhasználni, elengedhetetlen az adatok geometriai átalakítása, transzformációja. Az átalakítás célja, hogy a földrajzi adatokat **egységes vonatkozási rendszerben** tudjuk feldolgozni. A transzformáció első lépése középpontos vetítésű felvételek **tájékozása** vagyis a felvételi helyzet visszaállítása.

3.4.1. Képi koordináta rendszer

A felvétel tájékozása összetett feladat. Digitális felvételek esetén az első lépés egy megfelelő **képi koordináta rendszer** illesztése a raszteres adatokra. Az illesztés egy síktranszformáció felállítása, mely a raszter koordináta rendszeréből a kép koordináta rendszerébe transzformál. A transzformáció paramétereinek meghatározása mindkét rendszerben ismert pontok koordinátáinak megadásával történik. A közös pontok általában a laboratóriumi körülmények között bemért **keretjелеk**, melyek képi koordinátáit ismerjük. A közös pontok raszteres koordinátáit meghatározhatjuk manuálisan vagy automatikus módszerekkel. A **manuális** módszer a felvétel megfelelő nagyítása és eltolása után a keretjel képernyőn való megjelölését, vagyis keretjel raszter koordinátáinak mérését jelenti.

A **keretjелеk automatikus felkeresése** során a felvétel részleteit egy raszteres referencia adatbázis mintáihoz hasonlítjuk. A négyzetes minta kiválasztása a felvétel felbontásától és a felvevőrendszer típusától függ. Az összehasonlítást a keretjелеk lehetséges előfordulási helyére szűkíthetjük. Ez a teljes képterület alig néhány százaléka. Az összehasonlítás első felében az előfordulási területen belül a mintával azonos méretű maszkot mozgatunk. Az összehasonlítás második részében a **maszk és a minta közötti hasonlóságot** mérjük. A mérés történhet korreláció számítással vagy az eltérésnégyzetek átlagának meghatározásával (*Kraus 1997*):

$$r = \frac{n^2 \cdot \sum_{x=1}^n \sum_{y=1}^n a_{x,y} \cdot b_{x,y} - \sum_{x=1}^n \sum_{y=1}^n a_{x,y} \cdot \sum_{x=1}^n \sum_{y=1}^n b_{x,y}}{\sqrt{n^2 \cdot \sum_{x=1}^n \sum_{y=1}^n a_{x,y}^2 - \left(\sum_{x=1}^n \sum_{y=1}^n a_{x,y} \right)^2} \cdot \sqrt{n^2 \cdot \sum_{x=1}^n \sum_{y=1}^n b_{x,y}^2 - \left(\sum_{x=1}^n \sum_{y=1}^n b_{x,y} \right)^2}} \quad (3-3)$$

$$d = \frac{\sum_{x=1}^n \sum_{y=1}^n (a_{x,y} - b_{x,y})^2}{n^2} \quad (3-4)$$

ahol: r	korrelációs együttható, egy esetén teljes egyezés
d	eltérésnégyzetek átlaga, nulla esetén teljes egyezés
$a_{x,y}$	maszk x. oszlopában, y. sorában lévő szürkeségi érték
$b_{x,y}$	minta x. oszlopában, y. sorában lévő szürkeségi érték
n	a maszk és a minta mérete, a sorok és az oszlopok száma

Az összehasonlítások közül korreláció számítás során a maximumot, az eltérésnégyzetek átlagánál a minimumot kell megjegyezni. A **mintához legjobban hasonlító maszk** pozíciója

a keresett keretjel pozíciója. Ha nemcsak a legjobban hasonlító maszk, hanem környezetének hasonlóságát is tároljuk, akkor **pixel alatti pontosság** is elérhető. Ez egy kvadratikus felület illesztésével történhet. A másodfokú felület minimum, illetve maximum pontja már pixel alatti pontossággal határozható meg. Bizonyos felvevőrendszereken az egyes keretjelek képe eltér, amelyből a felvétel tájolása is meghatározható. Ezért a mintaadatbázisban minden keretjelnek külön adatsort kell létrehozni.

A képi koordináták és a raszter koordináták meghatározása után a két rendszer közötti transzformációt általában bilineáris (*mindkét változóra nézve elsőfokú tagok*) egyenletpárral oldjuk meg (Czimer 1997):

$$\begin{aligned}x'_i &= a_1x_i + a_2y_i + a_3x_iy_i + a_4 \\y'_i &= b_1x_i + b_2y_i + b_3x_iy_i + b_4\end{aligned}\tag{3-5}$$

ahol: x'_i, y'_i keretjel képi pozíciója
 x_i, y_i keretjel raszter pozíciója
 a_i, b_i a transzformáció együtthatói

Ha n számú keretjelet mértünk, akkor a transzformáció egyenleteit a következő mátrix egyenletekbe szervezhetjük:

$$\begin{aligned}A \cdot a = x' &= \begin{matrix} (n,4) & (4) & (n) \end{matrix} \begin{bmatrix} x_1 & y_1 & x_1y_1 & 1 \\ x_2 & y_2 & x_2y_2 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ x_n & y_n & x_ny_n & 1 \end{bmatrix} \cdot \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix} = \begin{bmatrix} x'_1 \\ x'_2 \\ \vdots \\ x'_n \end{bmatrix} \\ A \cdot b = y' &= \begin{matrix} (n,4) & (4) & (n) \end{matrix} \begin{bmatrix} x_1 & y_1 & x_1y_1 & 1 \\ x_2 & y_2 & x_2y_2 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ x_n & y_n & x_ny_n & 1 \end{bmatrix} \cdot \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix} = \begin{bmatrix} y'_1 \\ y'_2 \\ \vdots \\ y'_n \end{bmatrix}\end{aligned}\tag{3-6}$$

ahol: A a transzformáció egyenleteinek együttható mátrixa
 a, b a transzformáció együtthatóinak vektorai
 x', y' a képi koordináták vektorai

Az együtthatók meghatározása ezek után a legkisebb négyzetek módszerével történhet, ahol az $(A^T A)$ mátrix a normálegyenlet együttható mátrixa (Chester 1980):

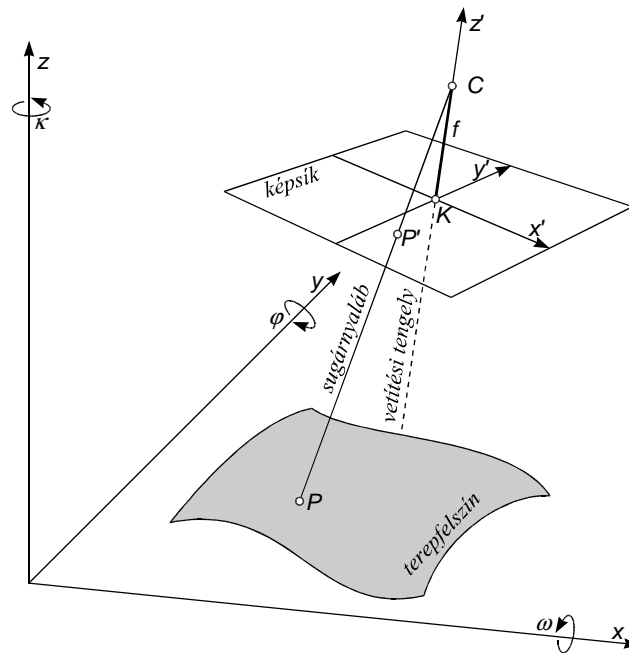
$$\begin{aligned}\begin{pmatrix} A^T \cdot A \\ (4,n) & (n,4) \end{pmatrix}^{-1} \begin{pmatrix} A^T \cdot x' \\ (4,n) & (n) \end{pmatrix} &= a \\ \begin{pmatrix} A^T \cdot A \\ (4,n) & (n,4) \end{pmatrix}^{-1} \begin{pmatrix} A^T \cdot y' \\ (4,n) & (n) \end{pmatrix} &= b\end{aligned}\tag{3-7}$$

3.4.2. A tájékozás alapegyenletei

A képi koordináta rendszer felvételhez illesztése után a **tájékozás sugárnyalábok segítségével** történik. A sugárnyalábok a felvevőrendszer leképezési középpontjából indulnak ki. Minden egyes térbeli ponthoz húzott sugárnyaláb pontosan egy képi pontot jelöl ki a

felvételen, viszont a felvétel egy képpontjához húzott sugárnyaláb végtelen sok térbeli pontot határoz meg. Egy térbeli pont és a hozzá tartozó képi pont között **perspektív kapcsolat** van.

Egy P térbeli pontot úgy képezhetünk le a képsíkra P' pontba, ha a pontot eltoljuk a leképezés C centrumába, majd térbeli forgatással elforgatjuk R forgatási mátrixszal a felvétel koordináta rendszerébe. A forgatás után a P ponton átmentő, a vetítési tengelyre merőleges tárgysík és a centrum közötti távolságot a transzformált pont z koordinátája hordozza. A leképezés utolsó lépése a z koordináta és a felvevőrendszer f fókusztávolságának segítségével a tárgysík vetítése a képsíkra (3-7. Ábra).



3-7. Ábra: Perspektív leképezés

A középpontos leképezés első részének, az eltolásnak és az elforgatásnak az egyenlete:

$$P'' = R \cdot (P - C) \quad (3-8)$$

$$\begin{bmatrix} p''_x \\ p''_y \\ p''_z \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \cdot \begin{bmatrix} p_x - c_x \\ p_y - c_y \\ p_z - c_z \end{bmatrix} = \begin{bmatrix} r_{11}(p_x - c_x) + r_{12}(p_y - c_y) + r_{13}(p_z - c_z) \\ r_{21}(p_x - c_x) + r_{22}(p_y - c_y) + r_{23}(p_z - c_z) \\ r_{31}(p_x - c_x) + r_{32}(p_y - c_y) + r_{33}(p_z - c_z) \end{bmatrix} \quad (3-9)$$

ahol: $P(p_x, p_y, p_z)$ a P pont térbeli pozíciója
 $P''(p''_x, p''_y, p''_z)$ az eltolt és elforgatott pont pozíciója a felvétel rendszerében
 $R(r_{11}, r_{12}, r_{13} \dots)$ a felvétel térbeli helyzetét leíró forgatási mátrix
 $C(p_x, p_y, p_z)$ a felvétel leképezési középpontjának pozíciója

Az egyenletben szereplő R forgatási mátrix a három térbeli tengely körüli forgatási mátrix szorzata. Az x tengely körül ω szöggel, az y tengely körül φ szöggel és z tengely körül κ szöggel forgatás mátrixai a következők (Chester 1980):

$$R_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \omega & -\sin \omega \\ 0 & \sin \omega & \cos \omega \end{bmatrix}, \quad R_y = \begin{bmatrix} \cos \varphi & 0 & \sin \varphi \\ 0 & 1 & 0 \\ -\sin \varphi & 0 & \cos \varphi \end{bmatrix}, \quad R_z = \begin{bmatrix} \cos \kappa & -\sin \kappa & 0 \\ \sin \kappa & \cos \kappa & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3-10)$$

Az x , y , z tengelyek körül rendre ω , φ , κ szöggel forgatás mátrixa az előbbi mátrixok szorzataként állítható elő:

$$R_{xyz} = R_x \cdot R_y \cdot R_z = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \quad (3-11)$$

$$R_{xyz} = \begin{bmatrix} \cos \varphi \cos \kappa & -\cos \varphi \sin \kappa & \sin \varphi \\ \cos \omega \sin \kappa + \sin \omega \sin \varphi \cos \kappa & \cos \omega \cos \kappa - \sin \omega \sin \varphi \sin \kappa & -\sin \omega \cos \varphi \\ \sin \omega \sin \kappa - \cos \omega \sin \varphi \cos \kappa & \sin \omega \cos \kappa + \cos \omega \sin \varphi \sin \kappa & \cos \omega \cos \varphi \end{bmatrix} \quad (3-12)$$

A perspektív leképezés a (3-9) egyenletből könnyen meghatározható. A képtávolság, a képsík és a centrum közötti távolság, a $-f$ fókusz távolság. A tárgy távolságot az eltolt és elforgatott pont p''_z koordinátája fejezi ki. Az x'' és y'' koordináták leképezése a képsíkra a képtávolság és a tárgy távolság hányadosával történik:

$$p'_x - k_x = -f \frac{p''_x}{p''_z} \quad (3-13)$$

$$p'_y - k_y = -f \frac{p''_y}{p''_z}$$

$$p'_x - k_x = -f \frac{r_{11}(p_x - c_x) + r_{12}(p_y - c_y) + r_{13}(p_z - c_z)}{r_{31}(p_x - c_x) + r_{32}(p_y - c_y) + r_{33}(p_z - c_z)} \quad (3-14)$$

$$p'_y - k_y = -f \frac{r_{21}(p_x - c_x) + r_{22}(p_y - c_y) + r_{23}(p_z - c_z)}{r_{31}(p_x - c_x) + r_{32}(p_y - c_y) + r_{33}(p_z - c_z)}$$

ahol: $P'(p'_x, p'_y)$ a képsíkra vetített pont koordinátái

$K(k_x, k_y)$ a képfőpont koordinátái

f kameraállandó, elméleti fókusz távolság

A (3-14) a középpontos leképezés alapegyenletei, amelyeket egy sugárnyaláb felállításával írtunk fel. Más szakirodalmak az egyenletpárt normál sztereogramnak vagy kollinearitási egyenleteknek nevezik. (A kollinearitás azt jelenti, hogy a P tárgypont, a P képpont és a perspektív centrum egy egyenesen van.) A képletben szereplő f kamera állandó és a K képfőpont pozíciója képezik a **belső tájékozási elemeket**. A **kameraállandó** a felvevőrendszer optikájának elméleti fókusz távolsága, a képsík és a leképezés középpontjának távolsága. A **képfőpont** a képsík és a vetítési tengely dőléspontja. A belső tájékozási elemeket általában laboratóriumi körülmények között határozzák meg. A leképezés C középpontjának három koordinátája és a térbeli forgatás ω , φ , κ szögei alkotják a **külső tájékozási elemeket**. Meghatározásuk illesztőpontok segítségével történik. Egy illesztőpont koordinátái mind a képi, mind a valós koordináta rendszerben ismertek. A (3-14) képletből látható, hogy egy

illesztőpontra egy egyenletpár írható fel, ezért a hat külső tájékozási elem meghatározásához minimum három illesztőpont szükséges.

3.4.3. Az alapegyenletek megoldása

A centrális leképezés alapegyenleteinek megoldása **Newton-féle linearizálással** lehetséges (Chester 1980). A lineáris formában felírható egyenletek iterációja, megfelelő előtájékozás után, kvadratikusan konvergál, és már néhány lépés után a kívánt eredményeket adja. Az **előtájékozás** feladata elsősorban a forgatási szögek közelítő megadása ± 90 fokos pontossággal. Légifelvételek esetén $\omega = \varphi = 0$, a κ szög pedig néhány illesztőpontból közelítőleg meghatározható. Az előtájékozás során a centrum koordinátáinak megadása az illesztőpontok súlypontja alapján történhet. A Newton-féle linearizálás rövid ismertetése a következő:

Az $f(x)=0$ függvényt x közelítő értéknél Taylor sorba fejtjük a magasabb rendű tagok elhagyásával. Az így kapott $f(x) + \delta f(x) = 0$ egyenletben a δ érték az egyenlet közelítő értékének javítását adja. A közelítő egyenlet többszörösen függvényekre is előállítható (Kraus 1998).

A Newton-féle linearizálás előnye a gyorsaságán kívül az, hogy semmilyen egyszerűsítést nem kell alkalmazni (például az analitikus fotogrammetriában használt $\sin(\alpha)=\alpha$, $\cos(\alpha)=1$, ha α kis érték) és közvetlenül a forgatási szögeket, koordinátákat kapjuk. További előny, hogy tetszőleges felvételi helyzetű (földi, légi, ferde, űr) képek tájékozására alkalmas. Az eljárás nullára rendezett két alapegyenlete a (3-14) képletből:

$$\begin{aligned} F &= (p'_x - k_x) \cdot (r_{31}(p_x - c_x) + r_{32}(p_y - c_y) + r_{33}(p_z - c_z)) + f \cdot (r_{11}(p_x - c_x) + r_{12}(p_y - c_y) + r_{13}(p_z - c_z)) \\ G &= (p'_y - k_y) \cdot (r_{31}(p_x - c_x) + r_{32}(p_y - c_y) + r_{33}(p_z - c_z)) + f \cdot (r_{21}(p_x - c_x) + r_{22}(p_y - c_y) + r_{23}(p_z - c_z)) \end{aligned} \quad (3-15)$$

A fenti két egyenletből parciális differenciálással állítható elő a linearizálás két alapegyenlete, ahol a d_i együtthatók rendre a külső tájékozási elemek javításai:

$$\begin{aligned} F + d_1 \frac{\partial F}{\partial \omega} + d_2 \frac{\partial F}{\partial \varphi} + d_3 \frac{\partial F}{\partial \kappa} + d_4 \frac{\partial F}{\partial c_x} + d_5 \frac{\partial F}{\partial c_y} + d_6 \frac{\partial F}{\partial c_z} &= 0 \\ G + d_1 \frac{\partial G}{\partial \omega} + d_2 \frac{\partial G}{\partial \varphi} + d_3 \frac{\partial G}{\partial \kappa} + d_4 \frac{\partial G}{\partial c_x} + d_5 \frac{\partial G}{\partial c_y} + d_6 \frac{\partial G}{\partial c_z} &= 0 \end{aligned} \quad (3-16)$$

Egy illesztőponthoz egy sugárnyaláb húzható, amelyből két differenciál egyenletet kapunk. A differenciál egyenlet hat együtthatójának meghatározásához minimum három illesztőpont, három sugárnyaláb, azaz hat egyenlet szükséges. Több differenciál egyenletet célszerű mátrix alakba rendezni, és mátrix formában megoldani, ahol n a sugárnyalábok, $2n$ az egyenletek száma:

$$A \cdot d = v = \begin{matrix} \begin{matrix} \frac{\partial F_1}{\partial \omega} & \frac{\partial F_1}{\partial \varphi} & \frac{\partial F_1}{\partial \kappa} & \frac{\partial F_1}{\partial c_x} & \frac{\partial F_1}{\partial c_y} & \frac{\partial F_1}{\partial c_z} \\ \frac{\partial G_1}{\partial \omega} & \frac{\partial G_1}{\partial \varphi} & \frac{\partial G_1}{\partial \kappa} & \frac{\partial G_1}{\partial c_x} & \frac{\partial G_1}{\partial c_y} & \frac{\partial G_1}{\partial c_z} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{\partial F_n}{\partial \omega} & \frac{\partial F_n}{\partial \varphi} & \frac{\partial F_n}{\partial \kappa} & \frac{\partial F_n}{\partial c_x} & \frac{\partial F_n}{\partial c_y} & \frac{\partial F_n}{\partial c_z} \\ \frac{\partial G_n}{\partial \omega} & \frac{\partial G_n}{\partial \varphi} & \frac{\partial G_n}{\partial \kappa} & \frac{\partial G_n}{\partial c_x} & \frac{\partial G_n}{\partial c_y} & \frac{\partial G_n}{\partial c_z} \\ \frac{\partial \omega}{\partial \omega} & \frac{\partial \varphi}{\partial \varphi} & \frac{\partial \kappa}{\partial \kappa} & \frac{\partial c_x}{\partial c_x} & \frac{\partial c_y}{\partial c_y} & \frac{\partial c_z}{\partial c_z} \end{matrix} \cdot \begin{matrix} d_1 \\ d_2 \\ d_3 \\ d_4 \\ d_5 \\ d_6 \end{matrix} = \begin{matrix} -F_1 \\ -G_1 \\ \vdots \\ -F_n \\ -G_n \end{matrix} \end{matrix} \quad (3-17)$$

Az egyenletrendszerből a d_i javítási együtthatók meghatározása kiegyenlítéssel, például a legkisebb négyzetek módszerével történhet. A kiegyenlítés a (3-7) képlethez hasonlóan a következő:

$$\left(\begin{matrix} A^T & A \\ (6,2n) & (2n,6) \end{matrix} \right)^{-1} \cdot \left(\begin{matrix} A^T & v \\ (6,2n) & (2n) \end{matrix} \right) = \begin{matrix} d \\ (6) \end{matrix} \quad (3-18)$$

DigiTerra szoftver környezetben a normál egyenletrendszert módosított Gauss eliminációval oldom meg. A módosítás speciális főelemkiválasztást és sebességre optimalizálást jelent. Az egyenletrendszer megoldása után a **külső tájékozási elemek javítása** így alakul:

$$\begin{aligned} \omega_1 &= \omega_0 + d_1 & \varphi_1 &= \varphi_0 + d_2 & \kappa_1 &= \kappa_0 + d_3 \\ c_{x1} &= c_{x0} + d_4 & c_{y1} &= c_{y0} + d_5 & c_{z1} &= c_{z0} + d_6 \end{aligned} \quad (3-19)$$

A javítások után a kiegyenlítést ismét el kell végezni. A sorozatos javításokat akkor kell leállítani, amikor a javítási együtthatók abszolút értéke egy előre megállapított ε értéknél kisebb. Ugyancsak le kell állítani az iterációt, ha a javítások nem konvergálnak. A külső tájékozási elemek meghatározása után az egyes pontok hibái a következő képlettel számíthatók:

$$\begin{aligned} e_x &= p'_x - k_x + f \frac{r_{11}(p_x - c_x) + r_{12}(p_y - c_y) + r_{13}(p_z - c_z)}{r_{31}(p_x - c_x) + r_{32}(p_y - c_y) + r_{33}(p_z - c_z)} \\ e_y &= p'_y - k_y + f \frac{r_{21}(p_x - c_x) + r_{22}(p_y - c_y) + r_{23}(p_z - c_z)}{r_{31}(p_x - c_x) + r_{32}(p_y - c_y) + r_{33}(p_z - c_z)} \end{aligned} \quad (3-20)$$

ahol: $E(e_x, e_y)$ hibavektor a képi koordináta rendszerben
 $P'(p'_x, p'_y)$ az illesztőpont képi koordinátái

A kapott hibák alapján az illesztőpontok pontosíthatók vagy kivehetők a tájékozásból. A felvételre rajzolt, az illesztőpontból kiinduló hibavektor sokat segít a tájékozás javításában. A Newton-féle linearizálás lehetőséget teremt a (3-15) képletben szereplő **bármely változó javítására**. A javítás kiterjedhet a belső tájékozási elemekre, az illesztőpont valós koordinátáira, de még a felvevőrendszer geometriai torzításainak csökkentésére is. A javítások a (3-16) egyenletek kiegészítését jelentik a megfelelő parciális deriváltak és további javítási együtthatók bevezetésével:

$$\text{Kameraállandó javítása: } F + d_7 \frac{\partial F}{\partial f}, \quad G + d_7 \frac{\partial G}{\partial f} \quad (3-21)$$

$$\text{Képfőpont javítása: } F + d_8 \frac{\partial F}{\partial k_x}, \quad G + d_9 \frac{\partial G}{\partial k_y} \quad (3-22)$$

$$\text{Illesztőpont javítása: } F + d_{10} \frac{\partial F}{\partial p_x} + d_{11} \frac{\partial F}{\partial p_y} + d_{12} \frac{\partial F}{\partial p_z}, \quad G + d_{10} \frac{\partial G}{\partial p_x} + d_{11} \frac{\partial G}{\partial p_y} + d_{12} \frac{\partial G}{\partial p_z} \quad (3-23)$$

Az utóbbi javítással eljutunk az illesztőpontból a **síkbeli illesztőpont, a magassági illesztőpont, illetve a kapcsolópont** vagy kötőpont fogalmához. Az illesztőpontnál mind a három koordinátát ismertnek tekintjük, síkbeli illesztőpont esetén a magasság ismeretlen, ellenben a magassági illesztőpontnál a síkbeli koordináták ismeretlenek, míg a kapcsolópont mindhárom koordinátáját ismeretlennek tekintjük. A síkbeli illesztőpontok alkalmazása a síkrajzi térképekről levett pontok esetén előnyös. Magassági illesztőpontok bevezetésével városok utcáin, terein, lapos tetejű épületein helyezhetünk el pontokat. A legnagyobb lehetőségeket viszont a kapcsolópontok kínálják, mikor egyszerre több képet akarunk tájékozni és ugyanazt az ismeretlen koordinátájú pontot kettő vagy több képen is megjelöljük.

3.4.4. Sugárnyaláb kiegyenlítés

Az előbbi gondolatvitellel eljutottunk a több felvétel együttes tájékozásához. Több, legalább 60% hosszirányú és 20% keresztirányú átfedéssel készült felvételt a fotogrammetriában felvételi tömbnek nevezünk. A tömb felvételeinek átfedő részein kapcsolópontokat hozhatunk létre. A kapcsolópontok perspektív kapcsolatot teremtenek az egyes felvételek között. Megfelelő számú kapcsolópont elhelyezése után elméletileg elegendő a felvételi tömb egészére mindössze három illesztőpontot megadnunk. A gyakorlatban, a megfelelő pontosság és az ellentmondások feloldása miatt a szükségesnél mindig több illesztőpontot és kapcsolópontot adunk meg. A tájékozás ismeretlenjeit ilyenkor kiegyenlítéssel határozzuk meg. Mivel a felvételek tájékozását sugárnyalábok kiegyenlítésével érjük el, ezért a módszert **sugárnyaláb kiegyenlítésnek** hívjuk. A szakirodalom számos tömbkiegyenlítési módszert említ. Ezek közül a legpontosabbnak és leghatékonyabbnak a sugárnyaláb kiegyenlítés tartja (Kraus 1997, Chester 1980). A sugárnyaláb kiegyenlítés általános egyenletei (3-14) képlet alapján egy i felvétel, egy j pont és egy s felvevőrendszer esetén a következők:

$$\begin{aligned} p'_{ijx} - k_{sx} &= -f_s \frac{r_{i11}(p_{jx} - c_{ix}) + r_{i12}(p_{jy} - c_{iy}) + r_{i13}(p_{jz} - c_{iz})}{r_{i31}(p_{jx} - c_{ix}) + r_{i32}(p_{jy} - c_{iy}) + r_{i33}(p_{jz} - c_{iz})} \\ p'_{ijy} - k_{sy} &= -f_s \frac{r_{i21}(p_{jx} - c_{ix}) + r_{i22}(p_{jy} - c_{iy}) + r_{i23}(p_{jz} - c_{iz})}{r_{i31}(p_{jx} - c_{ix}) + r_{i32}(p_{jy} - c_{iy}) + r_{i33}(p_{jz} - c_{iz})} \end{aligned} \quad (3-24)$$

Az egyenletekben szereplő R forgatási mátrix $r_{i11}, r_{i12} \dots$ tagjai az i felvétel külső tájékozási elemei közül az ω, φ, κ forgatási szögekből (3-12) képlet segítségével állíthatók elő. Az egyenletrendszerben most már több belső és külső tájékozási elem és pont meghatározás szerepel. Több belső tájékozási elem akkor fordul elő, ha a tömb felvételei eltérő kamerákkal készültek. A linearizálásban szereplő ismeretlenek száma a következő 3-2. Táblázat alapján számítható:

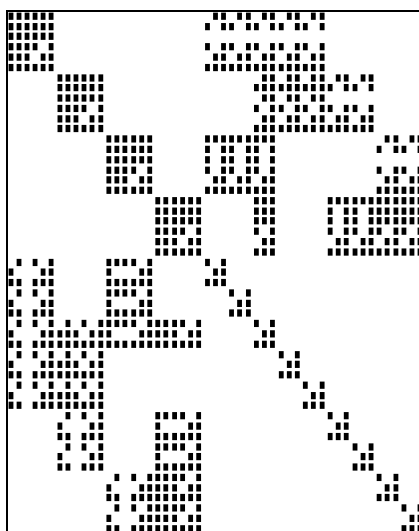
Tájékozási elem	Ismeretlenek száma
-----------------	--------------------

Belső tájékozás	3 · kamerák száma
Külső tájékozás	6 · felvételek száma
Síkbeli illesztőpont	1 · pontok száma
Magassági illesztőpont	2 · pontok száma
Kapcsolópont	3 · pontok száma

3-2. Táblázat: Az ismeretlenek száma a tájékozásban

Az előbbiekből látható, hogy egy illesztőpontra mérés egy sugárnyalábot állít fel, ami két egyenletet ad, ezért az ismeretlenek meghatározásához minimum feleannyi mérés szükséges. Ugyanakkor a nyalábok képenkénti eloszlása is fontos. Ha egy felvétel külső tájékozását akarjuk meghatározni, akkor a minimális feltétel, a három sugárnyaláb itt is követelmény.

Attól függően, hogy egy pontra történő méréskor mely elemeket tekintjük ismertnek, melyeket ismeretlennek és a pontra mely képekről történik mérés egy rendkívül összetett differenciál egyenletrendszer kapunk. Az egyenletrendszerben sokszor több ezer ismeretlen lehet. A normál egyenletrendszer együttható mátrixa 1-10%-ban feltöltött ritka mátrix (3-8. Ábra). A mátrix bal-felső és jobb-alsó része egy hiperdiagonál mátrix, melyet a számítások során figyelembe kell venni. Az első speciális szoftvertechnológiai módszer a mátrix tömörítése, és tömörített formában kezelése. Ez a tárterület optimális kihasználása mellett a számításokat is felgyorsítja. A főelem kiválasztás csak a főátló környezetében történik, a bázis transzformáció figyelni és átugorja az üres sorokat és oszlopokat, a C programkörnyezet és a mutatók használata gyorsítja a számításokat. A nagyfokú optimalizálásnak köszönhetően a *DigiTerra* sugárnyaláb kiegyenlítő modulja a jelenlegi PC-s környezetben pár másodperc alatt tájékoz egy 100 képből álló modellt.



3-8. Ábra: Két sorból és két oszlopból álló tömb normálegyenleteinek együttható mátrixa

A tömbkiegyenlítésnek számos további előnye van. Mivel a kiegyenlítés egyszerre történik, a hibák szisztematikusan tárhatók fel. A kiegyenlítés után egy lépésben indulhat az ortofotó készítés. Az ortofotó előállításakor a mozaikolás, az átfedések, a színegyensúly, a fényesség különbségek már készítés közben figyelembe vehetők. A kiegyenlítés előfeltétele a térkiértékelésnek, az automatikus képobjektum és felületmodell kinyerésnek.

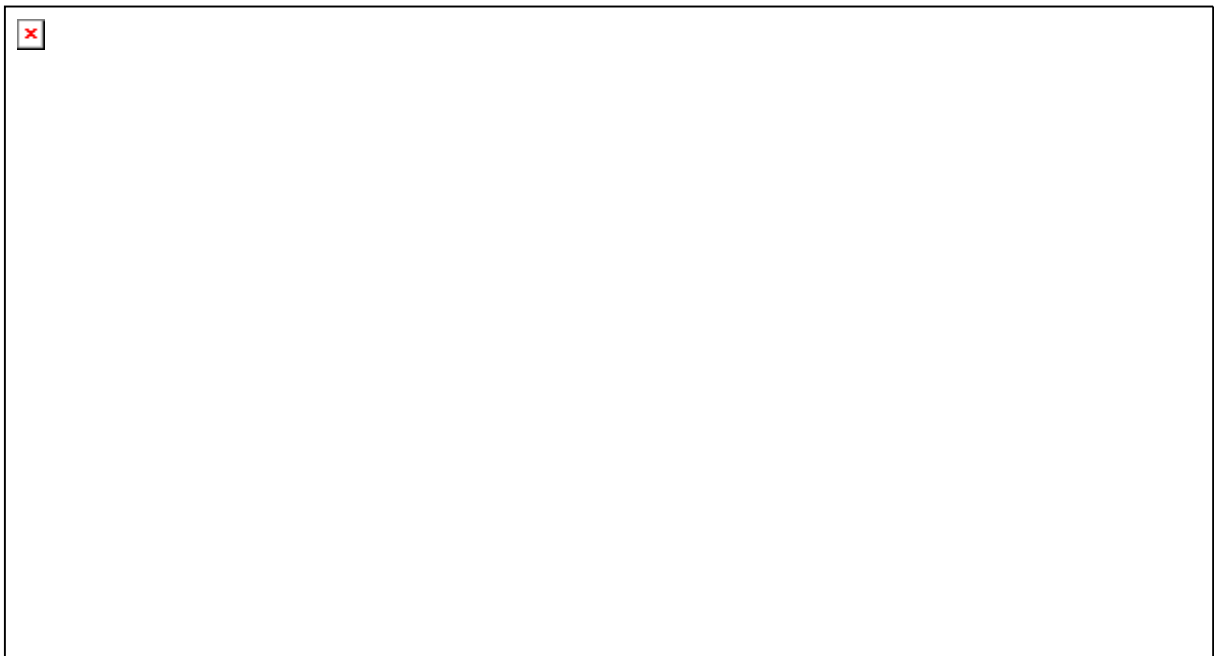
Tömbkiegyenlítés során gyakran 10-100 képpel kell dolgozni. Tekintettel a digitális képek méreteire (300MB-1GB) ez sok esetben hatalmas, nehezen mozgatható adathalmazt eredményez. A képeket ezért célszerű a tájékozási műveletek alatt tömörített formában kezelni (JPEG vagy Wavelet tömörítés).

3.4.5. Kapcsolópontok automatikus meghatározása

A kapcsolópontok automatikus felkeresése a felvételi tömb tájékozását nagyban felgyorsítja. Az automatikus felkeresés három részből áll. A feladat első részében **felvételpárokat** képezünk az előzetes tájékozási adatok alapján, a második részben digitális felvételek **jellegzetes képrészleteit kiemeljük**, végül a harmadik részben a különböző felvételekről származó képrészletek **összepakosítjuk**.

A felvételpárok képzése a számítások minimalizálása miatt fontos. Ha felvételenként 30-30 pontot emelünk ki, akkor a képrészlet összehasonlítások száma $30 \cdot 30 / 2 = 450$. Száz kép esetén a felvételpárok száma kevesebb, mint 100. Az összehasonlítások száma tehát 45000 körül alakul. Ha az előzetes tájékozási adatok nem állnak rendelkezésre, akkor a felvételenkénti 30 képrészletet minden más felvétel képrészletéhez hasonlítanunk kell. Száz kép esetén ez $30 \cdot 30 \cdot 100 / 2 = 4500000$ összehasonlítás. A számpéldából látható, hogy az előzetes tájékozás, a felvételek párosítása sokat gyorsít az eljárásán.

A szakirodalomban számos módszert találhatunk a jellegzetes képrészletek felkeresésére. A felkeresés **érdeklődési operátorok** segítségével történik (3-9. Ábra). Az operátorok először a digitális kép szürkeségi- vagy színértékeit módosítják egy fokális szűrővel. A fokális szűrők lehetnek élkiemelő (Sobel, Laplace) vagy élkiemelő és simító (Laplacian of Gaussian) vagy statisztikai (szórás, variancia) szűrők. A szűrőkben közös, hogy mindegyik a pixelértékekben jelentkező erős változásokat térképezik.



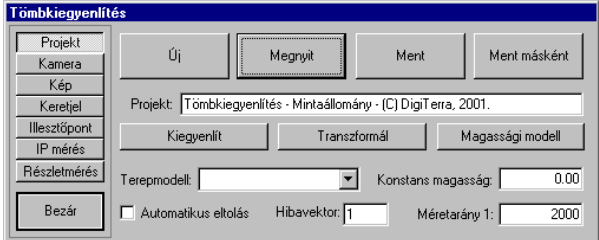
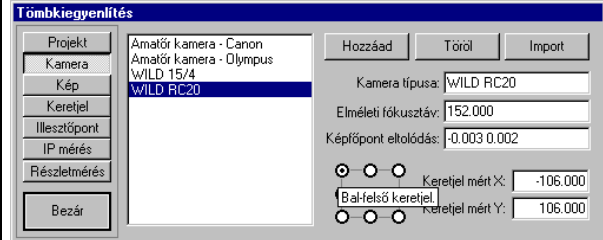
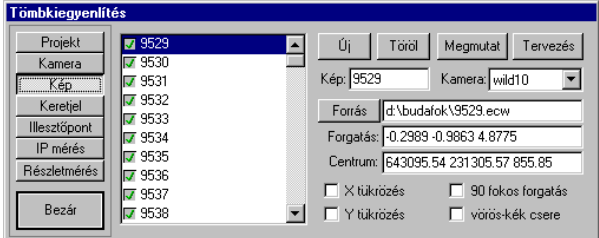
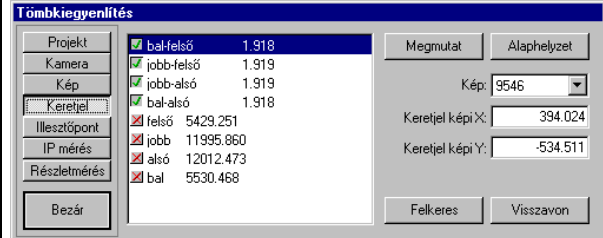
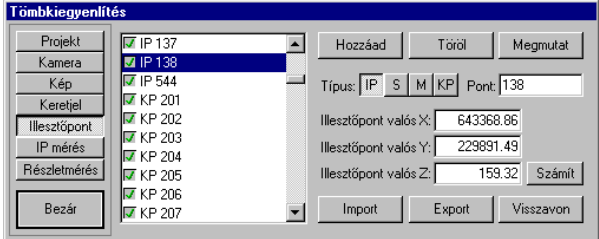
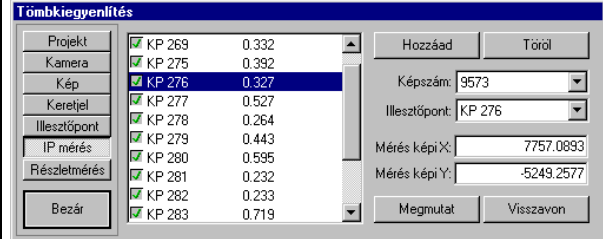
3-9. Ábra: Egy digitális légifelvétel és a felvételtől érdeklődési operátorokkal előállított két kép

Az operátorok a pixelek módosítása után a képrészlet méretével azonos ablakon belül számszerűsítik a képrészlet jellegzetességét, egyediségét. Ez általában a sorok és oszlopok szétválasztásából és a kovariancia mátrix felállításából áll. A kovariancia mátrixból kiolvasható a képrészlet változatossága és a képtartalom szabályos elhelyezkedése is (*Kraus 1997*). A képrészletek kiválasztását küszöbértékek felállításával lehet szabályozni.

A felkeresés harmadik részében a kiválasztott képrészleteket párosítjuk. A képrészletek közötti hasonlóság mérése a (3-3) és (3-4) összefüggések valamelyikével történhet. Mindkét számítás lehetőséget biztosít a pixel alatti pontosság elérésére. A képrészlet párokat a hasonlóság alapján sorba rendezzük. A kevésbé hasonló párokat elvetjük. Elvetjük továbbá azokat a párokat is, amelyek hasonlósága egy küszöbérték alatt van. A megfelelő párok kiválasztása után a kapcsolópontokra is felírhatjuk a centrális vetítés alapegyenleteit. Az utolsó szűrés a tömbkiegyenlítés után a pontra kapott hibák vizsgálatából áll. A (3-20) képletekkel meghatározott hibák alapján a kevésbé megbízható pontok kiszűrhetők.

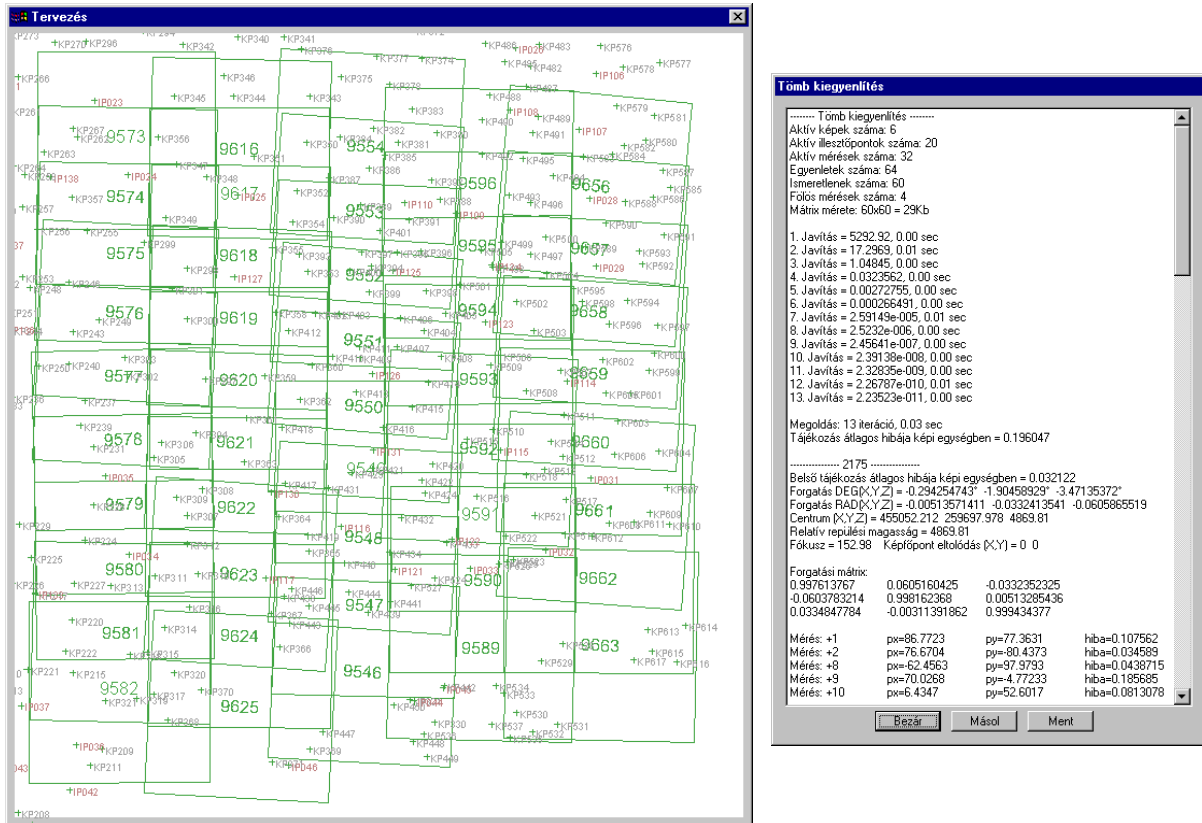
3.4.6. Tömbkiegyenlítés DigiTerra környezetben

DigiTerra Map programban a tömbkiegyenlítést egy több állapotú panel segítségével lehet végezni. A panel felépítése a tájékozási munka menetét követi (3-3. Táblázat). Az egyes munkafázisok a panel bal oldalán lévő gombsorral érhetők el.

	
<p>Tömbkiegyenlítés első panelje a kiegyenlítési projekt szervező funkcióit tárolja. Létrehozhatunk, elmenthetünk és visszatölthetünk tájékozásokat. Kiválaszthatjuk a tájékozáshoz használt terepmodellt, megadhatjuk a méretarányt és egyéb beállításokat.</p>	<p>A Kamera panelen létrehozhatunk új, vagy betölthetünk meglévő kamera beállításokat. Megadhatjuk az egyes kamerák elméleti fókusz távolságát, képfőpont eltolódását és keretjel koordinátáit.</p>
	
<p>A kép panel szolgál a tömb felvételeinek szervezésére. A kép kezelő funkciókon kívül a külső tájékozási elemek és a Tervező nézet indítása is itt kapott helyet.</p>	<p>A keretjel panelen a keretjelek megjelölésével lehet a képi koordinátarendszert illeszteni a képekre.</p>
	
<p>Illesztőpontok panelen lehet az illesztőpontokat, kapcsolópontokat létrehozni, importálni, exportálni és szerkeszteni.</p>	<p>Az IP mérés panelen történik a konkrét tájékozás. Ki kell választanunk a felvételt, a pontot és ezután létrehozhatjuk a mérést.</p>

3-3. Táblázat: *DigiTerra Map* tömbkiegyenlítés moduljának paneljei

A tömbkiegyenlítés során nagyon hasznos a tervező nézet és a kiegyenlítés menetét kijelző ablak (3-10. Ábra). A tervező nézeten betölthetjük a felvételi centrumok közelítő koordinátáit, grafikusan szerkeszthetjük a felvételek helyzetét, valamint elhelyezhetünk illesztő- és kapcsolópontokat. A kiegyenlítás menetét egy külön ablakban mutatja a program. Az ablakban minden kiegyenlítéssel kapcsolatos információ megjelenik. Az ablak tartalma felhasználható a hibák felderítésében és kinyomtatható, mint a tájékozásról készült jelentés.

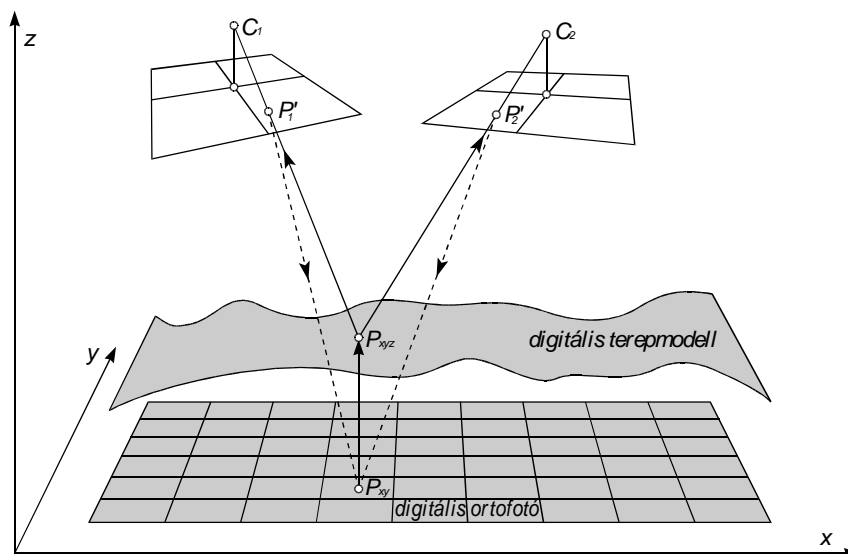


3-10. Ábra: Tervező nézet és a kiegyenlítés menetét mutató ablak

3.5. Ortofotó készítés

A digitális ortofotó előállítás a digitális fotogrammetriai szoftverek egyik legfontosabb funkciója. Az ortofotó készítés célja, hogy a középpontos vetítésű és tájékozott digitális felvételeket egy **geodéziai vetületbe transzformáljuk**. A transzformáció során középpontos vetítésről merőleges vetítésre térünk át. Az ortofotó készítést ezért szokás **képhelyesbítésnek** vagy **merőleges helyreállításnak** is hívni. Az analóg ortofotó készítés apró részterületek átfényképezéséből álló fáradságos művelet. A digitális változat a tájékozott digitális képek és az átalakítandó rész digitális felületmodelljének ismeretében egy könnyen megfogalmazható eljárás. Digitális ortofotót elsősorban légi- és űrfelvételekből állítunk elő.

A digitális ortofotó előállítás feltételezi a pontos tájékozást és egy részletes digitális felületmodell meglétét. Ha **valós ortofotóra** törekszünk, akkor a felületnek vissza kell adni az épületeket, a háztetőket, a terep törésvonalait és egyéb mesterséges alakzatait. A terepet és a tereptárgyakat leíró felületmodellt **borított felszínmodellnek** hívjuk. A borított felszínmodell egy lehetséges előállítási módja a későbbiekben ismertetett automatikus felületmodell kinyerési eljárás vagy a térkiértékelés. A borított felszínmodell készítése feltételezi a korszerű felületmodellezési eljárások alkalmazását. A *DigiTerra* program támogatja a raszteres, a szintvonal, az idomvonal, a törésvonal, a szabálytalan háromszögháló és a háromdimenziós test alapú modellezési eljárásokat. Mivel a borított felszínmodell előállítása munkaigényes, ezért sokszor megelégszünk a digitális **tereptmodell** alkalmazásával. Az így transzformált felvételt **hagyományos ortofotónak** nevezzük, amely a tereptmodellből kiemelkedő mesterséges alakzatok perspektív döléseit nem korrigálja.

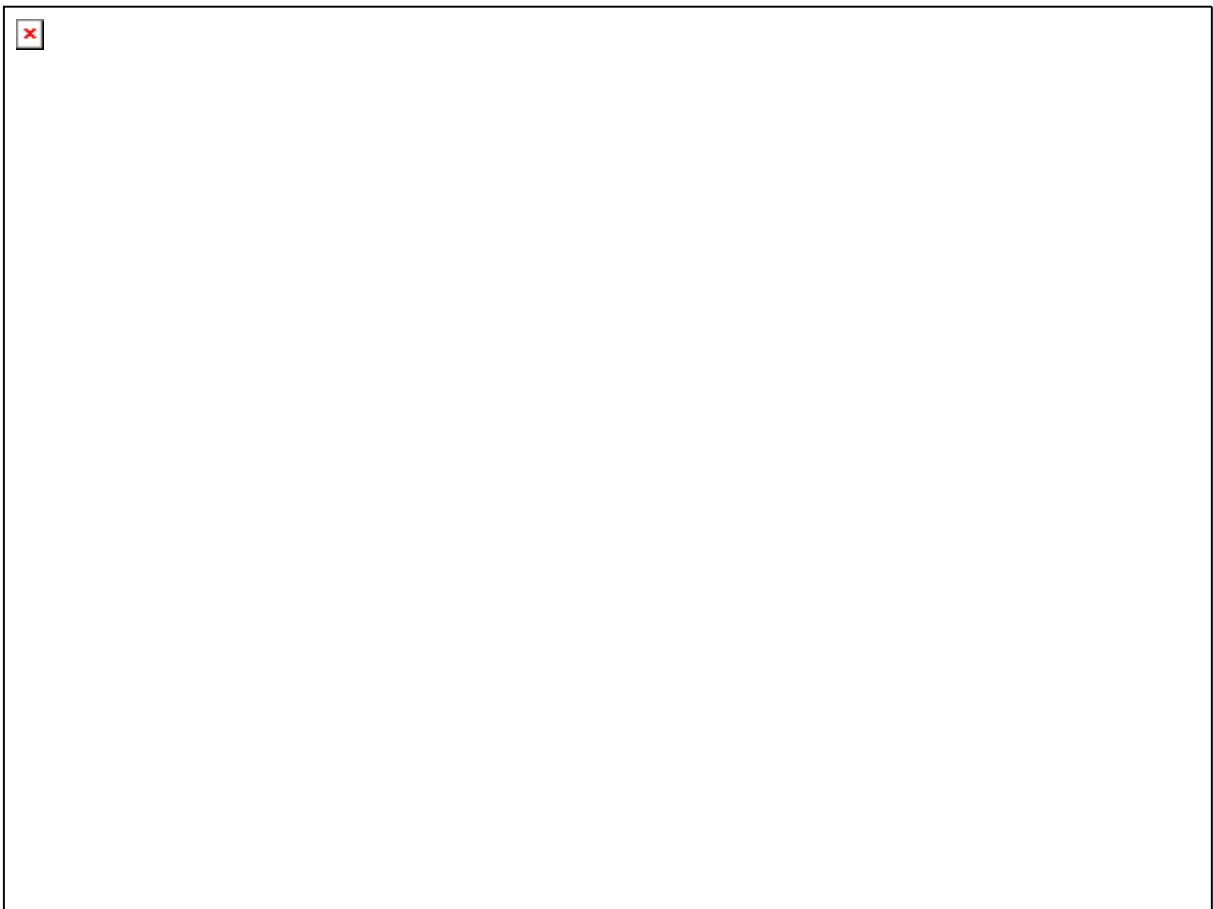


3-11. Ábra: Digitális ortofotó készítés két felvétel esetén

Az **ortofotó készítés** első lépése az eredmény vetületi rendszerben egy szabályos rácsháló definiálása (3-11. Ábra). A rácsháló befoglaló méreteit a felvételek által lefedett területhez vagy a vetület szelvényhálózatához igazítjuk. A rácsháló minden egyes képelemére egy térbeli pontot helyezünk. A pont magasságát a digitális felületmodell alapján határozzuk meg. Az így kapott térbeli pont és a tájékozott felvétel perspektív centruma között egy egyenest húzunk. Ahol az egyenes dőli a képsíkot, a kimetszett pixel szürkeségi- vagy színértékeit az ortofotó megfelelő képelemébe visszaírjuk.

A pixel kiválasztását **mintavételnek** nevezzük és megkülönböztetünk legközelebbi és bilineáris mintavételt, illetve kubikus konvolúciót. Legközelebbi mintavételnél azt a pixelt választjuk, amely a dőféspontot tartalmazza. Bilineáris mintavétel esetén egy elsőfokú felülettel súlyozzuk a szomszédos négy pixelt. A kubikus konvolúció a szomszédos tizenhat pixel súlyozását jelenti. A súlyokat a pixelközpontok dőfésponttól mért távolsága alapján két harmadfokú függvény szolgáltatja (*Erdas 1999*).

Az űrfelvételek és a kisméretarányú légifényképek esetében a Föld görbületével, az atmoszférával és a vonatkozási rendszer sajátosságaival is számolni kell. Ilyenkor célszerű a tájékozást, a korrekciókat és a számításokat egy **geocentrikus koordináta rendszerben** elvégezni. A sugárnyaláb kiegyenlítésbe a felvevőrendszer sajátosságai, az elrajzolási hibák is bevonhatók. Az eredmény ortofotó geometriai pontossága megfelelő tájékozás és jó felületmodell esetén egy pixel alatti (*3-12. Ábra*).



3-12. Ábra: Digitális ortofotó előtérben a digitális erdészeti térképpel

Ortofotó készítéskor képenként több száz megabájt adatot kell betölteni, transzformálni és kiírni. A felvételek teljes betöltése sok esetben nem lehetséges. A virtuális memória ilyenkor inkább lassít, mint gyorsít a feladaton. A megoldást a **felvételi blokkok** bevezetése biztosítja. A blokk a felvételt kisebb négyzetes részekre osztja. Egy blokk tárolásához elegendő pár megabájt, ezért az átalakítás már kevés központi memóriával is megoldható. A felvételi blokkok mellett egy **transzformációs blokkot** is bevezettem, mely az átalakítást hivatott felgyorsítani. A transzformációs blokk mérete kicsi 4x4 vagy 8x8 pixel, nagyságát a terepmodell felbontásához lehet igazítani. A számítási műveleteket elegendő a blokk

sarokpontjaira elvégezni. A számítás magában foglalja a magasság meghatározást, a dőféspont szerkesztést és a pixel kiválasztást. A köztes pontokra a sarokponti pozíciók interpolálásával kapjuk meg az eredeti kép pozícióit. Az egész aritmetikát használó interpoláció nagyságrenddel gyorsabb, mint a bonyolult számítási eljárás. Mérések alapján a transzformációs blokk 5...50-szeres gyorsulást eredményez. A gyorsításnak köszönhetően a transzformáció szűk keresztmetszete az adatok olvasása és írása lesz. A felvételi- és transzformációs blokkok bevezetésével a szakirodalomban eddig nem találkoztam.

Az ortofotó készítést általában a **felvételi tömb** valamennyi felvételén végrehajtjuk. Az együttes transzformációnak számos előnye van. Mivel a felvételek **átfednek**, ezért az átfedő részeken elegendő csak az egyik felvétel képelemét transzformálni. A felvételeken a kiemelkedő alakzatok a centrális vetítés miatt **kitakarának** bizonyos képrészleteket, amelyek viszont egy másik felvételtől láthatók. Az átfedés és a kitakarás gyorsítása érdekében létre kell hozni egy az ortofotó befoglaló méreteivel és felbontásával megegyező **kompetencia térképet**. Ez a térkép minden egyes eredmény pixelről megmondja, hogy azt melyik felvételtől kell kiválasztani. Az átfedési térkép a felvételek perspektív centrumainak közelségéből határozható meg. A közelség leggyorsabb eldöntése háromszögháló lefedés és *Thiessen*-poligonok készítésével történhet (4.7. fejezet). A kitakarás a láthatósági vizsgálattal határozható meg (3.3.2. fejezet).

Kifejlesztettem egy olyan algoritmust, mely az **átfedést és a kitakarást** egy menetben végzi el. Az eljárás megegyezik a láthatósági vizsgálattal, azzal a különbséggel, hogy a nézőpontok itt a perspektív centrumok és a láthatóság helyett a felvétel sorszámát és a centrum távolságát kell tárolni a pixelekben. A nem látható területek nulla értéket kapnak. Azok a pixelek, amelyek egy másik kép centrumához közelebb vannak, egyszerűen felülíródnak. Az eljárás utolsó részében egy képminőség javítás történik. Lényege, hogy az átfedési térkép határait a legközelebbi szürkeségi- vagy színérték változáshoz igazítja. A javítás után a két felvétel összeillesztése szinte észrevehetetlen. Az igazítás távolságát a felhasználó adja meg.

A felvételi tömb egyes képei általában eltérő kontraszt és fényerő viszonyokat mutatnak. Sok esetben a felvételen belül is találunk fényerő különbségeket (*tükröződés, vignettálás*). Az ortofotóval szemben támasztott egyik követelmény viszont az **egyenletes kontraszt és fényerő**. A felvételek együttes átalakítása lehetőséget nyújt a képminőség javítására a transzformáció közben. A felvételek kontraszt és fényerő viszonyai egy **színegyensúly térképpel** javíthatók. A térkép méretei az ortofotó befoglaló méreteivel megegyeznek, felbontása viszont helytakarékoság miatt kisebb. A színegyensúly térkép színcsatornánként tartalmazza a pixelcsoportok értékeinek átlagát és szórását. Az átlag segítségével a fényerő, a szórás felhasználásával a kontraszt állítható be. A fényerőt az átlagérték és a kívánt érték közötti különbséggel kell módosítani. A módosításnak van egy maximális értéke, amellyel megelőzhető, hogy a fehér területek sötétszürkévé váljanak. A kontrasztot a szórás alapján képzett szorzó módosítja. Itt is van egy küszöbérték, amely meggátolja a kontraszt jelentős elállítását. A színegyensúly térkép egy lokális kontraszt fokozó és lokális fényerő állító operátorként fogható fel.

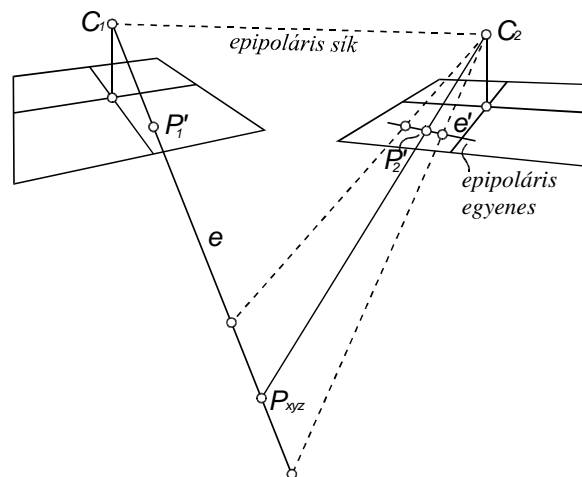
A kompetencia és színegyensúly térképpel együttesen kiejthető egy munkafázis, a transzformált felvételek mozaikolása. A **mozaikolás** az ortofotók egymáshoz illesztését, az egyes képek kontraszt és fényerő értékeinek módosítását jelenti. A mozaikolás egy köztes munkafázis, mely az eredetivel megegyező tárterületet és a transzformációval azonos feldolgozási időt igényel. Ezért kiejtése mindenképpen előnyös. A láthatósági térképet a

szakirodalom is említi, viszont kompetencia térképpel még nem találkoztam. A színegyensúly ortofotó készítés előtti beállításával kapcsolatos szakirodalmat sem ismerem. Az ortofotók elkészítése utáni mozaikolással viszont számos mű foglalkozik (*Kraus 1997, Burt 1984, Erdas 1999*).

A *DigiTerra* program a fent ismertetett funkciók túlnyomó részét támogatja. A felvételi tömb együttes transzformálása, a borított felszínmodell fogadása, a blokkonkénti átalakítás, a kontraszt és fényerő transzformáció közbeni módosítása, az eredmény ortofotó befoglaló méreteinek, felbontásának és fájl típusának beállítása mind elérhető. A kompetencia térkép és a színegyensúly térkép tesztelés alatt áll. A geocentrikus transzformáció, az atmoszférikus korrekció egy később verzióba fog beépülni.

3.6. Térbeli kiértékelés

Az ortofotó készítés mellett a digitális fotogrammetriai szoftverek másik fontos funkciója a térkiértékelés megvalósítása. A térkiértékeléshez legalább két tájékozott felvétel szükséges. A térkiértékelést is célszerű a tömbkiegyenlítéshez kapcsolni, ahol a két felvétel könnyedén kiválasztható. A felvételek között **perspektív kapcsolat** van. Az egyes felvételeken azonosított és összetartozó képpontokból kiszámíthatók a tárgypon térbeli koordinátái. A számítást a felvételek centrumaiból induló, a képpontokon keresztül húzott és a tárgypon tban metsződő két sugárnyaláb biztosítja. A két centrum, a két képpont és a tárgypon t egy síkon helyezkednek el. A síkot **epipoláris síknak** vagy magsíknak nevezzük. Az egyik kép centrumából a kiválasztott képponton keresztül húzott egyenest leképezhetjük a másik képsíkra. A leképezett egyenest **epipoláris egyenesnek** vagy magegyenesnek hívjuk. Az egyik képen kiválasztott képpont párja a másik képen az epipoláris egyenesen helyezkedik el. Az előbbieken röviden vázolt **epipoláris- vagy magsugár geometria** biztosítja a térkiértékeléshez szükséges számításokat (3-13. Ábra). A térkiértékelés vektoros eredményt szolgáltat: térbeli pontok, vonalláncok, hasábok, felületelemek.



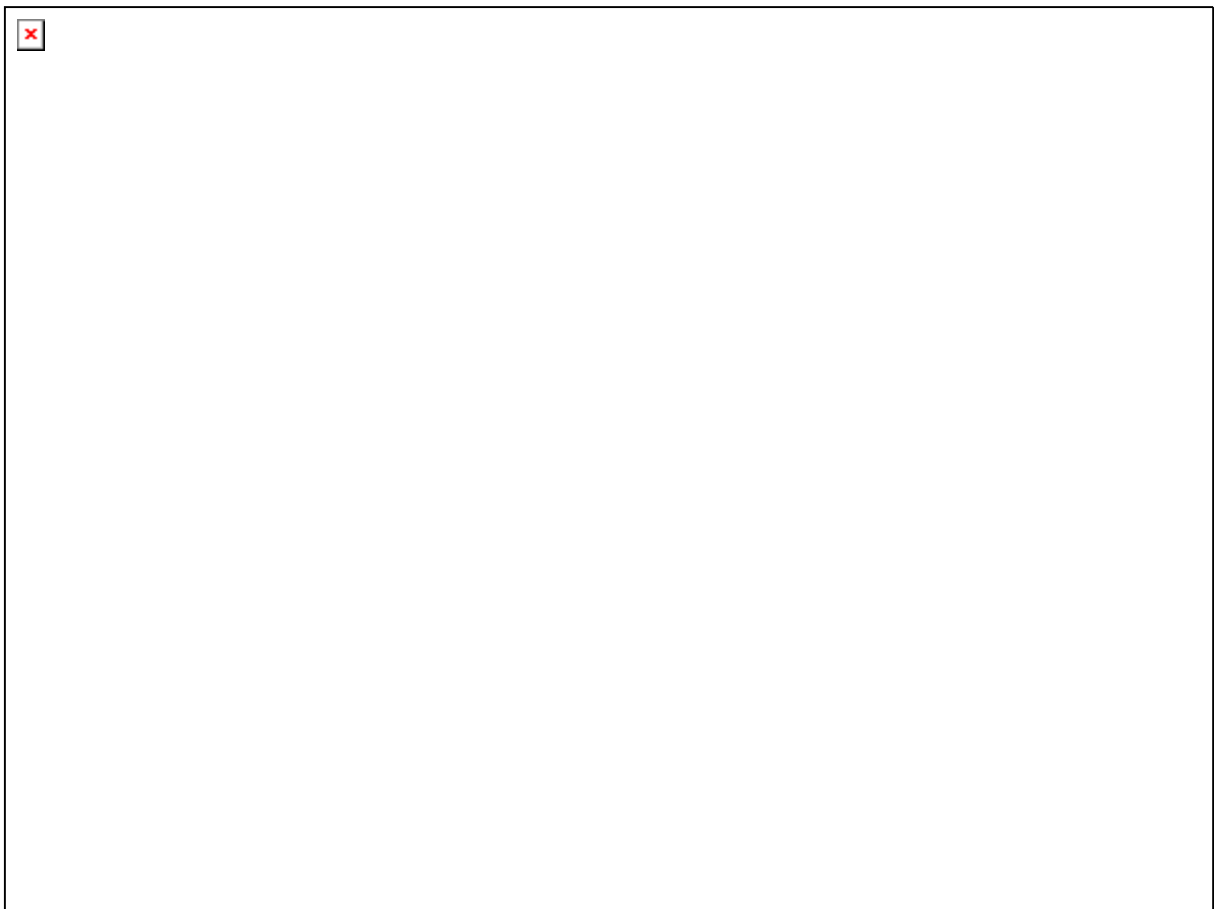
3-13. Ábra: Epipoláris geometria

A következőkben ismertetett két digitális térkiértékelési eljárás is az epipoláris geometrián alapul. Az első módszer bármilyen hardverkörnyezetben megvalósítható. Mivel a térbeli szemléltetés megvalósítása számítógépes környezetben még mindig körülményes, ezért ennél a módszernél a térbeli irányzást **két síkbeli irányzás** váltja fel. Az első síkbeli irányzás az egyik

képen a mérni kívánt képpont megjelölése. Ezután a program a másik képre kirajzolja az epipoláris egyenest, amelyen a megjelölt pont homológ párjának kell lenni.

A második irányzás az epipoláris geometriának köszönhetően már egydimenziós. A homológ pont felkeresése automatizálható, ami a kiértékelés munkáját nagyban gyorsítja. A felkeresés a megjelölt pont környezetében lévő pixelek és a másik képen az epipoláris egyenes melletti pixelek összehasonlításából áll. A hasonlóság mérése a (3-3) és (3-4) képletekkel történik. A legjobb hasonlóság mérési helyzete adja a homológ pont pozícióját, melyet a kiértékelő korigálhat. Ezt a módszert használja az 1996-ban készült *DigiTerra Stereo* program (Czímber 1997) és a 2002-ben a *DigiTerra Map* programba beépült térkiértékelő modul is (3-14. Ábra).

A második módszer a térlátáson, más szóval a **sztereoszkópián** alapul. A térlátás feltétele, hogy az eltérő pozícióból készített felvételeken csak bázisirányú parallaxisok legyenek. Az ilyen felvételek úgynevezett **normálképpárt** alkotnak, mely a kétképes kiértékelés normálesetének felel meg (Kraus 1998). A bázist a két felvételi centrum összekötésével kapjuk. A parallaxis a két képen eltérő módon leképeződött objektumok közötti eltérést jelenti. Tájékozott és átfedő felvételek esetén a normálképpár könnyen előállítható. A bázis meghatározása után a felvételeket úgy kell elforgatnunk középpontjaik körül, hogy a forgatás után felvételek x tengelye párhuzamos legyen a bázissal és az x tengely körüli forgatási szögük azonos legyen. A normálképek gyorsan előállíthatók, ezért elegendő az eredeti képeket tárolnunk és az átalakítást valós időben elvégeznünk.



3-14. Ábra: Térbeli kiértékelés a *DigiTerra Map* programban

Az előállított normálképek **sztereoszkópon** keresztül már térben szemlélhetők. A számítógép képernyőjén a térbeli szemlélés megvalósítható **anaglif** technikával is. Ilyenkor a felvételeket egymásra fektetjük, úgy hogy a vörös színcsatornához a bal kép, a zöld-kék színcsatornához pedig a jobb kép pixeleit rendeljük. A felvételeket anaglif (*vörös és zöldeskék szűrők*) szemüveggel kell szemlélni. Anaglif képek *DigiTerra Map* környezetben is előállíthatók.

A harmadik elterjedt, bár költséges módszer a **kitakarásos technika**. Itt a felvételeket másodpercenként legalább 25-ször váltogatjuk a képernyőn (*speciális grafikus kártyát, szoftvert és monitort igényel*). A kiértékelő szeme előtt egy folyadékkristályos szemüveg van, mely hol a bal, hol a jobb szem elől takarja el a képernyőt. A szemüveg és a képernyő szinkronban dolgoznak, ezért a bal szemmel a bal képet, a jobb szemmel a jobb képet látjuk. A gyors váltásokat és kitakarásokat az emberi szem nem érzékeli. A térbeli szemlélésnél az irányzás egy térbeli kurzorral történik. Az automatikus pontkeresés itt is megvalósítható. A normálképek miatt a korreláció vizsgálatot vízszintes sorok mentén kell végrehajtani.

3.7. Automatikus felületmodell előállítás

Az előbbi fejezetben ismertetett térbeli mérést, az irányzást és a homológ pont felkeresését egy digitális fotogrammetriai program automatikusan is végre tudja hajtani. A **mérés automatizálásával** a felvételek átfedő területeiről digitális felületmodell készíthető. Az automatikus felületmodell előállítás a digitális fotogrammetriai rendszerek legmagasabb szintű funkciója. Az eljárás nagy jelentőséggel bír az ortofotó készítésben azokon a területeken, ahol nincs domborzatmodell vagy az nem megfelelő pontosságú. Segítségével lehetővé válik a **valós ortofotó** elkészítése, ahol a képhelyesbítés nem a domborzat, hanem a borított felszín alapján történik.

Az automatikus felületmodell előállítás a homológ pontok, a képrészletek egyeztetése alapján csoportosítható. Az egyeztetése történhet a **képtérben, a normáltérben és az ortofotó térben** is. Ha a képtérben hasonlítunk össze, akkor képrészleteket össze kell forgatnunk az összehasonlítás előtt. A normáltérben a két normálkép között az összehasonlítás már vízszintes sorok mentén történik. Az ortofotó térben az egyeztetés a magassági síkokra elkészített ortofotók összehasonlításából áll. Mindhárom esetben a felvételek között perspektív kapcsolat van, amelyet az epipoláris geometria biztosít. A kapcsolat miatt a homológ pont felkeresése egydimenziós feladat. Az összehasonlítás, a képrészletek egyeztetése elvégezhető a **teljes területen** (*area-based matching*) vagy csak az érdeklődési operátorokkal kiemelt **jellegetes képrészletek** között (*feature-based matching*). Lehetőség van az érdeklődési operátorokkal kiemelt elemek térbeli **kapcsolatainak** összehasonlítására is (*relational matching*) (*Collins és mások 1995*).

A szakirodalomban a leggyakrabban a normáltér jellegetes képrészleteinek egyeztetésével találkozunk (*Kraus 1997, Baillard és mások 2000*). A gyakoriság nem véletlen, hisz ez az eljárás gyors és megvalósítása is viszonylag egyszerű. Nagyon kevés cikk található az ortofotó térben történő egyeztetésről (*Wiman 1998*). *DigiTerra* környezetben az ortofotó térben történő terület alapú egyeztetést valósítottam meg. A választás azért esett erre a módszerre, mert:

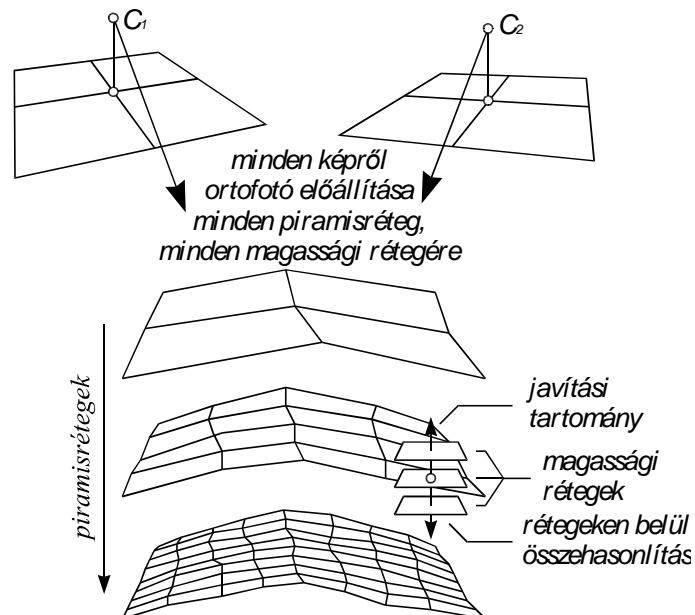
- nagyon sűrű felületmodellt eredményez,
- az összehasonlításba több kép is bevonható,
- a felvételi tömb egészén végrehajtható,
- a felületmodell mellett a digitális ortofotó is elkészül.

A választott módszer hátránya, hogy számításigényes és megvalósítása is bonyolult. A számítások megfelelő optimalizálásával viszont sikerült egy hatékony felületmodell kinyerési eljárást készítenem. A módszer hasonló a VLL (*Vertical Line Locus*) technikához (Kraus 1997), ahol szintén az ortofotó térben történik kis képrészletek előállítás és összehasonlítása. A módszerek közötti eltérés az, hogy a képrészletek helyett itt az egész kép szerepel.

Tekintettel a képek nagy méretére, elengedhetetlen volt az eredmény felületmodell részletekben történő megoldása, vagyis a **blokkosítás**. Természetesen a blokkokat a végeredményben össze kell illeszteni, amely a blokkok átfedésével biztosítható. A blokkok mérete általában 1024x1024 képelem.

Az egyes blokkok megoldása a blokkot átfedő forrásképek betöltésével indul. A megoldás több lépcsős, úgynevezett **piramistechnikát** alkalmaz. Az első lépésében egy kis felbontású sík felületből indulunk ki, amely a magassági javítás után a második rétegnek az alapját képezi. A második réteg felbontása az első kétszerese. A magassági javítást ezen a rétegen is elvégezzük. A rétegek felbontásának felezése és a magassági javítás mindaddig folytatódik, amíg a kívánt terepi felbontást el nem érjük (3-15. Ábra). Ezzel a technikával elkerülhető a hibás egyeztetések jelentős része.

Az egyes rétegek esetében a **magassági javítás** az aktuális felületmodell függőleges eltolásával kezdődik. Az eltolás pozitív és negatív irányban egyaránt négy-négy lépésben történhet. Az eltolás nagysága rendszerint megegyezik az aktuális piramisréteg terepi felbontásával. Az eltolás felületmodellekre minden egyes betöltött forrásképről elkészítjük az **ortofotókat**. Ezután a vizsgálat képelemenként folytatódik. A képelem új magassága az lesz, ahol az egyes ortofotó részletek között számított **korreláció** a legnagyobb. A korreláció számítás a (3-3) képlettel történik. Az ortofotó részletet általában egy 11x11 méretű ablak határozza meg.



3-15. Ábra: Automatikus felületmodell előállítás

Az ortofotó készítés és a korreláció számítás az eljárás leginkább **számításigényes része**. Az ortofotó készítés a transzformációs blokkok bevezetésével gyorsítható. A korreláció számítását pixeloszlopok előzetes összesítésével gyorsítottam. A 11x11 méretű ablakokra így nem kell a 121 értéket minden egyes pixel esetén összesíteni. Elegendő csupán egy oszlopösszeget levonni és hozzáadni az ablak statisztikáihoz (*összeg, négyzetösszeg, szorzatösszeg*). Ezzel a technikával a korrelációs ablak méretétől független számítási eljáráshoz jutunk.

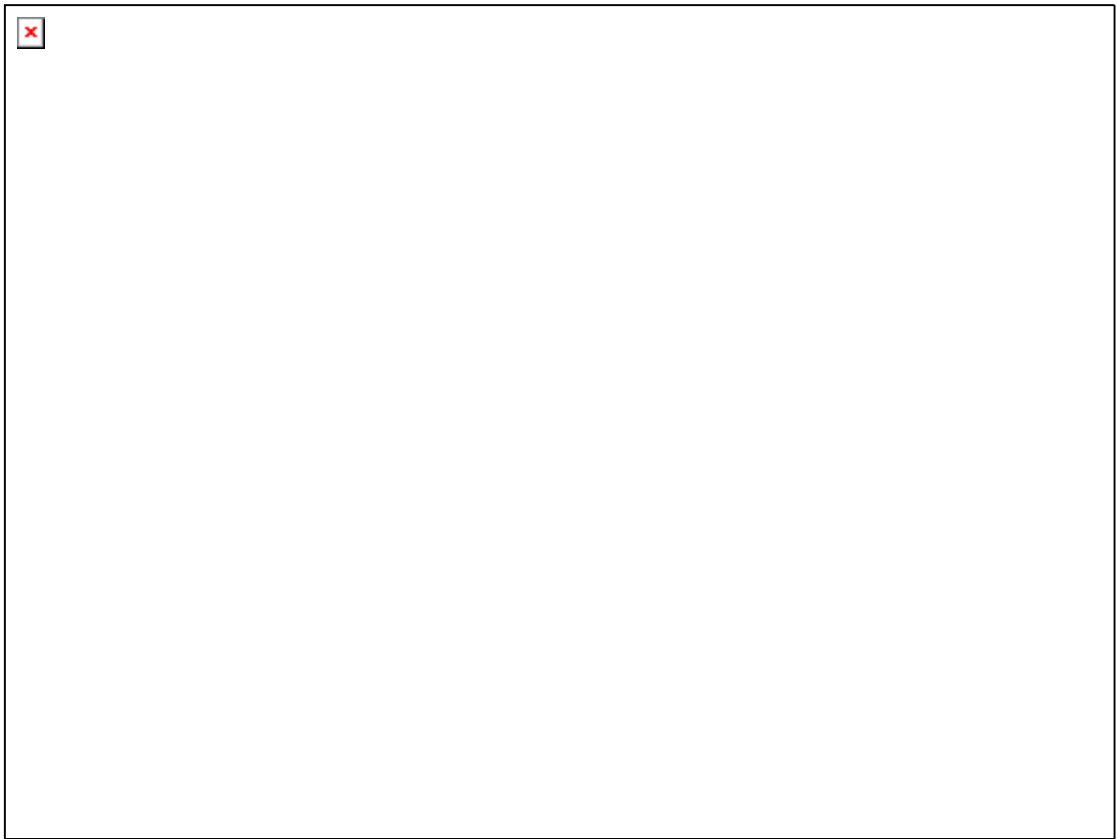
Az aktuális piramisréteg magassági javítása után a következő lépés a **simítás**. A simítás első feladata a gyenge korrelációval rendelkező területek törlése. Ezek a textúrában szegény, homogén részek (*vízfelület, erdő, szántóföld*) általában téves egyeztetéseket eredményeznek. A törlés után a gyors piramis interpolációval történik az üres területek magassági meghatározása. A simítás második feladata a tévesztések kiejtése közepesen vágó szűrőkkel. Az utolsó piramisrétegnél vagy beépített területek esetén ez a szűrő elhagyható. A simítás után az eljárás a következő réteg magassági javításával folytatódik. Ha elértük a kívánt terepi felbontást, akkor a blokk megoldottnak tekinthető.

Ha minden egyes blokk magassági javítását megoldottuk és a blokkokat összeillesztettük, akkor az eredmény felületmodell előállt. A módszernek köszönhetően a felületmodellel párhuzamosan elkészül a **korrelációs kép** és a **globális ortofotó** is, amelyeken az eljárás eredménye nyomon követhető. A gyenge korrelációval rendelkező területeken célszerű kiegészítő magassági méréseket végezni. Az általam készített magasság javító algoritmus ugyanis lehetővé teszi **meglévő pontok, vonalak és felületmodellek bevonását**. Ezzel a kiegészítéssel az automatikus eljárást könnyen a jó irányokba terelhetjük. Gyakoriak az olyan képterületek, amelyek a magasabb, kis felbontású rétegeken nem mutatnak részleteket. Az ilyen esetekben az egyeztetés többértelmű. Néhány magassági ponttal viszont áthidalható ez a sajátosság. Meglévő felületmodellek bevonásával a felületmodellek aktualizálhatók.

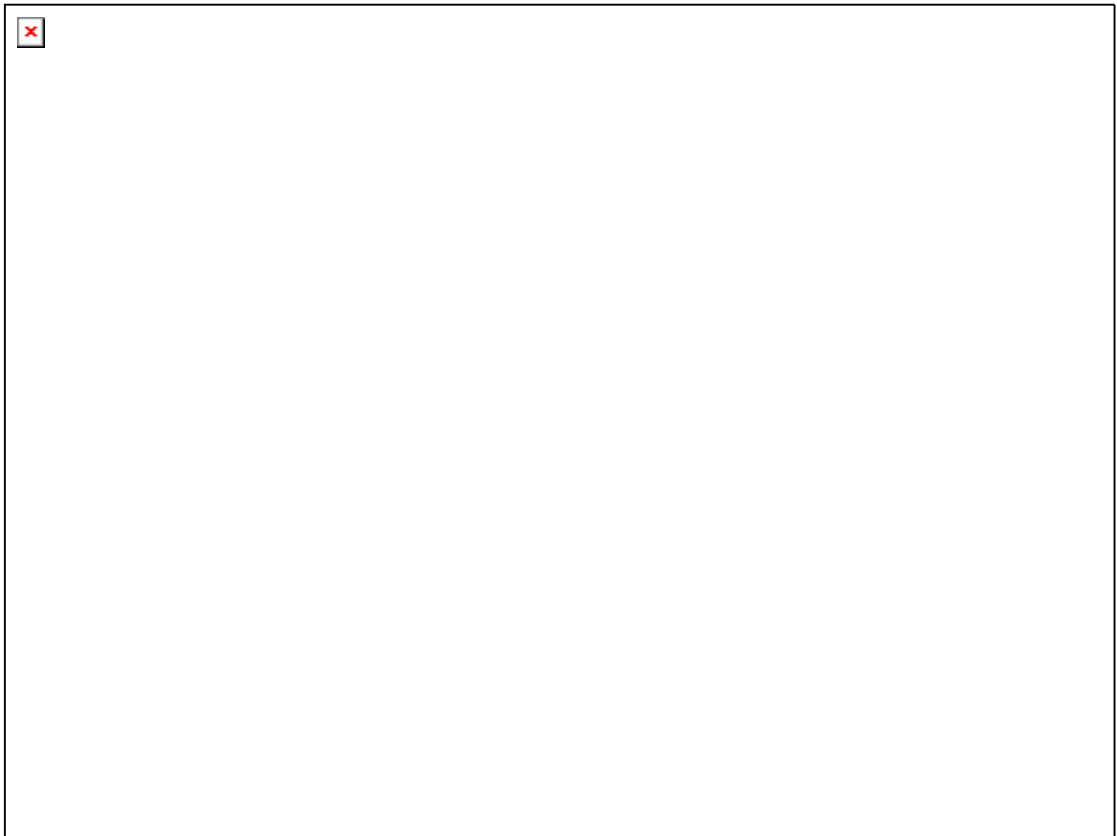
Eddigi tapasztalatok szerint a felületmodell **magassági pontossága** a képi tartalomtól függően a terepi felbontás 2..4-szeresében adható meg. Gyenge korreláció esetén a pontosság az előbbi érték többszöröse is lehet. Az eljárás segítségével a jelenlegi számítógépeken egy percen belül készíthető el a digitális felületmodell egy felvételpár átfedő részéről.

Az automatikusan kinyert borított felszín modelljéből átlagoló és minimumszűrők alkalmazásával levezethető a **domborzat digitális modellje**. A domborzatmodell lehetővé teszi lejtőkategória-, kitettség térképek előállítását, hidrológiai elemzések készítését. Az eljárás számos további lehetőséget rejt: automatikus mozaikolás és színegyensúly számítás, 3D városmodellek, famagasság mérés, földi- és ipari fotogrammetria stb.

Az automatikus felületmodell kinyerés funkció 2001-ben épült be a *DigiTerra Map* programba (3-16. *Ábra* és 3-17. *Ábra*). Az eljárásban a blokkosítás, a blokk egészének magassági javítása, az ortofotó előállítás és a korreláció számítás gyorsítása, a gyenge korrelációjú területek kizárása, a meglévő mérések bevonása és a piramis interpoláció tekinthető új megoldásnak (Czímber 2001).



3-16. *Ábra: Két tájékozott felvétel átfedő részéről automatikusan előállított felületmodell*

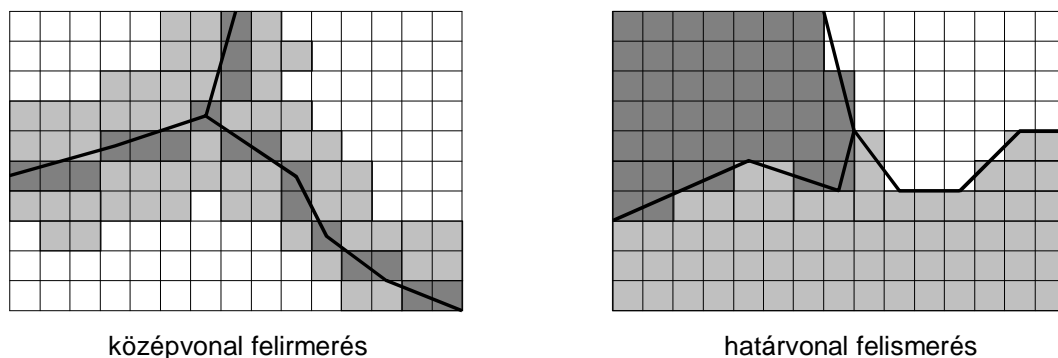


3-17. *Ábra: Az automatikusan kinyert felületmodellre húzott digitális ortofotó*

3.8. Raszter-vektor átalakítás

A fejezet a raszteres adatok vektorosra alakításáról szól és egyben megteremteni az átmenetet a dolgozatban szereplő két eltérő adatmodell, a raszteres és a vektoros között. A raszter-vektor átalakítás során a raszteres képekből, szkennelt térképekből, digitális felvételekből vektoros adatokat állítunk elő. Az eljárást röviden **vektorizálásnak** nevezzük. A vektorizálás jó minőségű, kevés feliratot, színezést tartalmazó térképek, például alap-, felmérési- és szintvonalas térképek, osztályozott úrfelvételek esetében hatékony. Az eljárás teljesen automatizálható, de egy strukturálatlan vektoros eredményt szolgáltat, ami további manuális feldolgozást, javítást igényel. Előfordulhat, hogy a vektorizálás és az utómunkálatok, a vektoros elemek javítása, osztályokba sorolása, a felesleges elemek kiszűrése több időbe kerül, mint a manuális digitalizálás. Bár bizonyos rendszerek képesek a vonal vastagsága, típusa alapján osztályozni a vektoros elemeket és a feliratokat is felismerik, de ez sem nyújt tökéletes megoldást a manuálisan szerkesztett, egymást átfedő térképi objektumok vektorizálásában.

A vektorizálás a digitális képanyag előállításával kezdődik. A következő lépés a képek tájékozása a digitális légifelvételek belső tájékozásához hasonlóan. A különbség csak annyi, hogy itt nem egy képi koordináta-rendszert, hanem egy geodéziai koordináta-rendszert kell illeszteni mindkét rendszerben ismert pontok segítségével. A tájékozás után indulhat a vektorizálás, melynek célja **vonalak vagy területek felismerése** (3-18. Ábra).



3-18. Ábra: A raszter-vektor átalakítás típusai

A raszteres **vonalak vektorizálása** egy vékonyító algoritmussal indul. Az eljárás végeredménye egy olyan raszterkép, amely 1 pixel vastagságú raszteres vonalakat, a középvonalakat tartalmazza. Az eljárást ezért szokás **középvonalas vektorizálásnak** is nevezni. A vékonyítás a szomszéd pixelek vizsgálatán alapul. Egy adott pixel nyolc szomszédjának ki/bekapcsolt állapota alapján $2^8=256$ variáció fordulhat elő. A vékonyítás a 256 variációra felállított döntési táblázat alapján történik. A táblázat elkészítésére a *Zhang és Suen (1984)* algoritmust használok bizonyos módosításokkal. Az algoritmus a bekapcsolt szomszédos pixelek száma és a sugarasan egymást követő pixelek értékváltozásának száma alapján osztályoz. A módosítás az egy pixelnyi kinyúlások, az átlós és a lépcsős formák kiszűrését jelenti. A táblázat alapján minden pixelről eldönthető, hogy kikapcsoljuk vagy változatlanul hagyjuk. Az iteratív eljárást addig kell futtatni az egész képen, amíg volt kikapcsolás. A kikapcsolást úgy gyorsítottam fel, hogy a következő vizsgálatot csak az előző kikapcsolások közvetlen közelében végzem el.

Az elvékonyított raszteren soronként kell az egyes **vonalakat felépíteni**. Az előző sor és az aktuális sor vizsgálata alapján a bekapcsolt pixelek középpontja között egy vonalszakasz húzható. A szakaszokat az azonos végpontok alapján vonalláncokká tudjuk összefűzni. A soronkénti felépítés előnye, hogy topológikus vektoros állomány építhető. Az állományban a vonalak nem metszik egymást, csak a csomópontokban találkoznak. A vonalszakaszok gyors felépítéséhez a pixelek szabályos elhelyezkedését, mint térbeli indexet használom ki.

A nyers vonalhalmaz után számos **utófeldolgozó művelet** következhet. Ezek közül a legfontosabb a **vonalak generalizálása**. A generalizáláshoz az általam fejlesztett iteratív algoritmus tűnt a leghatékonyabbnak. Az algoritmus lényege, hogy a vonallánc adott töréspontját akkor törölöm, ha a pont előtti és utáni töréspont között húzott egyenes a ponttól egy meghatározott távolságon belül van. A vizsgálatot minden második töréspontra el kell végezni. A vonallánc iteratív vizsgálata addig folytatódik, amíg volt ponttörlés. Másik két egyszerű utófeldolgozó módszer a rövid és elkülönülő **vonalláncok törlése**, illetve a vonalláncok **összekapcsolása** a közeli végpontok alapján. Előbbivel a térképen lévő nem kívánt „*maszatok*”, utóbbival pedig a szakadások szűrhetők ki, a szaggatott vonalak kapcsolhatók össze. A vonalláncok görbülete, töréspontjainak relatív helyzete alapján ívek és feliratok is felismerhetők.

A **foltok vektorizálása** osztályozott úrfelvételeken, színes, tematikus raszterképeken történhet. A szakirodalom (*Hoover és mások 1995*) rendszerint a határok felismerését említi. Ha két pixel színértéke vagy szűrkeségi értéke közötti különbség egy megadott számnál nagyobb, akkor a két pixel között egy határvonal húzható. A módszer hátránya, hogy a határvonalak sok esetben nem folytonosak, főleg árnyalatos képeken.

Az általam javasolt módszer a képfeldolgozásban ismert **területnövesztő szegmentálást** használja. Az eljárás elején minden egyes pixel egy elemi szegmenst alkot. Az iteratív területnövesztés során a szegmenseket összekapcsoljuk, ha a pixelek digitális értékeinek szórása egy megadott értéknél kisebb. Amennyiben nincs több összekapcsolás, akkor az eljárás befejeződik. Ezután következhet a kialakult szegmensek vektorizálása. A vektorizálás történhet a szegmensek körbejárásával vagy soronként a határvonalak felismerésével és összekapcsolásával. Előbbi egy poligon, míg utóbbi egy vonalas állományt eredményez. A határvonalak felismeréséhez középvonalas vektorizálásnál tárgyalt algoritmus és vonallánc generalizálás használható.

A vektorizálási eljárásban a saját tudományos eredményt a *Zhang és Suen* algoritmus módosítása, a topológikus vonal építés, az iteratív generalizálási algoritmus, az utófeldolgozás és a szegmentálás alkalmazása jelenti.

4. Vektoros eljárások

A földrajzi objektumok jellemzése geometriai és leíró adatokkal történik. A raszteres adatmodellben a szabályos rács és a georeferencia adatok hordozzák a geometriát, míg a rasztert felépítő pixelek értékei a leíró adatokat. A másik elterjedt adatmodellben, a vektorosban, a földrajzi objektumokat **szabálytalan geometriai elemek és a hozzájuk kapcsolt tulajdonságok** írják le. A geometriai elemek jellemző pontjait koordináták, vektorok határozzák meg. Innen származik az adatmodell elnevezése is: vektoros adatmodell.

Korábbi geoinformatikai rendszerek a vektoros és leíró adatokat külön adatállományban helyezték el. Az adatok összekapcsolásáról a rendszernek kellett gondoskodni. Az adatbázis-kezelő rendszerek fejlődésével azonban elérhetővé vált a **vektoros adatok és tulajdonságok egy rekordon belüli tárolása**. A korszerű adatbázisokban lehetőség van ugyanazon rekordon belül több geometriát is tárolni, például az objektumot jellemző geometrián kívül elhelyezhető az objektum középpontja, felirat elhelyezési pontja, befoglaló téglalapja, az objektum köré írható kör is. Az OpenGIS konzorcium számos utalást tesz a geometriai adatbázisok kialakítására (*OpenGIS 1998*).

A vektoros adatmodellről általánosan elmondható, hogy az adatok **előállításuk munkaiigényes**. Az adatok leggyakrabban geodéziai felmérésekből, GPS mérésekből, fotogrammetriai kiértékelésekből, analóg vagy raszteres térképek digitalizálásából származnak. Gyakori művelet a vektoros adatok **karbantartása és aktualizálása**. Mivel a vektoros adatok lényegi információt hordoznak, ezért ugyanarról a területről készített vektoros térkép mérete töredéke lehet raszteres párjának. A lényegi információ rendszerint időben állandó földrajzi objektumok leírását jelenti, ezért a vektoros adatok **elévülése sokkal hosszabb**, mint például egy raszteres úrfelvételé.

A vektoros adatmodellben külön kell gondoskodni a gyors térbeli lekérdezéseket biztosító **térbeli indexelésről** és az elemzésekhez szükséges **térbeli kapcsolatok kialakításáról**. Maguk az elemző eljárások bonyolult algoritmusok. Mind az algoritmusok megvalósítása, mind azok futásideje hosszabb időt vesz igénybe, mint raszteres párjaik. Vannak viszont kifejezetten vektoros adatmodellre készített eljárások, például az úthálózat elemzés. Ezek az algoritmusok mind a topológiát használják ki.

A vektoros adatok tárolására sok megoldás született. A leginkább elterjedt adattárolás (*ESRI 1998*), amelyet *DigiTerra* környezetben is használunk a következő:

- geometria típusa (*enumerációs érték*),
- geometriai objektumot felépítő elemek száma (*numerikus érték*),
- töréspontok száma elemenként (*numerikus lista, dinamikus bővítés*),
- töréspontok (*pontlista, dinamikus bővítés*).

A geometria **típusa** rendszerint pont, vonal vagy terület lehet. A görbe elemek jól közelíthetők vonalláncokkal, ha a töréspontok számát növeljük. Ebben az esetben elegendő három típusú eljárást fejleszteni. Ez a három csoport a pont-pont, pont-vonal és vonal-vonal kapcsolat alapján különül el. Bizonyos rendszerek támogatják a körök, ellipszisek, ívek és görbék ábrázolását is, de ez minden geometriai elem párkapcsolatra külön eljárás kifejlesztését igényli. Van még két érv, ami a vonalláncok mellett szól (*az ESRI cég a vonalláncokra teljes szoftvercsaládot épített fel*). A vonalláncok a terepen mért vagy a térképen digitalizált

pontokat kötnék össze. A pontok, mint geometriai elemek koordinátái lekérdezhetők és a terepen felkereshetők, kitűzhetők. A másik érv, hogy minden egyes vektoros térképnek van egy **névleges méretaránya**, amelyet az adatforrás, a felhasználás határoz meg. A térkép geometriai pontossága az adatforrástól és az adatelőállítás körülményeitől függ. A geometriai adatok ábrázolási pontosságát viszont a felhasználás befolyásolja.

Összetett geometriai objektumok leírása a geometriai elemek többszörözésével lehetséges. Az előbbi négy jellemzővel leírható összetett elem például a ponthalmaz, a szakadásokkal rendelkező vonallánc, a több poligonból felépülő régió, vagy a belső szigetekkel rendelkező poligon. Az egyes elemek töréspontjainak számát külön kell tárolni, viszont a töréspontok tárolhatók egy listában is.

A **töréspontok típusának** meghatározása is egy fontos kérdés. Bizonyos vektoros térképek esetén az adatábrázolás csak centiméteres pontosságot igényel és térbeli koordináták tárolása sem szükséges. Ezért *DigiTerra* környezetben négyféle töréspontot használunk. Az egyszeres pontosságú pontok koordinátáit nyolc, míg a dupla pontosságú pontokét tizenhat decimális számjeggyel ábrázoljuk. A síkbeli töréspontok két, a térbeli töréspontok három koordinátát tárolnak. Az egyszeres pontosságú síkbeli töréspont tárolása 8 bájtot, a dupla pontosságú térbeli töréspont tárolása 24 bájtot igényel. Ennek köszönhető, hogy *DigiTerra* adatállományok más formátumok (*ESRI 1998*) méretének 30-50%-a.

A térbeli koordinátákkal ábrázolt összetett objektumokkal leképezhetők bonyolult **térbeli objektumok** is, például épületek, tereptárgyak, élőlények, növények modelljei. Ezeknél az objektumoknál a geometriai elemek térbeli felületeket jelentenek. A felületeket előre elkészített textúrákkal boríthatjuk be.

A vektoros adatmodell **alkalmazása** mindenütt előnyös, ahol a cél a jellegzetes földrajzi objektumok ábrázolása, időtálló adminisztratív és tematikus térképek létrehozása, geodéziai mérések feldolgozása. Az adatmodell használata számos adminisztrációs, földhivatali, önkormányzati, és erdészeti rendszerben sikeres.

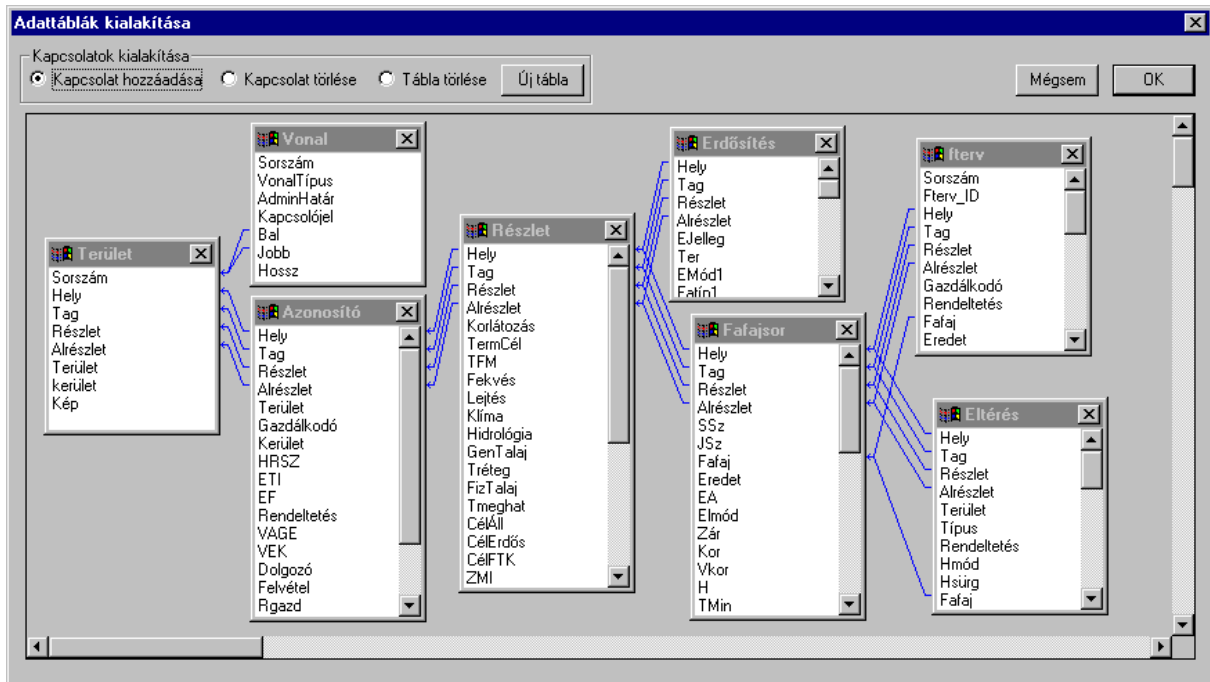
4.1. Adatbázis tervezés

A raszteres adatmodellben a kevés számú tulajdonságérték miatt nem tértem ki az adatbázis tervezésre. Vektoros adatmodellben az objektumtípusok, az objektumokat leíró tulajdonságok és az objektumok száma miatt az adatbázis tervezése, az adatmodell megválasztása egy nagyon fontos lépés (*Márkus, B. 1994*). Az adatbázis tervezés elősegíti az alkalmazás tervezését is. A megfelelő **adatmodell** kialakítása után az alkalmazás egyes elemei automatikusan generálhatók. A *DigiTerra* adatbázisok és alkalmazások fejlesztésénél sok esetben magának a programnak az adatmodellezőjét használtuk, amelynek lépéseit a következőkben ismertetem.

Adatbázis tervezés egy megfelelő adatmodell létrehozását jelenti. Az adatmodellben a valós világ egyedtípusainak felsorolása, az egyedtípusok tulajdonságainak és kapcsolatainak leírása történik. Az adatmodell kialakítása egy elméleti, egy logikai és egy fizikai fázisra különíthető el. Az **elméleti modellben** az egyedtípusok és kapcsolatok meghatározása a fő cél. Ezt a modellt szokás koncepcionális modellnek is nevezni, mivel meghatározza az adatbázis elvi működését. *DigiTerra* környezetben az elméleti modell az egyedtípusok grafikus elhelyezéséből és a közöttük lévő kapcsolatok kialakításából áll. A kapcsolat típusát már az

elméleti modellben definiálni kell. Az alárendelést nyilakkal, a mellérendelést vonalakkal jelöljük.

A **logikai modellben** az egyedtípusok leírásához szükséges jellemzők felsorolása a cél. A logikai modell első részében meghatározzuk a jellemzők csoportjait, elsőként az egyedtípusok közötti kapcsolatokért felelős csoportokat. A második részben a csoportok kibontása, majd a jellemzők végleges felsorolása következik (4-1. Ábra).



4-1. Ábra: Logikai adatmodell DigiTerra Map környezetben

A **fizikai modellben** a logikai modell számítógépes megvalósítása a cél. A megvalósítás során az egyedtípusokat objektumtípusokkal helyettesítjük. Az objektumtípusokat adattáblákban tároljuk. A tulajdonságokat az adattáblákban adatmezők képviselik. Az adatmezők létrehozásánál az adatmező neve, típusa és a tárolt adatok törvényszerűségeinek fizikai megfogalmazása a feladat.

Az adatbázis tervezés szoros kapcsolatban áll az **alkalmazás tervezéssel**. A megfelelően kialakított adatmodellből az alkalmazás bizonyos elemei automatikusan generálhatók. Ezek az elemek általában az adatfeltöltés űrlapjai, a relációs kapcsolatokból felépülő lekérdezések és a jelentések. Az előállított alkalmazás rétegei a következők: adatbázis réteg, adatelérési réteg, logikai réteg, felhasználói réteg.

4.2. Relációs adatbázis

A jelenlegi adatbázis-kezelő rendszerek szinte kivétel nélkül a **relációs adatmodellt** alkalmazzák. A relációs adatmodellben az objektumok közötti kapcsolatokat relációk, logikai kifejezések definiálják. A relációk módosíthatók, törölhetők és újabbak is megadhatók. A relációs adatmodell mellett az SQL (*Standard Query Language*) nyelv diadalát is meg kell említeni (Ullmann – Widom 1998). Az SQL nyelv egy harmadik generációs lekérdezőnyelv, melynek segítségével a lekérdezést kell megfogalmazni és nem kell törődni a megvalósítással. A nyelv tartalmaz adatdefiníciós (*CREATE*, *DROP*, *ALTER*), adatlekérdező (*SELECT*) és

adatmódosító (*INSERT*, *DELETE*, *UPDATE*) utasításokat is. A lekérdezés általános alakja a következő:

```
SELECT felsorolt adatmezők lekérdezése
FROM mezők lekérdezése a felsorolt adattáblákból
WHERE relációs kifejezés
GROUP BY csoportosítás a felsorolt adatmezők alapján
ORDER BY rendezés a felsorolt adatmezők alapján
```

A lekérdezések a relációk **optimalizálásával** gyorsíthatók. Sajnos az optimalizálás nem minden esetben kivitelezhető. Ha a kifejezésben számított tagok szerepelnek (*például két mező szorzata*), akkor a tagok értékét először az egész adattáblára ki kell számítani, majd az eredményt rendezni kell. Ha a kifejezés egy adattábla több adatmezőjéről tartalmaz relációkat, és ezek a relációk nem egyenlőségek, vagy a relációk között logikai „VAGY” művelet van, akkor nincs mód az optimalizálásra.

A hagyományos relációs modell mellett létezik egy új, mégis egyszerűbben megvalósítható modell, az **objektumrelációs modell**. Az objektumrelációs modell a relációs adatmodell egyik altípusának tekinthető. Az objektumreláció az objektumtípusok között definiált kapcsolatok alapján dolgozik. Ez a relációs típus lehetőséget ad az objektum orientált adatbázis kialakítására akár egy hagyományos relációs adatbázisban is. Az adatmodell tervezése sokkal könnyebb, az objektumtípusok közötti kapcsolatok egyszerűen áttekinthetők. Az objektum orientált modell sokkal közelebb áll az emberi gondolkodáshoz, mint a hálózatos, a hierarchikus vagy a relációs modell. Az adatbázis-kezelő szoftverek újabb verziói mind kínálnak eszközöket az objektumrelációs modell megvalósítására. A *DigiTerra Map* programban is az objektumrelációs modellt alkalmaztam.

Az objektumrelációs modellben az objektumok közötti kapcsolatok kijelölik azokat az adatmezőket, amelyeket indexelni kell. Az objektumrelációs lekérdezésekben a relációk kialakításával már nem kell foglalkozni, ezért a lekérdezések elkészítése szinte csak az objektumtípusok, az adattáblák kiválasztásából áll. Itt az objektumok közötti reláció mindig **egyenlőséget** feltételez. Több reláció esetén a relációk között mindig **logikai „ÉS” kapcsolat** van. A relációs kapcsolatban több adatmező is szerepelhet. Ezért az objektumrelációkat nagyon könnyű optimalizálni és programozni. Egy objektumhoz kapcsolódó rekordok kikeresése $\sigma(\log_2 n)$ műveletet igényel, ahol n a rekordok száma.

Az objektumokat tartalmazó adattáblákat a relációs kapcsolat kialakítása előtt indexelni kell. Az **indexelés** a relációban résztvevő adatmezők szerinti rendezést jelent. Az indexelés nem a rekordokat, hanem a rekordok sorszámait, indexeit rendezi (*Niemann 1998*). A rendezést a *DigiTerra* program valós időben végzi el. Ez azt jelenti, hogy nincsenek indexfájlok, minden index a memóriában jön létre. A **memóriában történő adatbázis-kezelés** és indexelés mellett szól, hogy a központi memória mérete egyre nagyobb, ugyanakkor egyre olcsóbb lesz. A memóriában tárolt rekordok tetszőleges sorrendben érhetők el. A háttértárak szekvenciális tárolású eszközök és nagyságrendekkel lassúbb elérésűek, mint a központi memória. Ha a háttértáron a relációs kapcsolat sorrendjében akarjuk elérni a rekordokat, akkor minden egyes rekord után az olvasófejét pozícionálni kell. A pozícionálás nagyságrendekkel több időt vesz igénybe, mint az olvasás. Egy adatfájl szekvenciális olvasási sebessége a mai számítógépeken 3-10 MB másodpercenként, ellenben a véletlenszerű olvasás csak 30-100 KB / másodperc.

Mivel az adatbázisok mérete összemérhető a jelenlegi számítógépek központi memóriájával, ezért a memória alapú adatbázis-kezelés optimális választás volt a *DigiTerra Map* program

adatbázismotorjának kialakításakor. A saját adatbázismotor fejlesztését az adatok gyors lekérdezése indokolta. Természetesen a program képes szabványos adatbázisokhoz is kapcsolódni (*OLEDB, ODBC*). Egy erdőgazdaság adatállományainak a betöltése és az indexek kialakítása néhány másodpercet vesz csak igénybe. Miután az adatok betöltődtek, a térképrajzolás, a lekérdezés és jelentéskészítés jelentősen felgyorsul. Ennek köszönhető, hogy a *Map* program egy országos méretű adatbázisból egy percen belül képes egy ezer oldalas jelentést előállítani. A **memória alapú adatbázis modellt** (*MMDM - Main Memory Database Model*) számos nagy adatbázis kiszolgálórendszer is használja. Az ilyen rendszerek egyidőben számos lekérdezést szolgálnak ki, amelyet a memóriában lévő adatok és indexek segítségével könnyen teljesítenek. Az adatok módosítása elsőként a memóriában megy végbe. Ha a kiszolgálók leterheltsége csökken, akkor kerül sor a módosított adatok kiírására. A memória alapú adatbázis-kezelésnek külön szakirodalma van (*Jagadish és mások 1994, Davis 1998*).

Több adattáblát tartalmazó objektumrelációs **lekérdezés megvalósítása** a *DigiTerra Map* programban a következő lépésekből áll. Az első feladat az **adattáblák betöltése** a háttértárolóról, ha még egy korábbi lekérdezés ezt nem tette meg. Amennyiben nincs elég hely a memóriában, akkor a legrégebben használt adattáblák törölhetők. Az adattáblák betöltése után az objektumreláció kialakítása következik. Ez a relációban résztvevő adatmezők szerinti **indexek felállítását** és rendezését jelenti. Sok esetben az indexelést már egy előző lekérdezés elkészítette. Az indexelés után következik a **rekordok permutációja**. A permutáció számlálók segítségével megy végbe. A számlálók száma a lekérdezésben résztvevő adattáblák számával egyezik meg. Kezdetben a számlálók a táblák legelső rekordja elé mutatnak. Az első adattábla esetében minden rekordra rá kell állítani a számlálót. A következő tábláknál a számlálókat az indexek alapján két eljárás mozgatja. Az eljárások a jól ismert bináris keresőeljárások változatai (*Niemann 1998*). Az első kódrészlet a kapcsolódó rekordok közül az első előfordulást, míg a második kódrészlet az utolsó előfordulást határozza meg. A kódrészletek rekordok szűrésére is használhatók.

```

FindFirst(key) // első előfordulás bináris keresése
{
  s=0; e=RecNum-1; diff=1; // változók inicializálása
  while(s<=e) // ciklus, amíg az alsó index kisebb, mint a felső
  {
    m=s+e>>1; // középső index kiszámítása
    diff=Compare(key, m); // középső rekord hasonlítása a kulcshoz
    if(diff<0) s=m+1; else e=m-1; // alsó vagy felső index mozgatása
  }
  if(diff==0) return m; else if(s>=RecNum) return -1; // találat
  if(Compare(key, s)==0) return s; else return -1; // kezdet
}

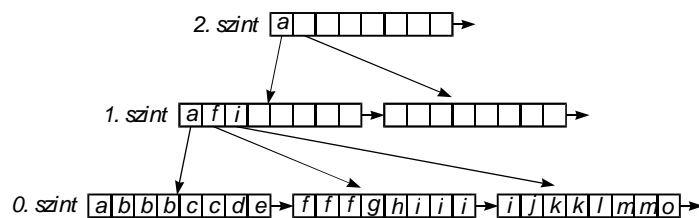
FindLast(key) // utolsó előfordulás bináris keresése
{
  s=0; e=RecNum-1; diff=1; // változók inicializálása
  while(s<=e) // ciklus, amíg az alsó index kisebb, mint a felső
  {
    m=s+e>>1; // középső index kiszámítása
    diff=Compare(key, m); // középső rekord hasonlítása a kulcshoz
    if(diff<=0) s=m+1; else e=m-1; // alsó vagy felső index mozgatása
  }
  if(diff==0) return m; else if(e<0) return -1; // találat
  if(Compare(key, e)==0) return e; else return -1; // kezdet
}

```

A rekordok permutációja után a szűrések, a csoportosítás és a rendezés következik. A **szűrés** minden egyes rekordkapcsolat esetén egy összetett kifejezés kiértékelését jelenti. A kifejezés gyors kiértékelése a kifejezés gépközeli kódra fordításából és végrehajtásából áll. A kifejezés fordító és kiértékelő saját fejlesztés eredménye. Amennyiben a kifejezés eredménye hamis,

akkor a rekordkapcsolatot ki kell venni a lekérdezésből. A **csoportosítás** lényege a csoportosítandó kifejezések szerinti rendezés és összevonás. Az összevonás, más szóval aggregáció típusai a következők lehetnek: *első érték, középső érték, utolsó érték, gyakori érték, összeg, darabszám, minimum, maximum, átlag, szórás*. A csoportosítást követően az értékek megváltoznak, amelyen újabb **szűrés** végezhető egy másodlagos szűrőkifejezéssel. Az utolsó lépésnek tekinthető az adatok **rendezése** valamely adatmező vagy adatmezők alapján. A rendezéshez mindenhol a *QuickSort+BubbleSort* algoritmust használom (Niemann 1998). A lekérdezés eredménye megjeleníthető, szerkeszthető és elmenthető.

Ha egy adattáblában gyakran történik adatmódosítás, akkor az indexek elkészítése, a rendezés időigényes folyamat. A már rendezett indexek közé a megfelelő helyre egy új index beszűrhető. A beszűrés viszont nagy indextömbök esetén időigényes, ugyanis a beszűrés hely fölötti indexeket feljebb kell mozgatni. Egy lehetséges megoldás az **indextömb darabolása**. Az egyes darabokon az új index elhelyezése, a beszűrés már rövid idő alatt elvégezhető. A darabolt index kezelése viszont összetettebb programozást igényel. Az indextömb darabjai egy **bináris fa** segítségével kapcsolhatók össze (4-2. Ábra). Bináris fában az adott szint darabjaira a föltte lévő szint elemei mutatnak. Ha egy darab megtelik indexekkel, akkor azt kétfelé kell választani és az új részt be kell szűrni az aktuális és a következő darab közé. Ha a darabokból indexeket törölünk, akkor előfordulhat, hogy egy darab kiürül, ilyenkor a darabokat össze kell vonni. A darabolás és összevonást a felső szinteken is végig kell vezetni.



4-2. Ábra: Bináris fa indexelés

Megfelelő szétválasztási és összevonási eljárásokkal egy kiegyensúlyozott bináris hozható létre. A kiegyensúlyozás optimális helykihasználást és azonos keresési időket eredményez. A darabolt indexelést és bináris fa alapú elérését a szakirodalom röviden *B-Tree* indexelésnek hívja (Niemann 1998). A *B-Tree* indexnek számos változata ismert, amelyek elsősorban a számítógépes megvalósításban, a darabolás és az összevonás eljárásában térnek el. A *DigiTerra* programok a bináris fa indexelés egy tömör, memória-alapú változatát használják.

4.3. Térbeli indexelés

A lekérdezések a rekordok sorszámainak megfelelő szempont szerinti rendezésével, a rekordok indexelésével gyorsíthatók fel. Geoinformatikai rendszerekben a lekérdezések a leíró adatokon kívül a geometriai adatokra is irányulnak. A geometriai adatok alapján történő lekérdeztést **térbeli lekérdezésnek** hívjuk. A térbeli lekérdezések a geometriai elemek indexelésével gyorsíthatók fel. Néhány példa a térbeli lekérdezésekre:

- zárt sokszögön belüli objektumok keresése,
- zárt sokszöggel átfedő objektumok keresése,
- geometriai alakzattal érintkező objektumok keresése,
- vonallánccal elmetezett objektumok keresése,
- ponttól adott távolságra lévő objektumok keresése.

Raszteres adatoknál a tárolás térbeli struktúrája megoldja a térbeli indexelést. A vektoros adatok hatékony lekérdezése viszont csak megfelelő térbeli indexelési módszerekkel lehetséges. Térbeli indexelésnek bőséges a szakirodalma (*Böhm és mások 1998*). Sok átfogó és összehasonlító mű készült (*Procopiuc 1997*). A térbeli indexelési eljárásokat célszerű a geometriai elemek kiterjedése alapján csoportosítani. Az első nagy csoport a kiterjedés nélküli pontok indexelését célozza meg.

Az egyik legrégebbi és leggyakrabban használt eljárás a **négyesfa** (*quadtree*) algoritmus, mely a kétdimenziós pontok által elfoglalt területet hierarchikusan negyedeli. A négyesfa algoritmus a bináris fa kétdimenziós változatának tekinthető. A legfontosabb eltérés, hogy a négyesfa index nem kiegyensúlyozott, ezért gyakori módosítás esetén az indexeket újra és újra fel kell építeni. A négyesfa algoritmus többdimenziós változata a **többesfa** (*k-d-Tree*) indexelés. Ennek kiegyensúlyozott változata a *k-d-B-Tree*, amelyben a geometriai elemek hozzáadása és törlése is megoldott. Nagyon hatékony eljárás a pontok indexelésére egy raszter bevezetése. A pontok sorszámait a raszter megfelelő cellái tárolják. Az indexelést **raszter-vagy rácsindexnek** nevezzük. Az eljárás előnye, hogy a felépítés és a keresés gyors, hátránya, hogy az index nem kiegyensúlyozott és bizonyos módosítási művelet után újra kell építeni.

A térbeli indexelésben a legnagyobb problémát a kiterjedéssel rendelkező geometriai elemek indexelése okozza. Az objektumok kiterjedését az egyes dimenziókban az objektumok minimális és maximális koordinátái, a befoglaló koordináták írják le. Amennyiben a lekérdező alakzat is kiterjedt alakzat, akkor a lekérdezési feltétel két dimenzió esetén a következő:

$$\begin{aligned} \text{geometria.xMin} &\leq \text{lekérdez.xMax} \text{ \&ES } \text{geometria.xMax} \geq \text{lekérdez.xMin} \text{ \&ES} \\ \text{geometria.yMin} &\leq \text{lekérdez.yMax} \text{ \&ES } \text{geometria.yMax} \geq \text{lekérdez.yMin} \end{aligned}$$

A feltételben a *geometria* a vizsgált alakzatot, a *lekérdez* a lekérdező alakzatot jelenti. Az *xMin*, *yMin*, *xMax*, *yMax* tulajdonságok az objektumok befoglaló koordinátáit tárolják. A feltétel a vizsgált alakzat több adatmezőjére is felállít egy-egy relációt, amit egyszerű rendezéssel nem lehet optimalizálni. A fenti lekérdezés felgyorsítására számos eljárás született. Az eljárások egyik része megpróbálja visszavezetni az indexelést az egydimenziós esetre, a másik csoport pedig régiókat definiál. A régiók lehetnek átfedő vagy elkülönülő régiók, tartalmazhatják a geometriai egy részét vagy egészét.

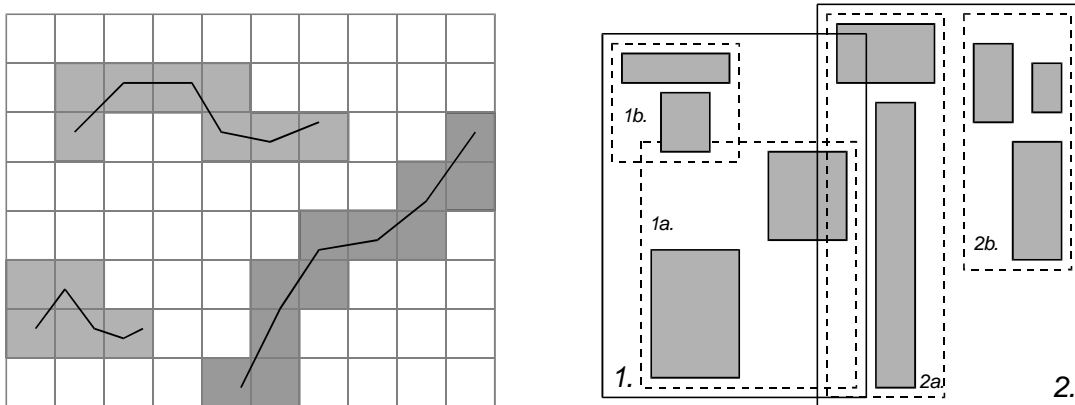
Az egydimenziós esetre példa a **z-függvény** bevezetése, mely az objektumok tulajdonságainak összevonásával egy új értéket állít elő. A z-függvény megalkotása nem egyszerű feladat. Ismerni kell a tulajdonságok értéktartományát és várható értékét. Az átalakítás az értékek normalizálásával és diszkrét csoportokba sorolásával folytatódik. Az egyes tulajdonságok diszkrét csoportjait ezután felváltva kell összeállítani. A *z-függvény* megalkotásában az első két lépés a problematikus.

Kiterjedő alakzatok esetén is használható a **raszter- vagy rácsindexelés** (4-3. *Ábra*). A pontok indexelésére használt módszert úgy kell kiegészíteni, hogy az objektum által lefedett indexcellák mindegyikébe be kell jegyezni az objektum sorszámát. A rácsméret megválasztásánál az átlagos objektumméret a mérvadó. Sajnos ez az indexelés sem kiegyensúlyozott.

Bizonyos módosításokkal a **négyesfa** eljárás is használható kiterjedt geometriai elemek indexelésére. A módosítás abban rejlik, hogy a négyesfa magasabb szintjein is helyezzünk el objektumokat. Az elhelyezés azon a szinten történik, ahol az objektumot elmetszik a negyedelő vonalak (*quad-CIF-tree*). Az eljárás hátránya, hogy nem kiegyensúlyozott és sok

kis kiterjedésű objektum kerül a felsőbb szintekre azért, mert épp a negyedelő vonalakon helyezkedik el. A négyesfa másik módosítása ezt az utóbbi hátrányt próbálja ellensúlyozni azzal, hogy a negyedek között bizonyos átfedést (10-20%) is megenged.

Kiegyensúlyozott indexelési eljárást biztosít a **régiófa** (*R-tree*) eljárás (4-3. Ábra), amely átfedő régiókba gyűjti a geometriai elemeket (Guttman 1984). A régiók indexelése darabolással és bináris fa segítségével történik. A darabolás során a régiót úgy kell szétválasztani, hogy a két rész lehetőleg azonos számú elemet tartalmazzon és a részek közötti átfedés minimális legyen. A régiófa eljárásban a darabolás megvalósítása a legnehezebb. Az eljárásnak a *B-Tree* indexeléshez hasonlóan számos változata van (*R+Tree*, *X-Tree*). A régiófa magasabb dimenziójú geometriai elemek indexelésére is használható.



4-3. Ábra: Raszter- és régiófa indexelés

A térbeli adatok tárolása és térbeli indexelés a mai napig az elsődleges oka, amiért a geoinformatikai rendszerek saját adatformátumot használnak. A térbeli indexeket sok esetben bonyolult szerkezetű adatfájlok tárolják. A szabványos adatbázis-kezelő rendszerek közül csak nagyon kevés és azok sem egységes módon valósítják meg a térbeli indexelést. Az *SQL* lekérdezőnyelv geometriai kiterjesztésére is számos próbálkozás történt (*GQL*). Az *OpenGIS* konzorcium egyik legfontosabb törekvése, hogy ezeket a problémákat megoldja és javaslatokat tegyen (*OpenGIS 1998*).

Az általam kidolgozott térbeli indexelés bármilyen dimenziójú és tetszőleges kiterjedésű geometriai elem esetén használható. Az eljárás további előnye, hogy tetszőleges adatbázisban megvalósítható, ugyanis a geometriai indexek tárolása nem külön állományokban, hanem a rekordon belül történik. A geometriai indexek az objektum **kiterjedésére és a középpontjának helyzetére utalnak**. Az új indexek alapján az indexelést ***EC-indexnek*** (*Extension & Center Indexing*) neveztem el. A geometriai indexmezők, amelyekkel a geometriai elemeket tároló adattábla mezőit bővíteni kell, a következők:

- *spExtension* – az objektum maximális kiterjedését leíró érték,
- *spCenterX* – az objektum középpontjának *x* koordinátája a kiterjedés egységében,
- *spCenterY* – az objektum középpontjának *y* koordinátája a kiterjedés egységében.

Új elem létrehozásakor vagy az indexekkel nem rendelkező adattábla indexeléskor a fenti adatmezőket létre kell hozni és fel kell tölteni. Az elem maximális kiterjedését egy előre meghatározott érték egész számú hatványaira kell fölfelé kerekíteni. Az eddigi vizsgálatok alapján a tízes értéket javaslom, így a kiterjedések például a következők lehetnek: 0.1, 10,

100, 1000. Az érték megválasztásához nem szükséges a geometriai elemek által elfoglalt területrészt előzetes ismerete, mint például a raszteres indexeknél. Az értéksorból azt az értéket kell választani, amely az objektum befoglaló méreteinél nagyobb. A geometriai elem középpontja a befoglaló koordinátákból és a maximális méretből a következő kódrészlettel számítható:

```
spCenterX = Int((xMin + xMax)*0.5 / spExtension);
spCenterY = Int((yMin + yMax)*0.5 / spExtension);
```

A kódrészletben az *xMin*, *yMin*, *xMax*, *yMax* értékek az objektum befoglaló koordinátáit tárolják. Az *Int* függvény a számított értékek egész részét képezi. Ha a képlet eredményei és a középpontok közötti eltérés nagyobb egynél, akkor a maximális kiterjedést növelni kell.

Az indexmezők létrehozása és feltöltése után az **indexelésre** a jól bevált *B-Tree* eljárás a legalkalmasabb. Az indexelés a geometriai indexmezőkből létrehozott kompozit index alapján történik. A kompozit index egy egydimenziós index, melyet az adatbázis-kezelők is el tudnak készíteni és karban tudnak tartani. A *B-Tree* algoritmus kiegyensúlyozott indexelést biztosít még gyakori módosítás esetén is.

A térbeli lekérdezés a lekérdező alakzat befoglaló méreteivel történik. A lekérdezést a maximális kiterjedés valamennyi értékére külön el kell végezni. A kiterjedés alapján szintekről beszélünk. A lekérdezés az *spCenterX* és az *spCenterY* értékeinek tartományaira hivatkozik. A lekérdezést a kompozit indexnek köszönhetően az adatbázis-kezelők képesek optimalizálni. Az egyes szintek lekérdezéseit az *SQL* nyelv *UNION* parancsával össze is lehet fűzni. Egy szint lekérdezésének *SQL* formulája:

```
SELECT * FROM table WHERE spExtension=szint AND minY<=spCenterY AND
spCenterY<=maxY AND minX<=spCenterX AND spCenterX<=maxX
```

A lekérdezésben az *xMin*, *yMin*, *xMax*, *yMax* értékek a lekérdező alakzat befoglaló koordinátáit jelentik. A lekérdezés *C* programozási környezetben is megvalósítható. A maximális kiterjedés értékcsoportjait, a szinteket érdemes fentről-lefelé végigjárni. Ilyenkor az objektumok nagyságuk sorrendjében rajzolódnak ki. A programvázlat a következő:

```
for(level=maxLevel; level>=minLevel; level/=10) // szintek lekérdezése
{ for(y=minY; y<=maxY; y++) // y tartomány lekérdezése
  { start=FindFirst(level, y, minX); // x tartomány első előfordulása
    if(object[start].y > y) y=object[start].y; // y tartomány javítása
    end=FindLast(level, y, maxX); // x tartomány utolsó előfordulása
    for(i=start; i<=end; i++) { ... } // objektumom feldolgozása
  }
}
```

Az indexelés és a lekérdezés könnyedén kiterjeszthető a harmadik dimenzióra is. Az indexelés egy topológikus vektoros állományon működik hatékonyan, ahol az objektumok kiterjedése közel azonos. Ezekben az állományokban a vonalláncok nem metszik egymást, csak a csomópontokban találkoznak. Az *EC-indexelést* összehasonlítva a régiófa indexekkel a következők állapíthatók meg:

- mindegyik kiegyensúlyozott és hatékony indexelési módszer,
- az *EC-indexelés* megvalósítása egyszerűbb,
- az *EC-index* meglévő adatbázisokban is működik,
- a régiófa index bonyolult régiódarabolási és összevonási eljárást használ,

- az EC-index befoglaló mérete csak négyzet alakú és fix oldalméretű lehet.

4.4. Valós idejű topológia

A topológia az objektumok térbeli kapcsolataival foglalkozik. A vektoros adatok esetében a topológia fontos szerephez jut. A topológia figyelembe vételével a vektoros adatokon végzett műveletek egyszerűsíthetők és felgyorsíthatók. A térbeli kapcsolatok kialakítása hozzájárul a **konzisztens és hibamentes vektoros állományok** létrehozásához. A topológia egyszerűsíti a lekérdezési és kirajzolási műveleteket is.

Ha a vektoros adatok előállításakor a térbeli kapcsolatok kialakítását is szem előtt tartjuk, akkor a **földmérési, térképezési gyakorlathoz** leginkább igazodó eljárást kapunk. Az eljárás elején a felmérési vagy digitalizált pontokat vonalláncokkal kötjük össze. A vonalláncok egymást nem metszik, kizárólag a **csomópontokban** találkoznak. Ha a vonalláncok mégis metszenék egymást, akkor ott metszéspontokat kell létrehozni és a vonalláncokat el kell darabolni. A záródó vonalláncok területeket hoznak létre. A topológia az egyes geometriai elemek esetében a következő kapcsolatok tárolását kívánja meg:

- Pont – mely vonalláncok felépítésében vesz részt.
- Vonallánc – mely pontokat érint, mely vonalláncokhoz kapcsolódik a vonallánc elején és végén, mely területek felépítésében vesz részt.
- Terület – mely vonalláncok alkotják.

Ha a pont három vagy több vonallánc felépítésében is részt vesz, akkor csomópontról beszélünk. A topológia hiányosságai rávilágítanak a vektoros állomány hibáira, a hézagokra, túlnyúlásokra, kettőzésekre.

A *DigiTerra Map* program tervezésekor egy olyan vektoros térképező eszköz megalkotását tűztem ki célul, mely a térképezés művelete alatt kialakítja a térbeli kapcsolatokat és feltárja a szerkesztés hibáit. A **topológia valós időben** történő kialakítása mind a mai napig csak néhány geoinformatikai program sajátossága. A program legfontosabb funkciói közé tartozik az új pontok és vonalláncok felvétele, a metszéspontok automatikus kialakítása, a meglévő elemek szerkesztése, a területek létrehozása záródó vonalláncokból és a hibák kijelzése. A funkciók közül ebben a fejezetben a vonalláncok topológiájával foglalkozom.

A vonallánc tárolása előtt vizsgáljuk meg két vonalszakasz viszonyát és az **elemi vonalmetszési feladatot**. A vonalszakaszok metszését célszerű paraméteres egyenletek segítségével elvégezni. A P_1, P_2 pontok közötti szakaszt jelöljük p -vel, a Q_1, Q_2 pontok közötti szakaszt q -val. A vonalszakaszok paraméteres vektoregyenlete a következő:

$$\begin{aligned}\bar{P} &= (\bar{P}_2 - \bar{P}_1) \cdot t + \bar{P}_1, & t &= 0 \dots 1 \\ \bar{Q} &= (\bar{Q}_2 - \bar{Q}_1) \cdot u + \bar{Q}_1, & u &= 0 \dots 1\end{aligned}\tag{4-1}$$

A t paraméter a p szakasz mentén, az u paraméter pedig a q szakasz mentén vesz fel nulla és egy közötti értékeket. A skaláris egyenletek az alábbiak:

$$\begin{aligned}P_x &= (P_{2x} - P_{1x}) \cdot t + P_{1x}, & P_y &= (P_{2y} - P_{1y}) \cdot t + P_{1y} \\ Q_x &= (Q_{2x} - Q_{1x}) \cdot u + Q_{1x}, & Q_y &= (Q_{2y} - Q_{1y}) \cdot t + Q_{1y}\end{aligned}\tag{4-2}$$

Vezessük be a következő segédváltozókat:

$$\begin{aligned} o_x &= P_{2x} - P_{1x}, & o_y &= P_{2y} - P_{1y} \\ n_x &= Q_{2x} - Q_{1x}, & n_y &= Q_{2y} - Q_{1y} \\ d &= o_x \cdot n_y - o_y \cdot n_x \end{aligned} \tag{4-3}$$

A vonalak metszéspontja ($P=Q$) a paraméterek meghatározásával kezdődik:

$$\begin{aligned} t &= \frac{n_x(P_{1y} - Q_{1y}) + n_y(Q_{1x} - P_{1x})}{d} \\ u &= \frac{o_x(P_{1y} - Q_{1y}) + o_y(Q_{1x} - P_{1x})}{d} \end{aligned} \tag{4-4}$$

Ha a két vonalszakasz metszi egymást, akkor t és u paraméter értéke mind a $[0;1]$ tartományba esik. A paraméterek ismeretében a metszéspont:

$$\begin{aligned} P_x = Q_x &= P_{1x} + o_x \cdot t = Q_{1x} + n_x \cdot u \\ P_y = Q_y &= P_{1y} + o_y \cdot t = Q_{1y} + n_y \cdot u \end{aligned} \tag{4-5}$$

A metszésvizsgálatba egy igazítási távolságot is be kell vonnunk. Ez a távolság a vektoros szerkesztés kulcseleme, amelyet **szerkesztési toleranciának** nevezünk. A távolság bevonásával az igazítási és metszési feladatok automatizálhatók. A szerkesztési távolság az elemi vonalmetszési feladat kilenc alapesetét különíti el (4-4. Ábra).

p és q szakasz	p igazítása $-s_p < t < s_p$	p darabolása $s_p \leq t \leq 1-s_p$	p igazítása $1-s_p < t < 1+s_p$	
q igazítása $-s_q < u < s_q$				p és q szakasz darabolása és igazítása
q darabolása $s_q \leq u \leq 1-s_q$				p szakasz: $t=0...1$ $s_p=s/ p $
q igazítása $1-s_q < u < 1+s_q$				q szakasz: $u=0...1$ $s_q=s/ q $

ahol s_a szerkesztési tolerancia

4-4. Ábra: Két vonalszakasz metszésének alapesetei

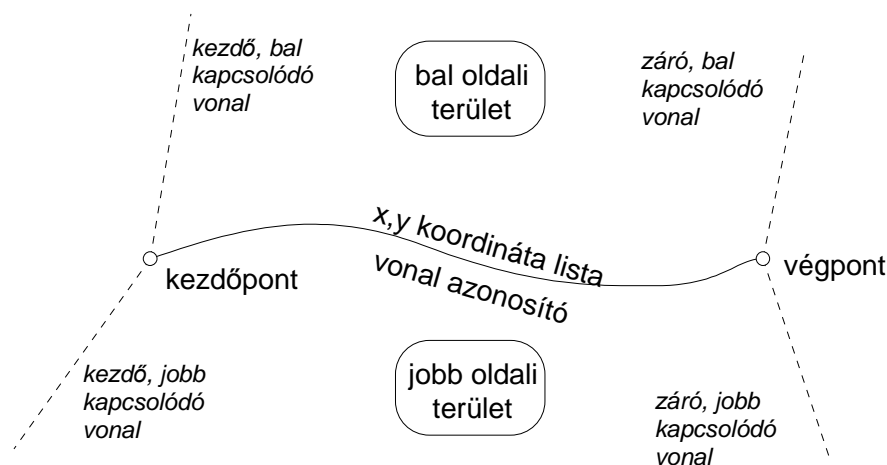
Az alapeset kiválasztása a tolerancia nagyságától és a paraméterek értékétől függ. Az alapeset kiválasztása után a vonalszakaszok végpontjai igazíthatók és a vonalszakaszok metszése, darabolása elvégezhető.

Az adatállományhoz egy **új vonallánc hozzáadása** az új vonal közelében lévő már tárolt vonalak betöltésével kezdődik. A közelséget az új vonal befoglaló koordinátái és az igazítás távolsága, a tolerancia befolyásolja. A vonalláncot vonalszakaszonként kell feldolgozni. Az első feladat a vonalszakasz két végpontjának igazítása a meglévő vonalakhoz. Az igazítás az elemi vonalmetszés feladata alapján történik. Az igazítás során a vonalszakasz végpontjait elmozdítjuk és a régi vonalláncokat a töréspontokban vagy a szakaszok mentén eldaraboljuk.

Az új vonalszakasz igazítása után következik a **régi vonalláncok igazítása** az új szakaszhoz, illetve a régi vonalláncok metszésvizsgálata az új szakasszal. Az igazítás és metszésvizsgálat szintén az elemi vonalmetszési feladat alapján dolgozik. Az igazítás a régi vonalláncok töréspontjait igazítja az új szakaszhoz és egyben a vonalláncot és a szakaszt is eldarabolja. Az új vonallánc szakaszainak feldolgozása után a szakaszok ismét összekapcsolhatók és az adatállományban vagy az adattáblában tárolhatók. A tárolás után azonnal el kell végezni az új elem térbeli indexelését, mivel ha újabb vonallánc jön, akkor már ezt is be kell vonni a vizsgálatba. Más szoftverekben használt eljárások vagy nem valósidejű topológiát, vagy interaktív igazítást használnak. Előbbi hátránya, hogy valós idejű szerkesztésre nem használható. Az interaktív igazítás hátránya, hogy a valós idejű metszést és darabolást nem végzi el, csak az új töréspontok igazítását a meglévő geometriai elemekhez.

Új vonallánc hozzáadásakor a topológiai adatok meghatározása is fontos feladat. Egy vonallánc topológiáját (4-5. Ábra) a következő adatok biztosítják:

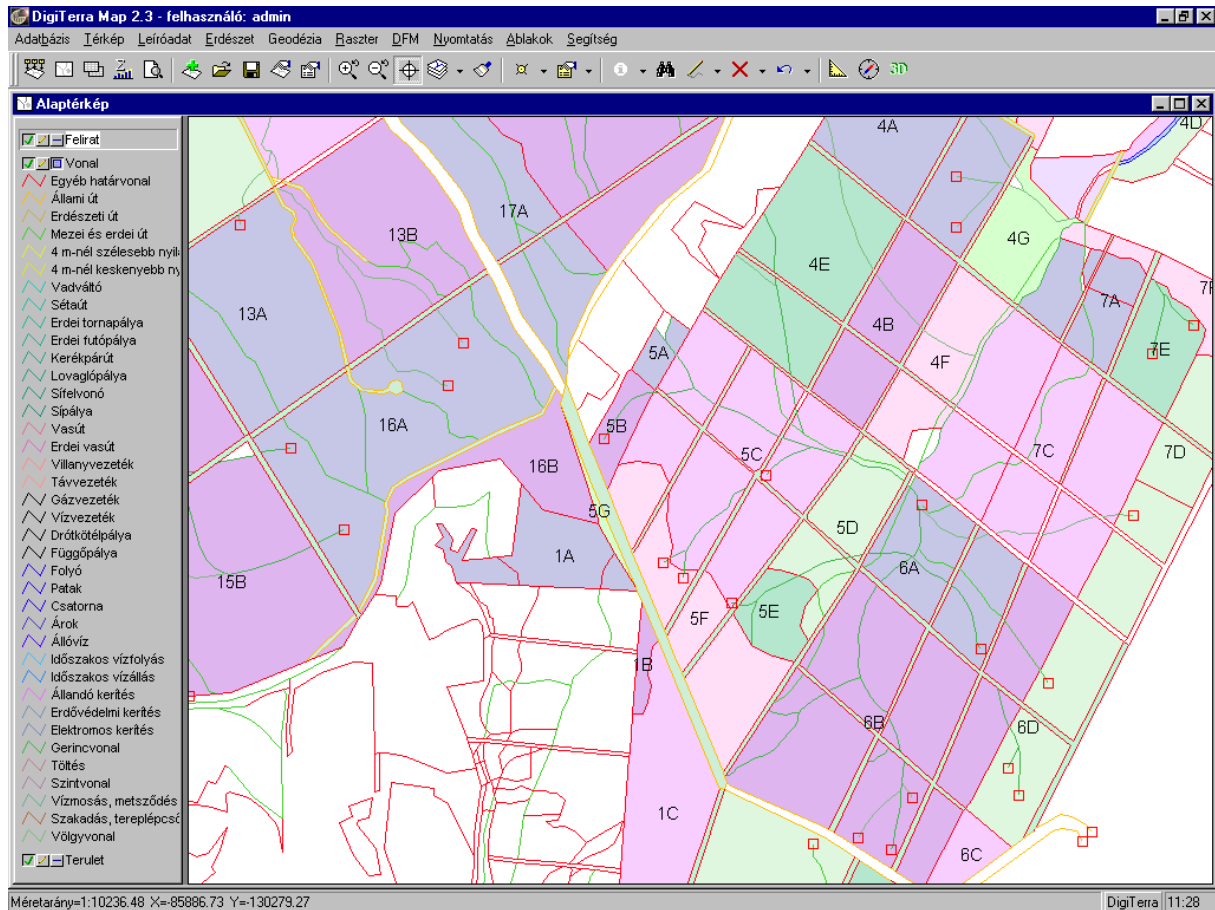
- a vonallánc elején és végén lévő két csomópont azonosítója,
- a köztes töréspontok azonosítói,
- kezdő- és végponthoz kapcsolódó vonalláncok azonosítói,
- a vonallánc két oldalán lévő terület azonosítója.



4-5. Ábra: Egy vonallánc topológiája

A vonallánc két csomópontja (*kezdőpont* és *végpont*), illetve a kapcsolódó vonalláncok azonosítása az igazítási és metszési feladat során határozható meg. A töréspontok rendszerint nem alkotnak önálló geometriai elemet, hanem a vonallánc geometriájában tárolódnak, ezért az azonosításra nincs szükség. A vonallánc két oldalán lévő terület csak akkor azonosítható, ha a területek már léteznek. A létező terület a korábbi vonalláncok terület azonosítóiból ismerhető fel. Egy vonalláncok által körbezárt területen belül minden egyes vonallánc belső terület felé eső azonosítója megegyezik.

A topológia valós idejű kialakítása után a szerkesztési hibák azonnal felderíthetők. A szerkesztési hiba kijelzése a csomópontok vizsgálatán alapul. Ha a csomópontba egy vonallánc fut be, akkor **csomópont vagy végpont hibáról** beszélünk (4-6. Ábra). A végpont hiba rövid vagy túlnyúló vonallánc esetén alakul ki. A rövid vonallánc az utolsó szakasz megnyújtásával, a hosszú vonallánc pedig a túlnyúló szakasz törlésével javítható. A törlési és nyújtási funkciók a topológia segítségével könnyedén kivitelezhetők.



4-6. Ábra: Vonalszerkesztés DigiTerra környezetben.
Az ábrán piros négyzetek jelölik a végponthibákat.

A topológia kialakítása után az érintkezésre, kapcsolódásra, határosságra irányuló térbeli lekérdezések egyszerű és hatékony eljárásokkal valósíthatók meg. A topológiai hálózatok (út- és vasúthálózat, számítógépes hálózatok) elemzésében a kezdő- és végponthoz kapcsolódó vonalláncok azonosítói vesznek részt. A **hálózatelemzés** olyan funkciókat takar, mint a hálózat pontjai közötti legrövidebb út felkeresése, a hálózat adott pontjából időegység alatt elérhető terület meghatározása, vagy az útvonal tervezés és optimalizálás.

A valós idejű topológia **térképek generalizálására** is használható. Kisebb méretarányú digitális térképek készítéséhez kisebb adatsűrűség is elegendő. A kisebb adatsűrűség a szerkesztési tolerancia növelésével érhető el. A szerkesztési tolerancia biztosítja, hogy a tolerancia távolságon belül nem lehet más töréspont. Ezért a vektoros térkép adatsűrűségének szabályozásában a minimális feltételt képezi. A szerkesztési tolerancia előzetes megválasztása után a digitális térképek generalizálása könnyen megoldható.

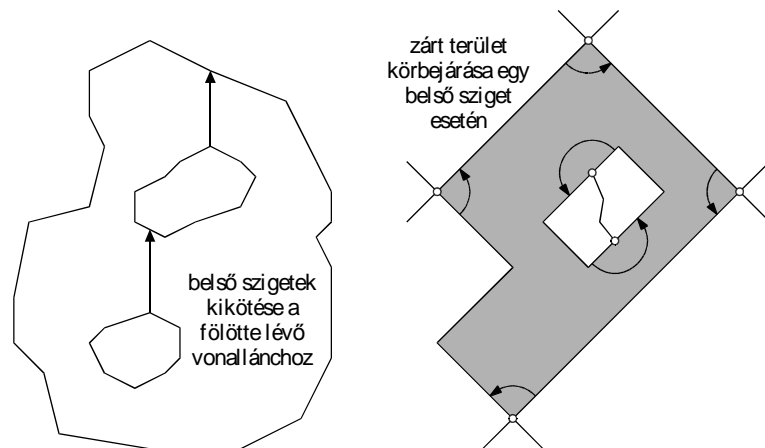
Az általam kidolgozott valós idejű topológia már bizonyította eredményességét. Az ország erdőterületeinek mintegy 60%-ról készült el a digitális térkép ezzel a technológiával. Sok erdőtervezési irodánál a valós idejű topológiát az új térképek előállítása mellett meglévő térképek aktualizálására is használják. Az erdészeti térképen kívül földmérési és egyéb tematikus térképek szerkesztésére is alkalmas.

4.5. Területek építése

Geoinformatikai rendszerek egyik fontos eljárása a **vektoros területek létrehozása**. A területeket legkönnyebben a töréspontok kijelölésével adhatjuk meg. Ebben az esetben viszont két terület érintkező oldalain a töréspontokra többször is rá kell mutatnunk, ami nagyban lassítja a munkát. Sokszor a területeket határoló vonalakat is elő kell állítani, de az is előfordulhat, hogy a vonalak már rendelkezésünkre állnak. A vonalak és területek együttes, töréspontenkénti létrehozása a lehetséges ráfordítási idő háromszorosát igényli.

A területek gyors elkészítésében a **vonalláncok topológiája** segíthet. Ha ismerjük a vonalláncok végpontjaiban a kapcsolódó vonalláncok azonosítóit, akkor a körbezárt területeket gyorsan és automatikusan határozhatjuk meg. A vonallánc topológiája a belső szigetek és az összetett, több poligonból álló területek létrehozásában is gyors és egyértelmű algoritmus fejlesztését teszi lehetővé.

Ha a területen **belső szigetek** is találhatóak, akkor a végponti kapcsolódásokon kívül a szigetek kikötése is szükséges (4-7. *Ábra*). A kikötés előtt a vonalláncokat a maximális függőleges koordináta alapján sorba kell rendeznünk. A kikötés során a szigetet alkotó vonalláncokat a közvetlenül fölötté húzódó vonalláncokhoz kell kapcsolnunk. Ha a vonallánc topológiája és a belső szigetek kikötése is megtörtént, akkor a területek **körbejárása** már könnyen megoldható. A körbejárás elvégezhető egy-, több- vagy az összes zárt terület esetén is (4-7. *Ábra*).



4-7. *Ábra*: Belső szigetek kikötése és a zárt terület körbejárása

Egy terület körbejárásához a terület egy belső pontját kell megadni. A körbejáráshoz meg kell határoznunk közvetlenül a pont fölött húzódó vonalláncot. A vonalláncból kiindulva a topológia felhasználásával a területet határoló vonalak körbejárhatók. A körbejárás során egy vonallánc végpontjából mindig az óramutató járásával ellentétes irányú legközelebbi vonalláncre kell átlépni. A körbejárt terület nagysága a területet alkotó sokszögek területének előjeles összegeként áll elő. Az egyes sokszögek területe pozitív lesz zárt terület esetén, és

negatív nyitott terület vagy belső sziget esetén. A területszámítás a geodéziában is használatos képlettel történik (Márkus, B. 1994):

$$T = \sum_{i=1}^n \frac{(x_{i+1} - x_i)(y_{i+1} + y_i)}{2} \quad (4-6)$$

ahol: T egy sokszög területe
 n sokszöget felépítő töréspontok száma
 x_i, y_i i -dik töréspont koordinátái (az $n+1$ -dik töréspont azonos az elsővel)

A körbejárás során az érintett vonallánccokat össze kell fűzni és azokból poligonokat kell építeni. A belső szigetek fordított körbejárása fontos feltétel. Csak ekkor ad helyes eredményt a megjelenítés és a területszámítás. A körbejárás után az egy, esetleg több poligonból álló területet el kell tárolni. A tárolás lehetséges a vonallánccok azonosítójával vagy a vonallánccok töréspontjaival is. A körbejárás során a kiszámított terület nagyságát a leíró adatok megfelelő adatmezőjébe lehet írni.

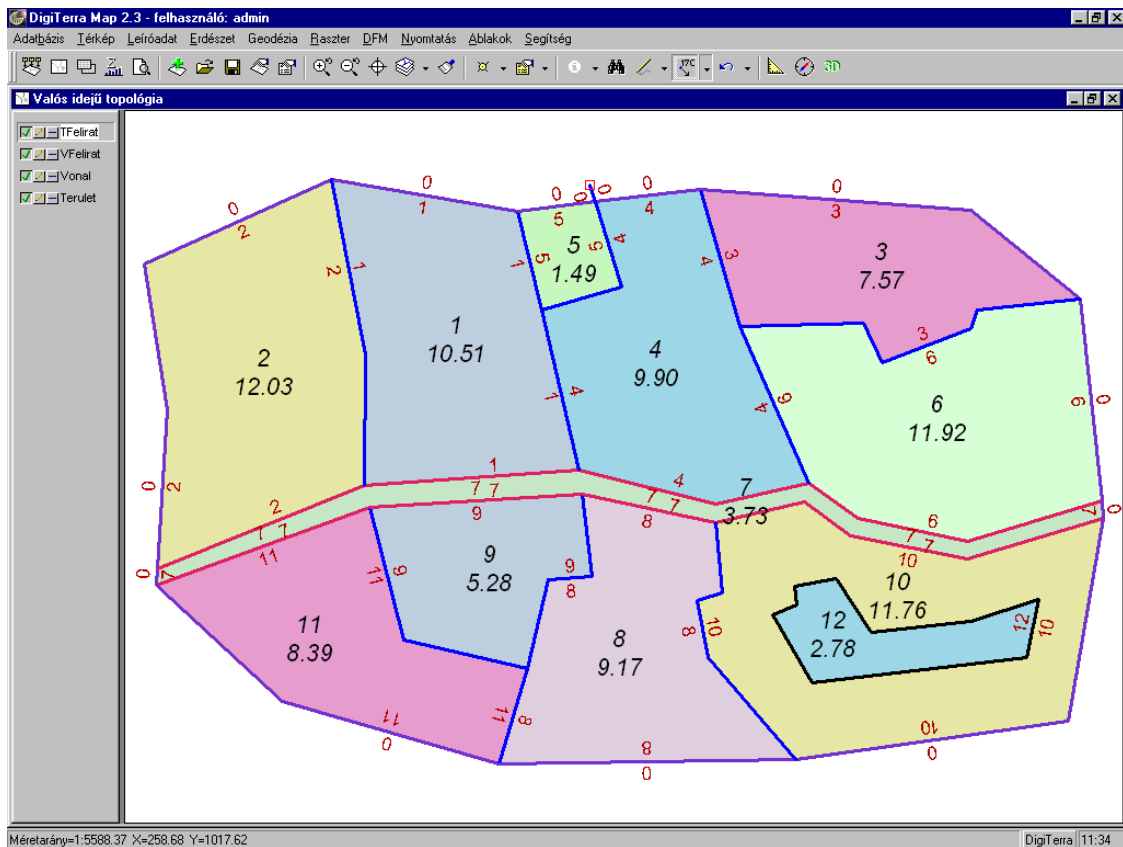
Több elemi területből álló **komplex terület** létrehozása az elemi területek belső pontjainak megadásával indul. A komplex terület építése megegyezik az egy terület építési eljárással. Az eltérés csak annyi, hogy a területek poligonjait össze kell fűzni és együtt tárolni. Szintén az egy terület építési eljárást használja az **összes terület** építési funkció. Az építkezés során nem kell megadnunk belső pontokat. Az algoritmus valamennyi vonallánccot körbejárja. A terület építésben már részt vett vonallánccokat az eljárás megjelöli. Az összes zárt terület körbejárása, köszönhetően a topológiának, egy nagyon gyors eljárás.

Gyakran előfordul, hogy a vonallánccokból már előzőleg építettünk területeket. A vonalak viszont időközben módosultak. A módosítás kiterjedhet meglévő vonalak törlésére, vagy új vonalak létrehozására. Egy vonallánc törlésével területek vonhatók össze, új vonallánc megadásával pedig meglévő területek oszthatók fel. A módosítások átvezetéséhez a területeket újra kell építeni. Az újraépítéssel szemben támasztott alapvető igény, hogy a **meglévő területek** leíró adatait megtartsa. Az átvétel a régi területek középpontjainak térbeli kapcsolásával történhet. A régi középpont kapcsolása az új területhez a térbeli indexeléssel gyorsítható fel.

A területépítés után a felhasznált vonallánccok bal és jobb oldali terület azonosítóit aktualizálni kell. A bal és jobb oldali azonosítók képezik a digitális térkép **vonaltérület topológiáját** (4-8. Ábra). Ez a kapcsolat biztosítja az egyes területek szomszédsági viszonyait. A szomszédság alapján számos térbeli elemzés valósítható meg. A bal és jobb területek leíró adatainak összehasonlítása alapján meghatározható a közöttük húzódó vonallánc típusa. Ha a vonalas elem két oldalán lévő területek más-más község határba tartoznak, akkor a vonal község határ. Ha más az erdőtag azonosító, akkor a vonal taghatár, különben részlethatár. Teljes egyezés esetén akár a területek között húzódó vonallánccot is törölhetjük. Ez a példa rávilágít, hogy a térképezés során a természetes vonalakat kell térképezni, a mesterséges és az adminisztratív határokat pedig a topológia és a területek leíró adataiból kell levezetnünk.

Néha a **területépítés fordítottjára** is szükségünk lehet. Ha a vonalas állomány nem áll rendelkezésünkre, viszont a területeket a valós idejű topológia funkcióival akarjuk szerkeszteni, akkor a területekből elő kell állítani a vonallánccokat. Az eljárás során a

területeket felépítő töréspontokból kell a vonalakat létrehozni. A valós idejű topológia gondoskodik arról, hogy az érintkező területek duplázott vonalait csak egyszer tárolja.



4-8. Ábra: Vonal-terület topológia a DigiTerra Map alatt

A területekből a vonalas állomány előállítása más szempontból is fontos. A vonalak topológikus előállításával és területépítéssel megoldható a **terület átfedés**, vagy terület fedvényezési művelet. Két terület állomány átfedése (*polygon overlay*) a vonalas állományok előállításával kezdődik. A vonalas állományokat a valós idejű topológiával indokolt összedolgozni. A művelet során számos új zárt alakzat jön létre. Az összedolgozás után a létrejövő zárt alakzatokból területeket kell építeni. Az eljárás befejező fázisában a két forrás állomány területeinek leíró adatait az új területekhez kell **csatolni**. A csatolásban szintén a térbeli indexelés segít. A csatolás során azt kell vizsgálni, hogy az új terület középpontja melyik régi területbe esik az egyik, illetve a másik forrásállomány esetében. Lesznek olyan új területek, amelyekhez két csatolás is történik és lesznek olyanok, amelyekhez csak egy. Ha minden új területet megtartunk, akkor **unióról**, ha csak azokat, amelyekhez két csatolás történt, akkor **metszetről**, és ha csak azokat tartjuk meg, amelyekhez csak az egyik forrás kapcsolódik, akkor **különbségről** beszélünk. A három terület fedvényezési műveletet a 2-4. táblázat is ábrázolja.

Szintén a valós idejű topológiával és terület építéssel összefüggő funkció a **védőzóna** vagy pufferzóna generálás. Az eljárás során az elemi geometriai alakzatok körül megadott távolságra egy vonalláncot hozunk létre. Az elemi geometriai alakzat a pont, a vonalláncok egyetlen szakasza és a területek egyetlen oldala. A létrehozott vonallánc két körívől és a köríveket összekötő két egyenesből áll. A köríveket rendszerint a szerkesztési tolerancia nagyságával megegyező szakaszokkal közelítjük. A létrehozás során ügyelni kell arra, hogy a

vonalláncok körüljárási iránya az óramutató járásával megegyezzen. Ha valamennyi elemi geometriai alakzat körül elkészültek a vonalláncok, akkor elkezdődhet a védőzónák építése. A zárt alakzatból területeket kell létrehozni. A körüljárási iránynak köszönhetően, a belső szigetek a területszámítás alapján könnyen kiszűrhetők.

A valós idejű vonal topológia és vonalláncok alapján a területek építése képezi a *DigiTerra Map* program vektoros moduljának gerincét. A két részletesen ismertetett funkció egy hatékony és könnyen kezelhető térképezési környezetet alkot.

4.6. Erdészeti digitális térképek

A *DigiTerra Map* program tervezésekor az elsődleges cél egy topológikus szerkesztő eszköz létrehozása volt. A másodlagos cél viszont ezzel az eszközzel a digitális erdészeti térképek készítése és karbantartása, valamint erdészeti tematikus térképek előállítása volt. 1996-ban az 1:10'000-es méretarányú erdészeti térképek digitalizálása a *DigiTerra Mérnöki Irodánál* megkezdődött. A **digitális erdészeti térkép formátum** alapjait ekkor dolgoztam ki. A térképek előállítása már az elején is topológikus szerkesztéssel és a vektoros elemekhez kapcsolt leíró adatokkal történt. Nem követtük a Magyarországon még a mai napig is divatos rétegekből, vonalakból és feliratokból álló, topológiát mellőző, leegyszerűsített digitális térképezés sémáját. Tekintettel az évenként aktualizált Országos Erdőállomány Adattárra, a digitális erdészeti térképek egyértelműen geoinformatikai megoldást kívántak.

A formátum az évek során csak kis mértékben módosult. 1998-ban indult az *Állami Erdészeti Szolgálat* szabványosítási törekvése, melybe a *Nyugat-Magyarországi Egyetem Földmérési és Távérzékelési Tanszékét*, s így engem is bevontak. A munka során az 1996-ban lefektetett formátumot tisztáztuk és rögzítettük. 1999-ben utasításként jelent meg a **Digitális Erdészeti Térképek** (*továbbiakban DET*) formátum specifikációja (*ÁESZ 1999*). Ettől az évtől a *DigiTerra* cég és az erdőtervezési irodák is a specifikáció szerint készítették a térképeket.

A *DET* csak elméleti és logikai adatmodellt definiál, ezért bármilyen szoftverkörnyezetben megvalósítható. A formátum leírás felsorolja, csoportosítja az erdészeti térképi objektumokat, valamint megadja a közöttük lévő kapcsolatokat. Az **elméleti adatmodell** az erdészeti üzemi térképen ábrázolt objektumokat jelöli meg.

A logikai adatmodell három objektumtípust definiál. A három típus az objektumok geometriai formája alapján különül el, úgymint pontok, vonalláncok és területek. A **logikai adatmodell** táblázatos formában rögzíti az objektumtípusok tulajdonságait és kapcsolatait (*4-1. Táblázat, 4-2. Táblázat, 4-3. Táblázat*). A logikai adatmodell tesz bizonyos utalásokat a fizikai adatmodellre is. Az enumerációs adattípusok mind előre meghatározott értékeket vehetnek fel. Az enumerációs típusok megnevezéseit és kódjait a formátum tartalmazza.

PONT		
Geometria: geometriai pont koordinátái		
Földrajzi objektumok: alap- és részletpontok, feliratok, szimbólumok, berendezések		
Adatmezők		
Név	Leírás	Típus
SORSZÁM	pont sorszáma	numerikus kulcs
PTÍPUS	pont típusa	enumerációs
FELIRAT	ponthoz tartozó opcionális felirat	szöveges
FELX	felirat beillesztési pontjának x koordinátája	numerikus
FELY	felirat beillesztési pontjának y koordinátája	numerikus
FELSZÖG	felirat szöge K-É irányban	numerikus
DÁTUM	pont utolsó módosításának dátuma	dátum
FORRÁS	adatforrás típusa	enumerációs
Kapcsolatok		
Kapcsolat	Leírás	Kötelező
SORSZÁM – kapcsolt adattábla kulcs	másodlagos adattáblák kapcsolása	nem

4-1. Táblázat: PONT adattípus logikai modellje

VONAL		
Geometria: geometriai vonallánc töréspontjainak koordinátái		
Földrajzi objektumok: határok, utak, vasutak, kerítések, vízrajz, töltés		
Adatmezők		
Név	Leírás	Típus
SORSZÁM	vonallánc sorszáma	numerikus kulcs
VTÍPUS	vonallánc természetes típusa	enumerációs
ADMINHAT	vonallánc adminisztratív határ jellege	enumerációs
ERDHAT	vonallánc erdészeti határ jellege	enumerációs
KAPCS	vonallánc két oldalán lévő területek kapcsolása	enumerációs
BAL	vonallánc bal oldalhoz kapcsolódó terület sorszáma	numerikus kulcs
JOBB	vonallánc jobb oldalhoz kapcsolódó terület sorszáma	numerikus kulcs
DÁTUM	vonallánc utolsó módosításának dátuma	dátum
FORRÁS	adatforrás típusa	enumerációs
Kapcsolatok		
Kapcsolat	Leírás	Kötelező
SORSZÁM – kapcsolt adattábla kulcs	másodlagos adattáblák kapcsolása	nem
BAL – bal oldali terület kapcsolása	vonallánc – terület topológia	igen
JOBB – jobb oldali terület kapcsolása	vonallánc – terület topológia	igen

4-2. Táblázat: VONAL adattípus logikai modellje

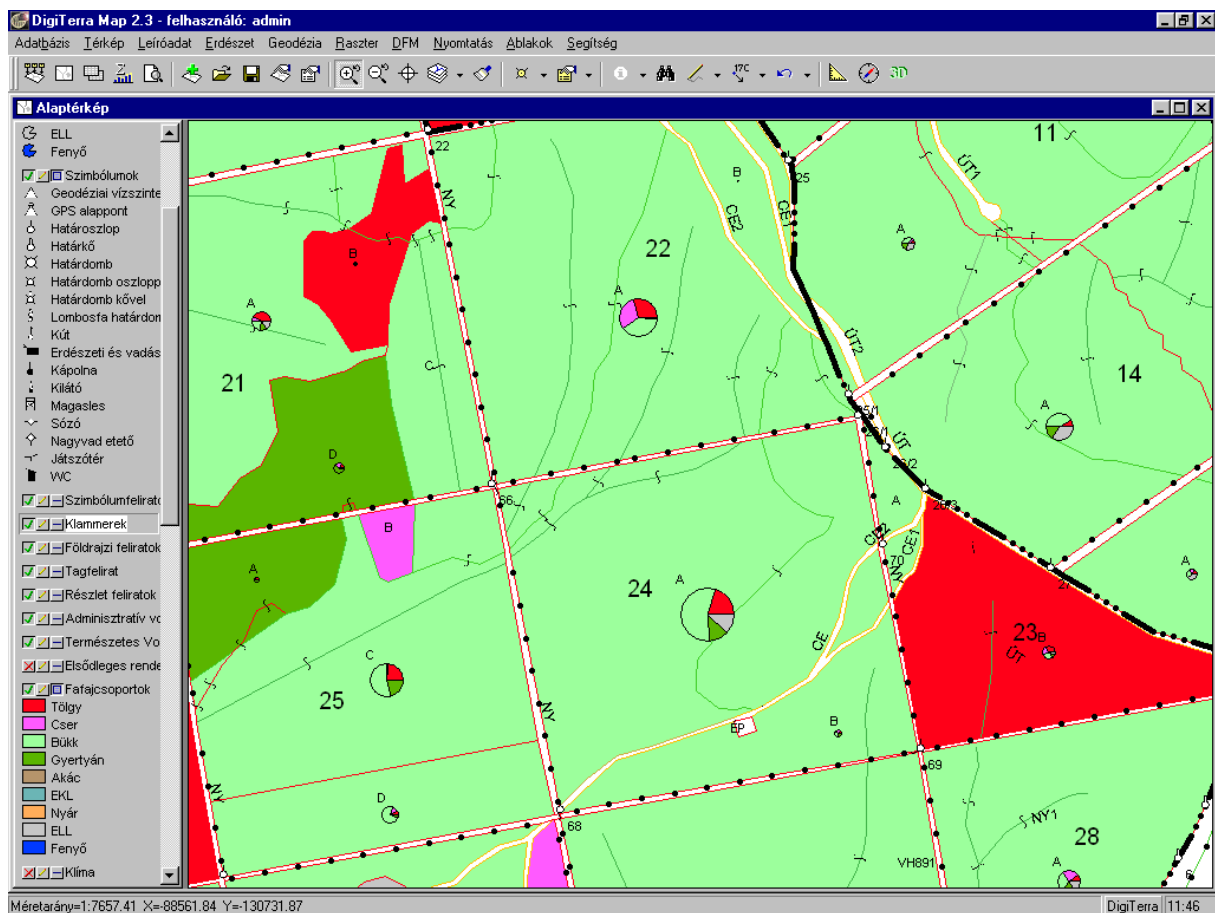
TERÜLET		
Geometria: több poligonból felépülő geometriai terület csúcspontjainak koordinátái		
Földrajzi objektumok: erdőrészletek, földrészletek, épületek		
Adatmezők		
Név	Leírás	Típus
SORSZÁM	terület sorszáma	numerikus kulcs
HELY	erdőrészlet helységkódja	enumerációs kulcs
TAG	erdőrészlet tag száma	numerikus kulcs
RÉSZLET	erdőrészlet részletkódja	enumerációs kulcs
ALRÉSZLET	erdőrészlet alrészlet száma	numerikus kulcs
HRSZ	terület helyrajzi száma	szöveges
MŰVÁG	terület művelési ága	szöveges
TERÜLET	objektum számított területe	numerikus
DÁTUM	terület utolsó módosításának dátuma	dátum
Kapcsolatok		
Kapcsolat	Leírás	Kötelező
SORSZÁM – kapcsolt sorszám	másodlagos adattáblák kapcsolása	nem
HELY – kapcsolt hely	üzemtervi adattáblák kapcsolása	igen
TAG – kapcsolt tag	üzemtervi adattáblák kapcsolása	igen
RÉSZLET – kapcsolt részlet	üzemtervi adattáblák kapcsolása	igen
ALRÉSZLET – kapcsolt alrészlet	üzemtervi adattáblák kapcsolása	igen

4-3. Táblázat: TERÜLET adattípus logikai modellje

A formátum alapján az erdészeti térkép objektumai besorolhatók a fenti három objektumtípus valamelyikéhez. Mindhárom objektumtípushoz egy-egy adatállomány tartozik, amelyekben az objektumok tárolódnak. *DigiTerra* környezetben az erdészeti digitális térkép több **tematikus fedvényből** épül fel. A tematikus fedvény az adatállomány valamilyen szempont szerint leválogatott objektumait tartalmazza. A leválogatást egy relációs kifejezés végzi. A tematikus fedvények egymás után rajzolódnak ki. A fedvényeken belül az objektumok **tematikus osztályokba** sorolhatók. Az osztályba sorolás az objektumok tulajdonságai alapján megy végbe. A tematikus osztályokhoz különböző grafikai beállítások tartoznak, úgymint vonalszín, vonaltípus, vonalvastagság, feliratméret, feliratstílus, feliratszín, kitöltéstípus, kitöltésszín. A tematikus fedvényeket és osztályokat rugalmasságuk miatt számos geoinformatikai rendszer használja. A *DigiTerra* programon belül az erdészeti digitális térképeket a következő tematikus fedvények építik fel (*a sorrend megfelel a kirajzolás sorrendjének*):

- TERÜLET – Erdőrészletek – *színezés a faállománytípus vagy egyéb jellemző szerint*
- VONAL – Természetes vonalak – *a vonaltípus szerint eltérő grafikai megjelenítés*
- VONAL – Adminisztratív vonalak – *község- tag- és egyéb határok megjelenítése*
- PONT – Kapcsolójelek – *elkülönülő erdőterületek összetartozását jelzik*
- PONT – Szimbólumok – *alap- és részletpontok, berendezések szimbólumai*
- PONT – Szimbólumfeliratok – *alap- és részletpontok sorszámai*
- PONT – Erdőtag és erdőrészlet feliratok – *tagok és részletek pozícionált feliratai*
- PONT – Földrajzi feliratok – *földrajzi helyek, belterületek megnevezése*

A digitális erdészeti térképpel szemben támasztott alapvető igény volt az analóg térképhez hasonló kartográfiai megjelenés (4-9. Ábra). A hasonló megjelenést csak saját rajzolófüggvények fejlesztésével tudtuk biztosítani. Külön rajzfüggvény gondoskodik a szimbólumok, a vonalak, a feliratok, a területek és a kapcsolójelek kirajzolásáról. Ezeknek a függvényeknek köszönhető, hogy a *DigiTerra Map* képes az erdészeti térképek minden kartográfiai igényt kielégítő megjelenítésére. A *DigiTerra* és az *Állami Erdészeti Szolgálat* munkatársai hosszú időn keresztül tervezték és javították a szimbólumokat, állították be a vonaltípusokat és a fedvényeket.



4-9. Ábra: Digitális erdészeti térkép részlete.
Bal oldalon látható a térképet felépítő fedvények listája.

A digitális erdészeti térképek előállításának és karbantartásának során a tematikus fedvények kis mértékben eltérnek az előbbi kiosztástól. A kiegészítés elsősorban a térképezést támogató fedvények, rendszerint raszteres fedvények bevonását jelenti. Egy lehetséges tervezési fedvény sorrendet mutat a következő lista:

- Digitális ortofotók, úrfelvételek, ha rendelkezésre áll
- Digitális domborzatmodell, ha rendelkezésre áll
- Szkennelt és transzformált földhivatali térképek, több méretarányban
- Szkennelt és transzformált erdészeti üzemi térképek
- Erdőrészletek, színezés a rendeltetés, a művelési ág vagy egyéb jellemző szerint
- Jogi állapotot tükröző földhivatali térkép, különös tekintettel a községhatárookra
- Korábbi üzemtervi térkép vonalas állománya

- Természetes vonalak, a vonaltípus szerint eltérő grafikai megjelenítés
- Erdőrészlet feliratok

Köztudott, hogy az erdészeti üzemi térképek 1962 óta a magyarországi sztereografikus vetületet és egy speciális szelvényezést használtak (Bácsatyai 1993). A geoinformatika és digitális erdészeti térképek megjelenése előtt az **Egységes Országos Vetületi rendszerre való áttérés** nagyon sok munkával lett volna csak megoldható. A *DigiTerra Map* tetszőleges vetületi rendszerben tud dolgozni és a vetületek közötti átszámításhoz is számos funkciót biztosít. A hazai, a nemzetközi és a történelmi vetületek közötti átszámítást elsősorban hasonlósági transzformációval oldja meg a program (Czimer 1997). A digitális erdészeti térképek előállítás után a vetületváltás gyors és problémamentes volt.

A **szelvényezés** is rugalmasan állítható a programon belül. A jelenleg használt vetületi rendszerek szelvényezéseit támogatja, de ettől eltérő szelvényezés is beállítható. A szelvényezés során a szelvényhatárra eső földrajzi feliratok duplázásáról kellett csak gondoskodni. A szelvényen az örkeresztek, a koordináta háló, a szelvényfeliratok, a koordináta feliratok és a szelvénykeret kirajzolása automatikus. A digitális erdészeti térképek állandó grafikai elemeit sablonok tartalmazzák. A vetületváltást és szelvényezési feladatokat a hatékony digitális térképezési környezet megoldotta. Ez a környezet azonban előtérbe hozott egy kényes kérdést, nevezetesen a **jogi és természetes állapotot** tükröző térkép közötti különbséget.

A program tervezésekor fontosnak tartottam a kartográfiai elemek **méretarányfüggő és független megjelenítését** és bevezettem a térképi egység, a rajzi egység és a kirajzolási méretarány fogalmát. A három jellemző segítségével ugyanabból a térképből tetszőleges méretarányú digitális térképet nyomtathatunk és mindegyik térképen a vonalvastagságok, a feliratok mérete azonos lesz. A méretarányfüggő megjelenítéssel elérhető, hogy a különböző méretarányú térképek nyomtatása más-más adatsűrűséggel történjen. Egy átnézeti térképen csak a községhatárok, a belterületek, a fontosabb utak és az erdőtömbök jelennek meg. Nagyobb méretarányú térképen fokozatosan rajzolódnak ki a nagy adatsűrűségű fedvények. Teljes részletesség elérésekor az átnézeti fedvények automatikusan kikapcsolódnak.

4.7. TIN felületmodellezés

A geoinformatikában gyakori feladat térbeli pontokra egy folytonos **felület illesztése**. A legtöbb földtudomány véletlenszerű, vagy a jellemző földrajzi objektumokhoz igazított adatgyűjtést végez. A szórt mérések földrajzi helyzetét koordináták tárolják. A harmadik koordináta általában a tengerszint feletti magasságot adja meg, de tárolhatja egy növényfaj előfordulási gyakoriságát, a talajvíz aktuális mélységét, az erdő egészségi állapotát, az átlagos hőmérsékletet, de akár a szén-dioxid koncentrációt is. Az előbbi lista is rámutat, hogy a felületillesztést a föld- és természettudományok szinte valamennyi területén alkalmazhatják. A felületillesztésre a geoinformatika gyors és megbízható eljárásokat fejlesztett ki.

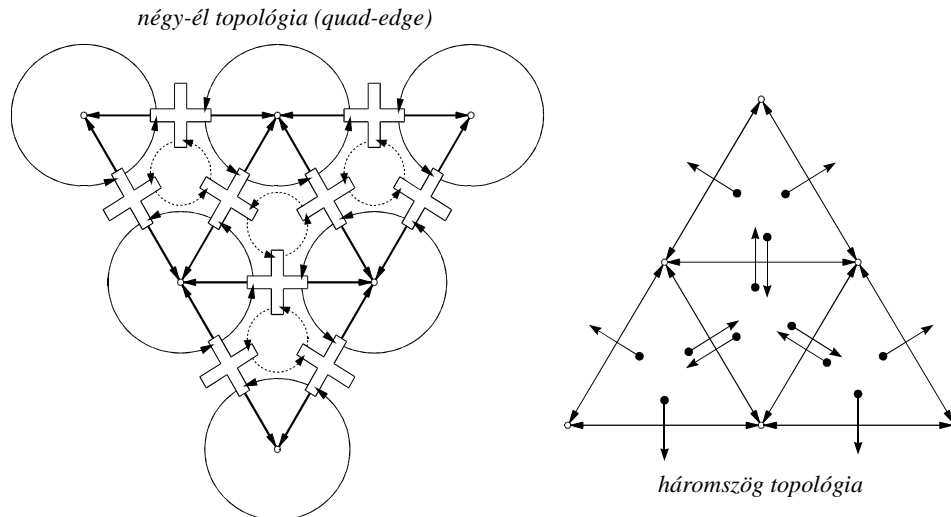
A szórt ponthalmazból előállíthatunk **raszteres felületet** a pontok magassági értékeinek **térbeli interpolációjával**. A térbeli interpolációt minden egyes rasztercella esetén el kell végezni. Az interpoláció alapja a közeli pontok magasságának súlyozása a pontok távolságától függő mennyiséggel. Minél távolabb van a pont az aktuális cellától, annál kisebb súlyt kap. A térbeli interpoláció azonban időigényes eljárás. Az eredmény raszter nem illeszkedik a szórt pontok koordinátaíhoz. Nagy szórással jellemezhető méréseknél a raszteres interpoláció

kifejezetten előnyös (*meteorológia, botanika, hidrológia*). Geodéziai és fotogrammetriai mérések feldolgozásánál viszont alapvető igény, hogy az előállított felület átmenjen a mért pontokon. Ez a követelmény vagy nagyon sűrű raszter interpolációjával vagy vektoros felületmodellezési eljárásokkal lehetséges. A felület egy kétdimenziós skalár mező, mely összetettsége miatt egyetlen függvénnyel nem írható le. A **vektoros felületmodell** véges számú elemi felületekből állítható össze. Az elemi felületeket koordinátaival adott csúcspontokkal és kontrolpontokkal, esetleg paraméterekkel határozzuk meg. Az eljárást a szakirodalom **végeselem módszernek** hívja. A végeselem módszer a mérnöki gyakorlatban igen elterjedt (*Segerlind 1984*).

A pontok közötti területek lefedésére a háromszög, mint a legegyszerűbb felületelem, a legalkalmasabb. A háromszögek csúcspontjai az egyes pontokon nyugszanak. A háromszögek elhelyezésénél feltétel, hogy a pontok konvex burkolóján belüli tartományt átfedés mentesen fedjék le. A háromszögek létrehozását **háromszögelésnek**, triangulációnak vagy háromszög lefedésnek hívjuk. A háromszögelés eredménye egy szabálytalan háromszöghálózat (*Triangulated Irregular Network = TIN*). Optimális háromszöglefedést biztosít a *Delaunay* háromszögelés (*Márkus, B. 1994*). A *Delaunay* háromszöglefedés feltétele, hogy az egyes háromszögek köré írt körön belül nem lehet más pont. Ha a *Delaunay* háromszögek oldalfelező merőlegeseit megrajzoljuk, akkor megkapjuk a *Thiessen*-poligonokat vagy *Voronoi*-diagramokat. A *Delaunay* háromszögelésre számos hatékony módszert dolgoztak ki. Az egyes eljárásokat több tudományos munka is elemzi és összehasonlítja (*Shewchuk 1997*). A eljárások közül jelenleg a leghatékonyabb a *Divide & Conquer* algoritmus és annak változatai. Az eljárás műveletigénye $O(n \log n)$. A *DigiTerra* programon belül is ezt az eljárást használjuk.

A *Divide & Conquer* algoritmust (*oszd meg és hódítsd meg*) Guibas és Stolfi (1985) dolgozta ki. Az algoritmus lényege röviden a következő. A szórt pontokat először rendezni, majd a rendezett ponthalmazt rekurzív módon felezni kell. A felezés addig tart, amíg kettő vagy három pont marad. Két pontra egy háromszög oldal, három pontra pedig egy háromszög illeszthető. Ezután indulhat a felezéssel ellentétes folyamat, az összevonás. Az összevonás során a két halmaz háromszög oldalait és háromszögeit össze kell kapcsolni úgy, hogy a *Delaunay* feltétel teljesüljön. Az összekapcsolás során a halmaz határain lévő oldalak lebomlanak és a két halmaz között új oldalak képződnek. Az összekapcsolást a megfelelő adatstruktúra biztosítja.

Két **adatstruktúra** terjedt el, mindkettő tárolja a topológiát is, azaz a háromszögoldalakat és a háromszögek szomszédsági kapcsolatait (4-10. Ábra). Az adatstruktúrákat szintén Guibas és Stolfi (1985) definiálta. Az első struktúra a **négy-él** (*quad-edge*) névre hallgat. A négy-él topológiája bonyolult, viszont a háromszögelés gyors és tömör eljárásokkal valósítható meg. A geoinformatikában előnyösebb a másik adatstruktúra, a **háromszög topológia** használata. Itt a térbeli kapcsolatok könnyen átláthatók, a tárolás kevesebb helyet igényel, de az eljárás programozása bonyolultabb. A háromszög topológia alapján a felületmodell megjelenítése és elemzése könnyebb.



4-10. Ábra: A négy-él és a háromszög adatstruktúra

DigiTerra környezetben egy hibrid megoldást választottam, mely a gyors és tömör négy-él struktúrával végzi a háromszögelést, majd áttér a háromszög topológiára. A háromszög topológia tárolja a koordinátaival adott pontokat és a háromszögeket. A háromszögek megadása hat mutatóval történik, három mutat a háromszög csúcspontjaira és három a szomszédos háromszögekre. Egy n pontból álló ponthalmaz háromszögelése megközelítőleg $2n$ háromszöget és $3n$ háromszögoldalt eredményez. A pont tárolásához $3 \cdot 4 = 12$ bájt, a négy-él és a háromszög tárolásához $6 \cdot 4 = 24$ bájt szükséges. Ezek alapján a tárigény háromszög topológia esetén $n \cdot 12 + 2n \cdot 24 = 60 \cdot n$ bájt, míg négy-él struktúra esetén $n \cdot 12 + 3n \cdot 24 = 84 \cdot n$ bájt.

A háromszögelés során sokszor felmerül az igény, hogy a háromszögek oldalai kövessék a meglévő geometriai vonalakat. Ekkor **kényszerített háromszöglefedésről** beszélünk. A kényszerítés könnyen megvalósítható egy a pontokhoz rendelt jelző bevezetésével. A háromszögelés előtt a kényszerített vonalakat létre kell hoznunk és a jelzőket be kell állítanunk. A háromszögelés során a kényszerített vonalakon nyugvó oldalakra nem kell a *Delaunay* feltételt megvizsgálni és a vonalakat nem szabad lebontani. A kényszerítés bevezetésével a felület struktúráját visszatükröző háromszögelés érhető el.

A háromszögelés a felületmodellezés első része, mely a pontok közötti térrész lefedéséről gondoskodik. A háromszög csúcspontjainak koordináta alapján egy belső pont magassága legegyszerűbben lineáris interpolációval határozható meg. A háromszögeken belüli lineáris interpoláció viszont egy síklapokból álló, törésekkel rendelkező felületet eredményez. Egy megfelelően sima felület előállításához további jellemzők bevonása szükséges és a sík háromszöglapok helyett magasabb rendű felületeket kell használnunk.

Az elemi felületek meghatározására a következő peremfeltételeket dolgoztam ki:

- Az elemi felületeket a háromszögek területén belül kell felállítani. A felület magassága a csúcspontokban megegyezzen a pont magasságával. A pontok a keresett felület kontrolpontjai legyenek.
- A pontokban a felület első-, és lehetőleg a második parciális deriváltjai is előre meghatározott értékek legyenek. Az első parciális deriváltakat a ponthoz kapcsolódó háromszögek normálvektora alapján számítom. A kapcsolódó háromszögek kijelölését a

háromszög topológia biztosítja. A normálvektorok vízszintes összetevőiből előállított deriváltak súlyozott átlaga adja a pont parciális deriváltjait. A súlyozás a háromszögek nagyságának reciprok értékével történik. A második parciális deriváltak az első parciális deriváltakból számíthatók az előbbi módszerhez hasonlóan.

- Az elemi felületek a háromszögek élein kapcsolódnak. A kapcsolódásnak nullad-, elsőrendű vagy magasabb rendűnek kell lennie. A nulladrendű kapcsolódás azt jelenti, hogy a felületek magassága az éleken megegyezik. Az elsőrendű kapcsolatnál az élek mentén az első parciális deriváltak is megegyeznek, amely törésmentes kapcsolatot eredményez a felületek között. A másodrendű kapcsolatnál a második parciális deriváltak is megegyeznek, amely törésmentes és sima érintkezést jelent. Ezt az utóbbi feltételt, amely az éleken azonos görbületet eredményez, nagyon nehéz biztosítani.
- Az elemi felületek vegyék figyelembe a kényszerített háromszögelés vonalait is. Idomvonalak esetén (*szintvonal, völgyvonal, gerincvonal*) a felületelemek között törésmentes, elsőrendű vagy magasabb rendű kapcsolat legyen. Törésvonalak esetén (*részű, tereplépcső, szakadék, támfal, partvonal*) viszont a kapcsolódó felületelemek között éles törések, nulladrendű érintkezések legyenek.

A parciális deriváltak ismeretében az elemi felületek felírhatók. A szakirodalom rendszerint a *Bézier*-háromszöget javasolja elemi felületnek (*Pfeifer – Pottman 1996*). Más szerzők ettől bizonyos mértékben eltérő módszereket választottak (*Akima 1970, Mélykúti 1993*). A *Bézier* felület közel szabályos háromszögek esetén elsőrendű kapcsolatot biztosít. A *Bézier*-háromszög egy három paraméteres, harmadfokú polinom, mely a *Breinstein* polinomból és a *Bézier*-görbéből vezethető le (*Horváth 1989*). Az i -dik, n -ed fokú, t paraméterű *Breinstein* polinom alakja:

$$b_{i,n}(t) = \frac{n!}{(n-i)!i!} t^i (1-t)^{n-i} \quad t = 0 \dots 1 \quad (4-7)$$

A *Breinstein* polinom megegyezik a binomiális eloszlás sűrűségfüggvényével. Mivel $b_{i,n}(t)$ valószínűségi sűrűségfüggvény, ezért fennáll:

$$\sum_{i=0}^n b_{i,n}(t) = 1 \quad (4-8)$$

Bézier alkalmazta először a *Breinstein* polinomok lineáris kombinációját a kontrolpontokkal adott görbék és felületek illesztésére. Egy n -ed fokú P_i kontrolpontokkal jellemzett *Bézier*-görbe alakja a következő:

$$B_n(t) = \sum_{i=0}^n b_{i,n}(t) P_i \quad (4-9)$$

Az összefüggés alapján egy harmadfokú *Bézier*-görbe egyenlete:

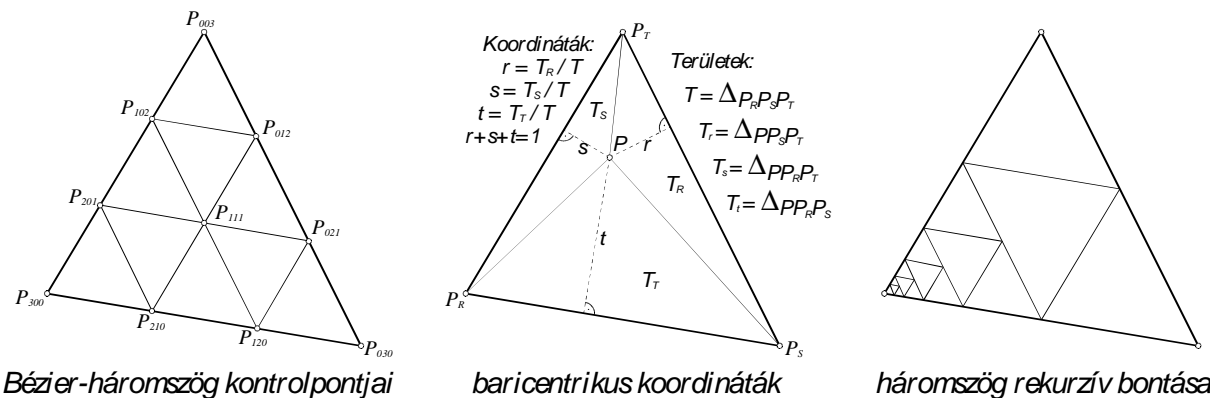
$$B_3(t) = (1-t)^3 P_0 + 3(1-t)^2 t P_1 + 3(1-t) t^2 P_2 + t^3 P_3 \quad (4-10)$$

A *Breinstein* polinomokhoz hasonlóan kell képezni a harmadfokú *Bézier*-háromszöget is (*Pfeifer – Pottman 1996*):

$$T_n(r, s, t) = \sum_{i+j+k=n} \frac{n!}{i!j!k!} r^i s^j t^k P_{ijk} \quad (4-11)$$

ahol: n a Bézier-háromszög fokszáma, jelen esetben $n=3$
 r, s, t a háromszögon belüli P pont baricentrikus koordinátái
 i, j, k indexek, értékük a következő lehet: 0, 1, 2, 3
 P_{ijk} Bézier-háromszöget kifesztítő kontrolpontok (4-11. Ábra)

A P pont a háromszöget három segédháromszögre bontja. A baricentrikus koordináták a segédháromszögek és a háromszög területének hányadosából számíthatók. Az élek harmadában felállított kontrolpontok magasságai a háromszög csúcspontjainak magasságából és a csúcspontokban meghatározott parciális deriváltakból állíthatók elő. A parciális deriváltak és az oldalhosszak harmadának szorzata mind egy-egy vektort alkot, amely kijelöli a kontrolpontok térbeli helyzetét. A középső kontrolpont magassága a belső kontrolpontokból számítható egyszerű átlaggal vagy egy másodfokú görbével. Törésvonalak esetén a parciális deriváltak számítása eltér. Ilyenkor a törésvonalra eső oldal mentén a parciális deriváltakat a sík háromszögből kell előállítani.



4-11. Ábra: Bézier-háromszögek, baricentrikus koordináták és a rekurzív bontás

Egy Bézier-háromszögon belüli pont magasságának számítása a következő kóddal történik:

```
// p1,p2,p3 -> x,y,z : a háromszög csúcspontjainak koordinátái
// p1,p2,p3 -> dx,dy : háromszög csúcspontjainak parciális deriváltjai
// r,s,t : keresett magasságú pont baricentrikus koordinátái

// az élek harmadában felállított kontrolpontok magasságai
P210 = ((p2->x - p1->x)*p1->dx + (p2->y - p1->y)*p1->dy)/3 + p1->z;
P201 = ((p3->x - p1->x)*p1->dx + (p3->y - p1->y)*p1->dy)/3 + p1->z;
P120 = ((p1->x - p2->x)*p2->dx + (p1->y - p2->y)*p2->dy)/3 + p2->z;
P021 = ((p3->x - p2->x)*p2->dx + (p3->y - p2->y)*p2->dy)/3 + p2->z;
P102 = ((p1->x - p3->x)*p3->dx + (p1->y - p3->y)*p3->dy)/3 + p3->z;
P012 = ((p2->x - p3->x)*p3->dx + (p2->y - p3->y)*p3->dy)/3 + p3->z;
P111 = (P210 + P201 + P120 + P021 + P102 + P012)/6;

// sík háromszöglap esetén lineáris interpoláció
if(flat) return p1->z * r + p2->z * s + p3->z * t;

// harmadfokú Bézier-háromszög interpoláció
return p1->z*r*r*r + p2->z*s*s*s + p3->z*t*t*t +
    3 * ( P210*r*r*s + P201*r*r*t + P120*r*s*s + P021*s*s*t +
    P102*r*t*t + P012*s*t*t ) + 6 * P111*r*s*t;
```

A szabályostól eltérő háromszögeknél a *Bézier*-háromszög nem garantálja az elsőrendű kapcsolatot, ezért hajlamos a kilengésekre és a szintvonalak sajátos hullámzó alakot vesznek fel. A kilengések elkerülése érdekében egy új módszert dolgoztam ki, melyet a **háromszögek rekurzív bontásával** érek el (4-11. Ábra). A bontás során a háromszög csúcspontjai és oldalfelező pontjai között négy új háromszöget hozok létre. A bontást tetszőleges szintig lehet folytatni. Az oldalfelező pontok magasságait és parciális deriváltjait az oldal két végpontjának magasságaiból és parciális deriváltjaiból számítom egy harmadfokú *Bézier*-görbével. Mivel az oldalfelező pont meghatározásában csak az oldal két végpontja vesz részt, ezért az elsőrendű érintkezés garantált. Az oldalfelező pont jellemzőinek számítása egyszerű képletekkel történik:

```
// p1,p2 -> x,y,z : az él két végpontjának koordinátái
// p1,p2,p3 -> dx,dy : az él két végpontjának parciális deriváltjai

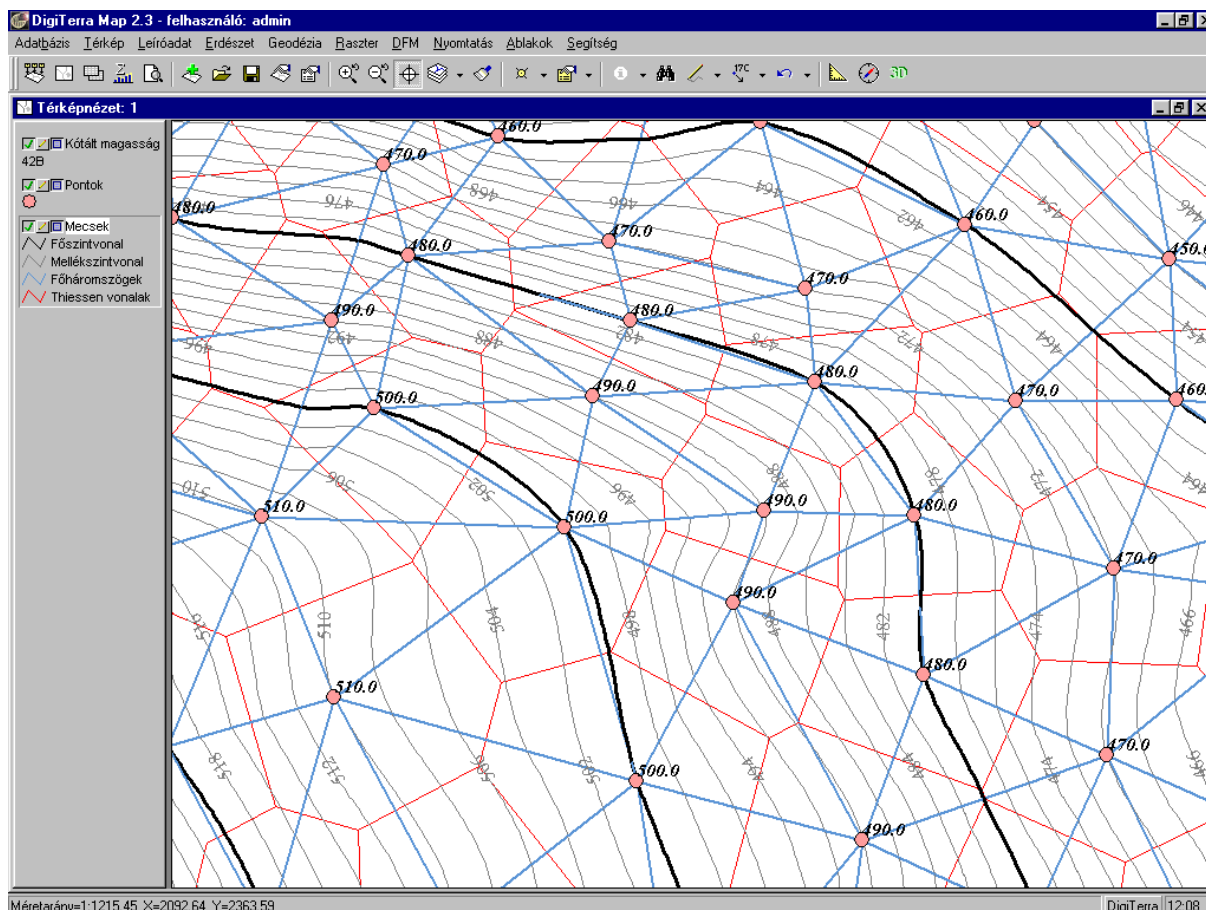
// az oldalfelező pont koordinátáinak meghatározása
P21 = ((p2->x - p1->x)*p1->dx + (p2->y - p1->y)*p1->dy)/3 + p1->z;
P12 = ((p1->x - p2->x)*p2->dx + (p1->y - p2->y)*p2->dy)/3 + p2->z;
p->x = (p1->x + p2->x)*0.5;
p->y = (p1->y + p2->y)*0.5;
p->z = (p1->z + p2->z + 3*(P21+P12))/8; // Bézier-görbe t=0.5 esetén

// az oldalfelező pont parciális deriváltjai
p->dx = 3*(P21 - P12) / (p2->x - p1->x);
p->dy = 3*(P21 - P12) / (p2->y - p1->y);
```

A magasságok és parciális deriváltak a számítás során csillapíthatók, ami a szélsőséges értékek kiszűrésére szolgál. A rekurzív bontást mindig csak egy bizonyos szintig kell elvégezni. Megjelenítéskor ezt a szintet a kirajzolás felbontása határozza meg. Magasság meghatározáskor a bontást egy megkívánt szintig kell csak folytatni. Három-négy rekurzív bontás már megfelelő pontosságot eredményez. A kívánt pontosságot adó szint elérése után az interpoláció már lineáris. Általában a következő magasság meghatározás is az előző rekurzív bontás háromszögén belül történik, ezért a bontás optimalizálható.

Az elemi felületek felírása után a **felületmodellezéshez** minden rendelkezésünkre áll. A legtöbb eljárás az elemi felületek egészét vagy annak nagyobb csoportját érinti. Ilyenek a szintvonalazás, a magasság, a lejtés vagy a kitétség szerinti színezés, a térbeli megjelenítés, a földtömeg számítás. Ezek az eljárások mind könnyedén kivitelezhetők a háromszögek rekurzív bontásával. A rekurzív bontás valamivel több számítással *Bézier*-háromszögek esetén is elvégezhető. A bontás során elérhető egy olyan szint, amelyen mint sík háromszöglapon a kívánt elemzések könnyedén végrehajthatók.

A **szintvonalazás** a megfelelő rekurzív bontás után a sík háromszöglapon belül történik. A sík lap és a szintvonalak metszéspontjának iránya könnyen kifejezhető a sík normálvektorából. A szintvonalazást a háromszög legalacsonyabb és legmagasabb pontja közötti oldalon kell elvégezni. A legalacsonyabb pontból kiindulva a szintvonalak egységnyi távolságra indulnak ki és a szomszédos háromszögoldalig tartanak. A fő- és mellékszintvonalak feliratozására két raszteres hálót állítottam fel, mely megfelelő sűrűségű szintvonalfeliratokat biztosít. Ha az adott raszteren belül nem rajzoltam ki még szintvonal feliratot, akkor szintvonal rajzolásakor a pozíciót bejegyzem. Az összes szintvonal kirajzolása után következik a feliratok kiírása a bejegyzett pozíciókba (4-12. Ábra).



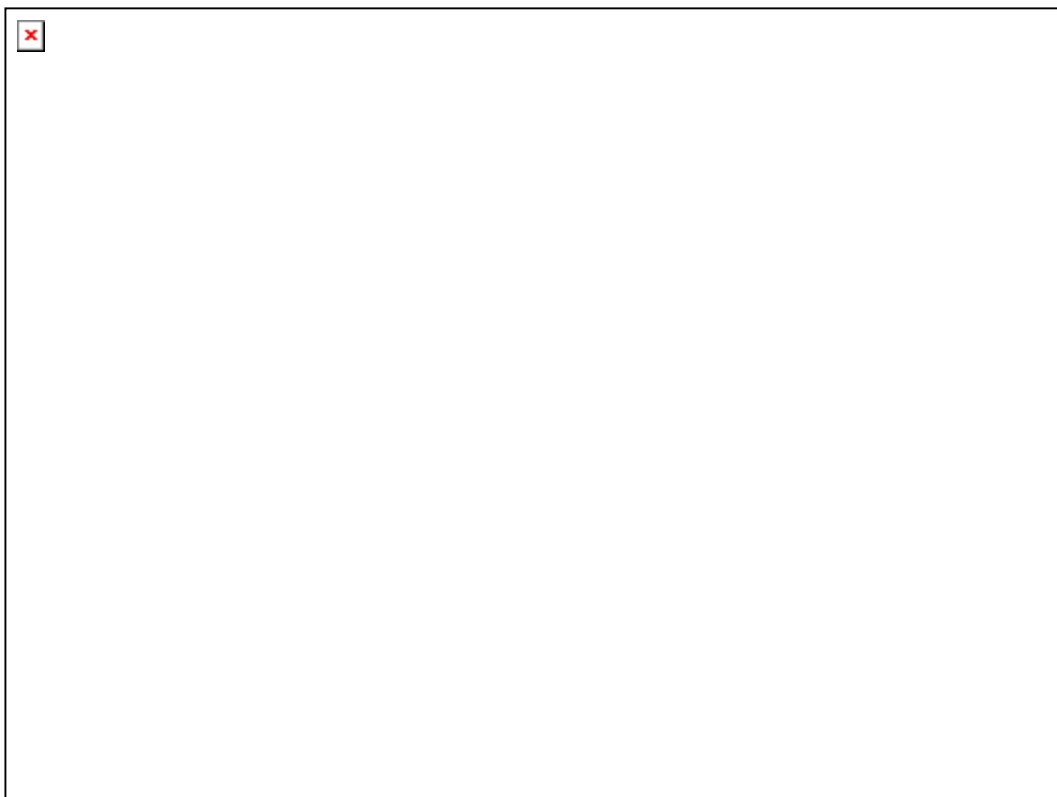
4-12. Ábra: A DigiTerra Map a háromszög topológia alapján valós időben rajzolja a szintvonalakat, a feliratokat, a háromszögeket és a Thiessen-poligonokat

A magasság, a lejtés és a kitétség szerint **színezett felületek** megjelenítése egy színskála kiválasztásával kezdődik. A rekurzív bontással kapott háromszögek csúcspontjaihoz a magasság vagy a (3-2) képletekkel kapott értékek alapján hozzárendeljük a színskála valamelyik színértékét. A háromszögek színezése *Gouraud* árnyékolással történhet, mely a csúcspontok színértékeinek először az élek mentén, majd az élek közötti lineáris interpolációját jelenti (4-13. Ábra, 4-14. Ábra).

A **térbeli megjelenítés** is a rekurzív bontás végén kapott háromszögeket dolgozza fel. A megjelenítéshez a *DigiTerra Map* program a *Silicon Graphic* cég által kifejlesztett és a *Microsoft Windows* operációs rendszerben is elérhető *OpenGL* grafikus nyelvet hívja segítségül (*Microsoft 1998*). Az *OpenGL* nyelv kihasználja a hardver nyújtotta lehetőségeket, gondoskodik a háromszögek térbeli leképezéséről, színezéséről és árnyékolásáról. Az *OpenGL* eljárásai képesek a háromszögeket valamilyen raszteres textúrával is kitölteni, így a térbeli felületekre úrfelvételek, ortofotók húzhatók. A térbeli felületeken megjelenhetnek a vektoros vonalak, szimbólumok és feliratok is (4-15. Ábra). A térbeli megjelenítés történhet anaglif technikával.



4-13. Ábra: TIN alapján készített, magassági színezésű terepmodell perspektív képe



4-14. Ábra: A kitettség alapján valós időben színezett felületmodell perspektív képe



4-15. Ábra: Térbeli megjelenítési lehetőségek a DigiTerra Map programban

A **földtömegszámítás** a felület alatti térrész térfogatát határozza meg. A rekurzív bontással a háromszög alapú hasábok térfogata könnyedén számítható és összesíthető. Felületek közötti térrész a közös területen belüli térfogatok különbségeként számítható.

Az **egyedi magasságértékek** lekérdezése vagy a metszetkészítés a *Bézier*-háromszögek esetében valamivel gyorsabb a rekurzív bontásnál, de ezek a számítások kisebb számban jelentkeznek. Egy pont magasságának meghatározáshoz ki kell keresnünk azt a háromszöget, amely a pontot tartalmazza. A keresés bármely háromszögből kiindulhat. A keresést a kiindulási háromszögből abba az irányba kell folytatni, amely háromszögoldalon túl a pont található. A továbblépést a pont irányába a háromszög topológia jelöli ki. Amikor eljutunk a pontot tartalmazó háromszögbe, akkor kezdődhet a *Bézier*-háromszög paramétereinek meghatározása, illetve a rekurzív bontás. Rendszerint a következő pont is a kikeresett háromszögön belül van, ezért a magasság meghatározás gyorsítható.

5. Összefoglalás

5.1. Előzmények, célkitűzések

A geoinformatika, a földrajzi információk kezelésének tudománya egyre nagyobb jelentőséggel bír mind a kutatás, mind a gazdaság területén. Jelentősége abban rejlik, hogy az információk 80%-a földrajzi helyhez köthető. A geoinformatika a természeti erőforrások, így az erdők térképezésében és kezelésében egy hatékony eszközt képvisel. A geoinformatika kapcsolatot teremt az egyes tudomány- és szakterületek között azzal, hogy egységesíti az adatok kezelését. A geoinformatika szerves része korunk dinamikusan fejlődő iparágának, az informatikának.

A geoinformatika erdészeti alkalmazása Magyarországon az 1990-es évek elején kezdődött. Az Állami Erdészeti Szolgálatnál és a Soproni Egyetemen folytak ilyen jellegű kutatások. Mindegyik esetben maradtak megoldatlan kérdések. Egyik rendszer sem jutott el az országos gyakorlati alkalmazás szintjére. Az erdészeti térképezés, az erdőállományok adatbázisának kezelése viszont egyre inkább egy geoinformatikai megoldásra várt.

A dolgozat egy geoinformatikai rendszer fejlesztéséről szól. A rendszer megszületését a magyar erdészeti szakma hívta életre. A rendszer elkészítése, mint gyakorlati célkitűzés, egyben tudományos célkitűzés is volt. A fejlesztés nyolc éve alatt számos tudományos eredménnyel bíró algoritmus és szoftver technológia született.

A kutatás a szerző diplomatervéből indult 1994-ben. A diplomaterv a III. Térinformatika a Felsőoktatásban Szimpóziumon díjat nyert. 1996-ban számos algoritmus beérett, és megszületethetett egy digitális fotogrammetriai, egy digitális képfeldolgozó és egy térképező szoftver. A szerző a programokat és a velük készült alkalmazásokat számos hazai és külföldi konferencián mutatta be. 1997-ben elindult az erdészeti üzemi térképek digitalizálása. 1998-ban megtörtént a különálló programok összekapcsolása, mely a következő években újabb modulokkal bővült. 1999-re elkészült a Digitális Erdészeti Térképek (DET) formátum leírása, melynek kidolgozásában a szerző is közreműködött.

5.2. Kutatási módszerek

A kutatási téma meghatározása a geoinformatika adott területének kiválasztását jelentette. Gyakori volt az igények szerinti témaválasztás is. Ezután következett a nyomtatott és az elektronikus szakirodalom tanulmányozása. A szakirodalom alapos vizsgálatát az algoritmustervezés, programozás, tesztelés, majd a többszöri javítás követte. Az egyes algoritmusokból programmodulok, végül programok épültek fel. A modulokat már tényleges gyakorlati alkalmazásokon lehetett kipróbálni.

5.3. Tudományos eredmények

Gyakorlati és tudományos szempontból a legfontosabb eredmény a *DigiTerra Map* geoinformatikai szoftver elkészítése. A gyakorlat részéről ez egy magas szinten integrált, mégis könnyen kezelhető magyar nyelvű program megszületését jelenti. A szoftver eredményességét az erdészeti gyakorlati alkalmazások sora és az ország 60%-áról készített digitális erdészeti térképek igazolják. A programot az Állami Erdészeti Szolgálat, geodéziai cégek (*FŐMI, Geodézia Rt, megyei földhivatalok*), tizenöt Erdészeti Részvénytársaság, egyetemek és kutatóintézetek használják. Ez mintegy 300 telepített programot jelent.

A program tudományos szempontból több új algoritmust és technológiát vonultat fel. A nyolc éves munka jelentős részét az algoritmus fejlesztés tette ki. Az algoritmusok a nemzetközi viszonylatban is a legújabb kutatási területeken születtek. Az eredményeket hazai és külföldi konferenciákon mutatta be a szerző.

Raszter feldolgozás

Geoinformatikai rendszerekben a raszteres adatok megjelenítése, feldolgozása egyre nagyobb szerephez jut. A *DigiTerra* program raszteres moduljának fejlesztése során számos szoftvertechnológia és új eljárás született. A raszteres feldolgozás vázát egy valós idejű algoritmikus képfeldolgozó alkotja, mely egy hatékony szoftvertechnológia. A képfeldolgozóval az elemzések valós időben készülnek el. Ez a technológia képes helyettesíteni egy bonyolult, több menüpontból álló képfeldolgozó programot. A technológiához szorosan kapcsolódik egy új raszter formátum kidolgozása is, mely piramis és blokk technikát használ. A szerző kidolgozta a helytakarékos piramis technikát és alkalmazta a *Wavelet*-tömörítést az új formátumon.

Nagyméretű raszteres felületmodellek gyors interpolációjára a dolgozat bemutat egy piramis interpolációnak nevezett módszert. A digitális terepmodellek láthatósági, hidrológiai elemzéséhez használt eljárásokat a szerző korszerűsítette, a hidrológiai elemzéseket egy algoritmusra fűzte fel. A hidrológiai elemzések vízösszefolyási térkép készítésére, vízgyűjtők elkülönítésére, árvíz- és gátmodellezésre használhatók.

A dolgozat ismerteti egy erdészeti és környezetvédelmi jelentőséggel bíró terjedési modellt, mely erdőtüzek, víz és levegőszennyezések modellezésére használható. A raszteres feldolgozás részét képezi a raszteres állományok vektorizálása, melyhez saját konverziós és utófeldolgozó eljárások születtek.

Középpontos vetítésű felvételek tájékozása, feldolgozása

A raszteres adatok jelentős részét távérzékelési módszerekkel állítják elő. A távérzékelte felvételek túlnyomó többsége középpontos vetítéssel készül. A felvételek feldolgozásához elengedhetetlen a felvételi helyzet visszaállítása, vagyis a felvételek tájékozása.

A dolgozat a digitális felvételek tájékozását új megközelítésben tárgyalja. Számos fejezet foglalkozik a tájékozási munka automatizálásával. A felvételi tömbök tájékozásához a szerző a sugárnyaláb kiegyenlítést használja, melyhez új megoldási módszereket készített. A tájékozás után három nagyobb fejezet foglalkozik a felvételek feldolgozásával.

Az első a digitális ortofotó készítés új eljárásait ismerteti. Az ortofotó készítés a középpontos vetítésű felvételtől merőleges vetítésűt állít elő digitális felületmodell segítségével. Új módszernek a felvételi tömb együttes transzformálása, a blokk technika bevezetése, a kompetencia és színegyensúly térképek kidolgozása tekinthető. Utóbbi kettővel a szerző az átalakítás alatt mozaikolja a digitális felvételeket.

A második feldolgozási módszer a térbeli kiértékelés megvalósítása szoftveres környezetben. A térbeli kiértékelést a szerző két síkbeli irányzásra vezette vissza, melyből a másodikat automatizálta. A kiértékelés nem igényel térbeli szemléletet, ezért bármilyen számítógépen megvalósítható, mégis pixel alatti pontosságot biztosít. A kiértékelés a felvételi tömb bármely két vagy több felvételének átfedő részén végrehajtható.

A harmadik feldolgozási eljárás az automatikus felületmodell előállítás. A szerző a kevésbé kutatott ortofotó térben történő terület alapú egyeztetést dolgozta ki. A módszer nagyon sűrű felületmodellt, globális ortofotót és korrelációs térképet szolgáltat a felvételi tömb egészéről. Az eljárásban a blokkosítás, a blokk egészének magassági javítása, az ortofotó előállítás és a korreláció számítás gyorsítása, a gyenge korrelációjú területek kizárása, a meglévő mérések bevonása és a piramis interpoláció tekinthető új megoldásnak.

Adatbázis-kezelés, térbeli indexelés

Geoinformatikai rendszerek másik elterjedt adatmodellje a vektoros adatmodell. Az adatmodellben a földrajzi objektumokat geometriai alakzatok és a hozzájuk kapcsolt leíró adatok írják le. Az objektumok közötti térbeli kapcsolatokról, a topológiáról külön kell gondoskodni.

Az adatbázisok tervezése és kezelése egy rendkívül összetett feladatkör. A dolgozat ismertet egy vizuális adatmodellező eszközt, mely alkalmazások tervezésére is használható. A szerző kidolgozott egy központi memóriában működő adatbázis-kezelőt. Az adatbázis-kezelés alapját az objektumrelációs modellt képezi. A közeljövőben a memória alapú adatbázis-kezelés egyre nagyobb szerephez jut.

Egy geoinformatikai rendszernek a leíró adatokon kívül a térbeli adatokat is kezelnie kell. A térbeli adatok hatékony feldolgozása térbeli indexeléssel lehetséges. A dolgozat bemutat egy új térbeli indexelési módszert. Az indexelés bármilyen dimenziójú és tetszőleges kiterjedésű geometriai elem esetén használható. Az eljárás további előnye, hogy minden adatbázisban megvalósítható, mivel az indexek tárolása nem külön állományokban, hanem a rekordon belül történik.

Topológia, digitális térképek

A vektoros adatoknál a topológia kialakítása alapvető feladat. A topológia figyelembe vételével a vektoros adatokon végzett műveletek egyszerűsíthetők és felgyorsíthatók. A térbeli kapcsolatok kialakítása hozzájárul a konzisztens és hibamentes vektoros állományok létrehozásához.

A szerző ismertet egy térképező eszközt, mely valós időben alakítja ki a topológiát, így a szerkesztési hibák azonnal látszanak. A javítások, a területek építése és aktualizálása felgyorsul. A valós idejű topológia hálózat elemzésre, térképek generalizálására, terület elszámolásra, fedvényezési műveletekre és védőzóna generálásra is használható.

A digitális erdészeti térképek készítése is a valós idejű topológián alapul. A szerző a digitális erdészeti térkép formátum alapjait 1996-ban dolgozta ki. 1998-ban indult az Állami Erdészeti Szolgálat szabványosítási törekvése, melybe a szerző is bekapcsolódott. A Digitális Erdészeti Térképek (DET) formátum leírása 1999-ben készült el. A dolgozat bemutatja a digitális erdészeti térképek felépítését, fedvény kiosztását. A digitális erdészeti térképek új vetületi rendszere az Egységes Országos Vetület lett. Az áttérésben a *DigiTerra Map* program segített. A program képes a digitális erdészeti térképek minden kartográfiai igényt kielégítő megjelenítésére.

Felületmodellezés

A geoinformatikában gyakori feladat térbeli pontokra egy folytonos felület illesztése. A szerző a szabálytalan háromszöghálózat alapú felületmodellezéshez új módszereket dolgozott ki. A dolgozat ismerteti a kényszerített háromszögelést és az elemi felületek illesztésének peremfeltételeit, amely alkalmas magassági pontokkal, törés- és idomvonalakkal definiált felületek rekonstrukciójára. A szerző bemutatja a rekurzív háromszögbontási algoritmust, mely képes a szintvonalak, a lejtőkategória-, kitettség- és megvilágítás térképek valós idejű előállítására. A dolgozat ismerteti a térbeli megjelenítés, a földtömegszámítás és a metszetkészítés lehetőségeit is.

5.4. Tézisek

1. Elkészült egy valós idejű algoritmikus képfeldolgozó, melyhez kapcsolódik egy új raszteres formátum. A képfeldolgozó tartalmaz egy új interpolációs módszert, egy új terjedési modellt, illetve javított láthatósági, hidrológiai és vektorizálási algoritmusokat.
2. Digitális felvételek, felvételi tömbök tájékozását új megközelítésben tárgyalja a dolgozat. A felvételek feldolgozásában a következő területeken születtek új megoldások: ortofotó készítés, térbeli kiértékelés és automatikus felület-modell előállítás.
3. Elkészült egy központi memóriában működő adatbázis-kezelő, mely objektum-relációs modellt használ. Az adatbázis-kezeléshez tartozik egy új térbeli indexelési eljárás is.
4. A dolgozat ismerteti egy új valós idejű topológikus térképező algoritmust. A digitális erdészeti térképek is ezzel az eljárással készültek.

A felületmodellezés területén előrelépés a háromszögelés és a peremfeltételek összehangolása, valamint a rekurzív háromszögbontási algoritmus, mely valós idejű feldolgozást biztosít.

5.5. Eredmények hasznosítása, további teendők

A tudományos munkát kézzel fogható gyakorlati eredmények sora kíséri. A fejlesztés gyakorlati célkitűzését, a *DigiTerra Map* programot a szakma használja. A szerző további teendői közé tartozik az eljárások és technológiák széleskörű publikálása és továbbfejlesztése. A raszteres eljárások, a digitális fotogrammetria, a felületmodellezés területén még számos kutatási kihívás vár.

A geoinformatikai rendszerek jelenlegi fejlődését az adatbázis-kezelő rendszerek és a számítógépes hálózatok, az Internet fejlődése határozza meg. A jövő geoinformatikai rendszere egy kliens böngésző-elemző program lesz, amelybe az Interneten keresztül töltődnek le a komponensek, az adatok, esetleg az eredmények. A további kutatásoknak figyelembe kell venni ezeket a fejlődési irányokat.

6. Irodalomjegyzék

6.1. Nyomtatott irodalom

- Akima, H.** (1970): A New Method of Interpolation and Smooth Curve Fitting Based on Local Procedures, *Journal of the Association for Computing Machinery, Vol.17. No.4. 589-602. October 1970.*
- Állami Erdészeti Szolgálat** (1999): Digitális Erdészeti Térképek (DET) formátum specifikációja, Állami Erdészeti Szolgálat, Végrehajtási Utasítás
- Bácsatyai, L. – Czímber, K. – Király G.** (2001): A digitális fotogrammetria újabb eredményei a Nyugat-Magyarországi Egyetem Földmérési és Távérzékelési Tanszékén, *Geomatikai Közlemények IV. Sopron. 83-102.*
- Bácsatyai, L.** (1993): Magyarországi Vetületek, Mezőgazdasági Szaktudás Kiadó
- Baillard, C. – Dissard, O.** (2000): A Stereo Matching Algorithm for Urban Digital Elevation Models, *Photogrammetric Engineering & Remote Sensing, September 2000, 1119-1128.*
- Bidló, A. – Czímber, K. – Juhász, J. – Kovács, G.** (1998). Kecskeméti arborétum talajnyilvántartási rendszere, *Erdészeti Lapok, 1998 május.*
- Chester, C. S.** (editor) (1980): Manual of Photogrammetry, *Fourth Edition, American Society of Photogrammetry, 1980.*
- Czímber, K.** (1994) Digitális felületi modellek az erdészeti tervezésben. *Diplomaterv, Sopron, 1994.*
- Czímber, K.** (1994) Digitális felületi modellek erdészeti alkalmazása. *III. Térinformatika a felsőoktatásban szimpózium kiadványa, Budapest, 61-63.*
- Czímber, K. – Márkus, I.** (1997): Erdőterületek térképezése és monitorozása távérzékelési módszerek és GIS alkalmazásával, *VII. Földfotó Szeminárium kiadványa, Budapest, 5 oldal*
- Czímber, K.** (1997): Erdőterületek térképezése és monitorozása távérzékelési módszerek és GIS alkalmazásával, *VII. Földfotó Szeminárium kiadványa, Budapest, 5 oldal*
- Czímber, K.** (1997): Geoinformatika. *Egyetemi jegyzet. Soproni Egyetem, 113 oldal*
- Csornai, G. – Dalia, O.** (1991): Távérzékelés, *Jegyzet, Székesfehérvár*
- Daubechies, I.** (1988): *Comm. Pure Appl. Math., 41, 909-996.*
- Detrekői, Á. – Szabó, Gy.** (1995): Bevezetés a térinformatikába, *Nemzeti Tankönyvkiadó, Budapest*
- ER Mapper** (1998): ER Mapper 6.0 User Guide, *Earth Resource Mapping Pty Ltd. 922 oldal*
- Guibas, L. J. – Stolfi, J.** (1985): Primitives for the Manipulation of General Subdivisions and the Computation of Voronoi Diagrams, *ACM Transactions on Graphics 4(2), April 1985, 74-123 pages*
- Horváth, I.** (szerkesztő) (1989): CAD technológiai füzetek, *4. Füzet, Országos Műszaki Információs Központ és Könyvtár, Budapest*

- Hunting Technical Service** (1995): Introduction to Digital Image Processing, *Hunting Technical Service, London*
- Kenedy, P. J. – Karteris, M.** (editors) (1993): Satellite Technology and GIS for Mediterranean Forest Mapping and Fire Management, *Thessaloniki, Greece, 513 oldal*
- Kraus, K.** (1997): Fernerkundung I-II., *Ferd. Dümmler Verlag, Bonn*
- Kraus, K.** (1997): Photogrammetry II., *Ferd. Dümmler Verlag, Bonn*
- Kraus, K.** (1998): Fotogrammetria I., Tertia Kiadó, Budapest
- Márkus, B.** (szerkesztő) (1994): NCGIA Core Curriculum, *I-IV kötet, Magyar kiadás, Székesfehérvár*
- Márkus, I. – Czimber, K. és mások** (1996): MARS & Environment Related Applications, Forest Ecosystems Mapping, *Final Report. PHARE Programme. 135 oldal*
- Márkus, I. – Czimber, K. és mások** (1999): Development of GIS of Fertő-Hanság National Park and Szigetköz Land Protection District, *Preliminary and Final Reports. PHARE-CBC Programme, 145 oldal*
- Mélykúti, G.** (1995): A digitális terepmodellezés és a GIS kapcsolata, *Kandidátusi értekezés. Budapest*
- Pfeifer, N. – Pottman, H.** (1996): Surface Models on the Basis of a Triangular Mesh – Surface Reconstruction, *International Archives of Photogrammetry and Remote Sensing, Vol.31. No.3. 638-643.*
- Platt, D. S.** (2001): Bemutatkozik a Microsoft .Net, Szak Kiadó, Bicske
- Richards, J. A.** (1986): Remote Sensing Digital Image Analysis, *Spinger Verlag*
- Segerlind, L. J.** (1984): Applied Finite Element Analysis, *2nd edition, John Wiley, New York*
- Shewchuk, J. R.** (1997): Engineering a 2D Quality Mesh Generator and Delaunay Triangulator, *Program description, Carnegie Mellon University, Pittsburgh*
- Ullmann, J. D. – Widom, J.** (1998): Adatbázis rendszerek, *Panem Könykiadó, Budapest, 503 oldal*
- Wiman, H.** (1998): Automatic Generation of Digital Surface Models Through Matching in Object Space, *Photogrammetric Record 16(91), 1998.*
- Závoti, J.** (1995): Wavelet-transzformáció a térinformatikai adatrendszerben, *Geodézia és Kartográfia, 4.szám . 197-200. 1995.*
- Zhang, T. Y. – Suen, C. Y.** (1984): A Fast Parallel Algorithm for Thinning Digital Patterns, *Comm. ACM, vol. 27, no. 3, 236-239.*

6.2. Elektronikus irodalom

- Arge, L. – Chase, J. – Halpin, P. – Toma, L. – Vitter, J. – Urban, D. – Wickremesinghe, R.** (1994): Flow Computation on Massive Grid Terrains, *NEC Research Index, iteseer.nj.nec.com*
- Böhm, C. – Brechtold, S. – Kriegel, H. P. – Michel, U.** (1998): Multidimensional Index Structures in Relational Databases, *NEC Research Index, iteseer.nj.nec.com*

- Burt, P. J. – Adelson, E. H.** (1984): A Multiresolution Spline With Application to Image Mosaics, *NEC Research Index*, iteseer.nj.nec.com
- Collins, R. – Hanson, A. – Riseman, E. – Schultz, H.** (1995): Automatic Extraction of Buildings and Terrain from Aerial Images, *NEC Research Index*, iteseer.nj.nec.com
- Czimer, K.** (2000): Geoinformatika, *Elektronikus jegyzet, Nyugat-Magyarországi Egyetem*, geo.eke.hu/geoinfo
- Davis, J. R.** (1998): The TimesTen Main-Memory Data Manager: Meeting the Application Performance, *DataBase Associates International, Inc.*, iteseer.nj.nec.com
- ERDAS Inc.** (1999): Erdas Field Guide, *Fifth Edition, Revised and Expanded*, *ERDAS Inc. Atlanta, Georgia*, www.erdas.com
- ESRI Inc.** (1998): ESRI Shape File Technical Description, *Environmental Systems Research Institute, Inc.*, www.esri.com
- Guttman, A.** (1984): R-Trees – A Dynamic Index Structure for Spatial Searching, *NEC Research Index*, iteseer.nj.nec.com
- Hoover, A. – Jean-Baptiste, G. – Jiang, X. – Flynn, P. – Bunke, H. – Goldgof, D. – Bowyer, K. – Eggert, D. – Fitzgibbon, A. – Fisher, R.** (1995): An Experimental Comparison of Range Image Segmentation Algorithms, *NEC Research Index*, iteseer.nj.nec.com
- Intergraph Corp.** (2001): Working with GeoMedia Professional, *Intergraph Corporation Huntsville, Alabama*
- Jagdish, H. V. – Lieuwen, D. – Rastogi, R. – Silberschatz, A. – Sudarshan, S.** (1994): Dalí: A High Performance Main Memory Storage Manager, *AT&T Bell Labs.*, iteseer.nj.nec.com
- Microsoft** (1998): MSDN Library, Visual Studio 6.0, *Electronic Documentation*
- Niemann, T.** (1998): A Compact Guide to Sorting and Searching, epaperpress.com
- OpenGIS Consortium** (1998): OpenGIS Specification, *Third Edition*, *OpenGIS Consortium, Recommendation Paper*, www.opengis.org
- OpenGIS Consortium** (2001): Geography Markup Language GML 2.0, *OpenGIS Consortium, Recommendation Paper*, www.opengis.org
- Ortega, A. – Chrysafis, C.** (1998): Line Based, Reduced Memory, Wavelet Image Compression, *NEC Research Index*, iteseer.nj.nec.com
- Procopiuc, O.** (1997): Data Structures for Spatial Database Systems, *NEC Research Index*, iteseer.nj.nec.com
- Shekhar, S. – Chawla, S. – Ravada, S. – Fetterer, A. – Liu, X. – Lu, C. T.** (1998): Spatial Databases: Accomplishments and Research Needs, *NEC Research Index*, iteseer.nj.nec.com
- Tarboton, D. G. – Ames, P. D.** (2001): Advances in the mapping of flow networks from digital elevation data, *NEC Research Index*, iteseer.nj.nec.com
- Tarboton, D. G.** (1996): A New Method for the Determination of Flow Directions and Upslope Areas in Grid Digital Elevation Models, *NEC Research Index*, iteseer.nj.nec.com

6.3. Kutatással kapcsolatos előadások, poszterek

- Bácsatyai, L. – Czimber, K. – Király, G.** (2001): A digitális fotogrammetria újabb eredményei a NYME Földmérési és Távérzékelési Tanszékén, *Somogyi József 70 éves – Tudományos előadó ülés, Sopron, MTA GGKI, 2001. December 6-7.*
- Bácsatyai, L. – Czimber, K.** (1995): GIS activities at University of Forestry, *EASICC meeting, Sopron, 1995. november 3.*
- Bácsatyai, L. – Czimber, K.** (1996): Erdészeti térinformatikai alkalmazások, *Térinformatika a regionális fejlesztésekben, Debrecen, 1996 január 25-26.*
- Bácsatyai, L. – Czimber, K.** (2000): Térinformatikai stratégia a Soproni Egyetem Erdőmérnöki Karán, *Budapest, HUNGIS - IX: Térinformatika a Felsőoktatásban Szimpózium, 2000. október 16.*
- Bácsatyai, L. – Czimber, K.** (2000): Térinformatikai szoftverek a kutatásban és az oktatásban a Nyugat-Magyarországi Egyetem Erdőmérnöki Karán, *HUNGIS - VIII: Térinformatika a Felsőoktatásban Szimpózium, Budapest, 2000. október 16.*
- Czimber K.** (1996): Erdészeti- és geoinformatika szoftverbemutató, *Zalaegerszegi Erdőtervezési Iroda és Erdőfelügyelőség, Zalaegerszeg, 1996. október 12.*
- Czimber, K. – Gazdag, I.** (1995): Térinformatika az erdőgazdálkodásban, *1995. évi OEE konferencia. EFE, Sopron, 1995. augusztus 24.*
- Czimber, K. – Márkus, I.** (1997): Erdőterületek térképezése és monitorozása távérzékelési módszerek és GIS alkalmazásával, *VII. Földfotó Szeminárium, Budapest 1997. március 13-14.*
- Czimber, K. – Morschauser, T. – Salamon, É.** (1995): Comparative Analysis of Vegetation Map based on Digital Elevation Model, *XXV. EURECO - Nemzetközi Ökológiai Kongresszus, MKKTE, Budapest, 1995. augusztus 20-25.*
- Czimber, K.** (1994): Digitális felületmodellek, *1994. évi WOODtech, OEE konferencia. EFE, Sopron, 1994. április.*
- Czimber, K.** (1994): Digitális felületmodellek, *III. Biológiai Vándorgyűlés. JATE, Pécs, 1994. július.*
- Czimber, K.** (1994): Digitális felületmodellezés a Fertő-tó fenék- és iszaptérképezésében, *VI. Fertő tavi anket a Magyar Hidrológiai Társaság rendezésében. Sopron, 1994. június 1.*
- Czimber, K.** (1996): DigiTerra Image software demonstration, *Workshop on Soil Degradation Assesment, MTA Talajtani és Agrokémiai Kutató-intézete. Budapest, 1996. szeptember 19.*
- Czimber, K.** (1996): Erdészeti- és geoinformatika szoftverbemutató, *KAEG Rt, Győr, 1996. október 23.*
- Czimber, K.** (1996): Erdészeti- és geoinformatika szoftverbemutató, *Kecskeméti Erdőfelügyelőség, Kecskemét, 1996. október 24.*
- Czimber, K.** (1996): Erdészeti- és geoinformatika szoftverbemutató, *SE Földmérési és Távérzékelési Tanszék, Sopron, 1996. december 10.*

- Czímber, K.** (1997): Erdészeti- és geoinformatika országos szoftverbemutató, *Sopron, 1997. március 14.*
- Czímber, K.** (1997): Erdészeti- és geoinformatika szoftverbemutató, *Pilisi Parkerdő Rt, Visegrád, 1997. március 10.*
- Czímber, K.** (1997): Erdészeti- és geoinformatika szoftverbemutató, *VERGA Rt, Veszprém, 1997. április 18.*
- Czímber, K.** (1997): Erdészeti- és geoinformatika szoftverbemutató, *OEE rendezvény, SEFAG Rt, Kaposvár, 1997. május 24.*
- Czímber, K.** (1997): Erdészeti- és geoinformatika szoftverbemutató, *BEFAG Rt, Keszthely, 1997. szeptember 10.*
- Czímber, K.** (1997): Erdészeti- és geoinformatika szoftverbemutató, *OEE rendezvény, Zalaerdő Rt, Kaposvár, 1997. október 28.*
- Czímber, K.** (1997): Erdészeti- és geoinformatika szoftverbemutató, *Állami Erdészeti Szolgálat, Budapest, 1997. január 10.*
- Czímber, K.** (1998): Erdészeti Információs Rendszer bemutatása, *Alföldi Erdőkért KHT Informaikai Fórum, Lónya, 1998. április 28.*
- Czímber, K.** (1998): Erdészeti Információs Rendszer bemutatása, *Kecskemét, 1998. szeptember 10.*
- Czímber, K.** (1999): DigiTerra a „Magyar Erdőkért”, *IX. Országos Térinformatikai Konferencia, 1999. Szeptember 22-24. Szolnok*
- Czímber, K.** (1999): Erdészeti Információs Rendszer, *Állami Erdészeti Szolgálat szakmai rendezvénye, Zamárdi, 1999. Május 13.*
- Czímber, K.** (2000): Korszerű térinformatikai funkciók DigiTerra Map programban, *X. Országos Térinformatikai Konferencia, Szolnok, 2000. Szeptember 27-29.*
- Czímber, K.** (2001): A vektortérképtől az ortofotó automatikus előállításáig, *Nemzetközi Térinformatika Világnap, Térinformatika a Kormányzati Feladatok Végrehajtásában Földművelésügyi és Vidékfejlesztési Minisztérium, Budapest, 2001. November 14.*
- Czímber, K.** (2001): GIS in Forestry, *EURISY/Hungarian Space Office, Summer School for PhD Students on Environmental Security, Budapest, 2001. Szeptember 1-7.*
- Czímber, K.** (2001): Hatékony légifelvétel kiértékelés a DigiTerra Map segítségével, *XI. Országos Térinformatikai Konferencia, Szolnok, 2001. Szeptember 26-28.*
- Czímber, K.** (2001): Légifelvétel kiértékelés a DigiTerra Map segítségével, *Műszaki Térinformatika Konferencia, Budapest, Matáv Tölösi Konferencia Központ, 2001. November 12-13.*
- Czímber, K.** (2002): DigiTerra fejlesztési eredmények 2001-2002, *X. Országos Térinformatikai Konferencia, Szolnok, 2002. November 14-15.*
- Czímber, K.** (2002): Korszerű térinformatikai rendszerek, a képfeldolgozás módszerei, *Önkormányzati Térinformatika, Nagykanizsa, 2002. Január 24.*
- Márkus, I. – Czímber, K. – Király, G.** (1996): Forest Mapping Using Satellite Imagery and GIS in Hungary, *MERA Project 1994-1996. Results Conference. Pozsony, 1996. december 10-11.*

- Márkus, I. – Czímber, K.** (1996): Experiments of Forest Ecosystems Mapping in Hungary in the Frame of the PHARE MERA Project, *Application of Remote Sensing in European Forest Monitoring. International Workshop. BOKU Wien, 1996. október 14-16.*
- Márkus, I. – Czímber, K.** (1996): Forest Mapping Using Satellite Imagery and GIS in Hungary, *MERA Technical Workshop on Environmental Mapping and Monitoring Using Satellite Remote Sensing and GIS. Varsó, 1996. november 28-29.*
- Márkus, I. – Király, G. – Czímber, K. – Szentesi, L.** (2000): Research on Application of Remote Sensing and GIS in the Forestry, *Das III. Internationale Symposium: Die Applikation der Fernerkundung in der Forstwirtschaft. TU Zvolen, 2000. Szeptember 12-14.*
- Mátyás, Cs. – Czímber, K. – Berki, I.** (2000): A magyar erdőtakaró klimatikus fenyegetettsége: egy modellezés első eredményei, *Házi Tudományos Konferencia, Sopron, 2000. December 15.*