

# TÉRINFORMATIKA

*Előadási jegyzet*

*Készítette:* Dr. Katona Endre  
SZTE Képfeldolgozás és Számítógépes Grafika Tanszék

*Ellenőrizte:* Dr. Mucsi László  
SZTE Természeti Földrajzi és Geoinformatikai Tanszék

A jegyzetben talált esetleges hibákat és egyéb észrevételeket  
kérem jelezni a [katona@inf.u-szeged.hu](mailto:katona@inf.u-szeged.hu) címre

Szegedi Tudományegyetem, 2013.

## Tartalom

BEVEZETÉS.....	5
1. INFORMATIKAI ALAPOK.....	8
1.1. Raszteres és vektoros adatábrázolás.....	8
1.1.1. Raszteres adatábrázolás.....	8
1.1.2. Vektoros adatábrázolás.....	9
1.1.3. Összehasonlítás.....	10
1.1.4. Konverziók.....	10
1.2. A térinformatika speciális hardware eszközei.....	11
1.2.1. Beviteli eszközök.....	11
1.2.2. Kiviteli eszközök.....	12
1.3. Raszteres szoftver rendszerek.....	12
1.4. Vektoros szoftver rendszerek.....	13
1.4.1. Rajz strukturálása.....	13
1.4.2. Rajzrészlet kijelölése.....	14
1.4.3. Pontosságot biztosító eszközök.....	14
1.4.4. További CAD lehetőségek.....	15
2. TÉRKÉPÉSZETI ALAPOK.....	16
2.1. Térképtípusok.....	16
2.2. Koordináta rendszerek, vetületi rendszerek.....	17
2.2.1. A Föld alakja.....	17
2.2.2. Gömbi geometria.....	17
2.2.3. Gömbi koordináta-rendszerek.....	18
2.2.4. Vetületi rendszerek.....	18
2.3. Transzformációk.....	25
2.3.1. Transzformáció típusok.....	25
2.3.2. Együtthatók meghatározása.....	26
2.3.3. A transzformáció számítása.....	27
2.4. A térképezés alapjai.....	27
2.4.1. Terepfelmérés.....	28
2.4.2. Távérzékelés.....	30
3. VEKTOROS TÉRINFORMATIKAI RENDSZEREK.....	34
3.1. A modellalkotás folyamata.....	34
3.2. Vektoros adatmodellek.....	35
3.2.1. Alapvető objektumtípusok.....	35
3.2.2. Spagetti modell.....	36
3.2.3. Topológikus modellek.....	37
3.2.4. Tartománytérkép (folttérkép).....	37
3.2.5. Hálózat.....	39
3.2.6. Folytonos felület.....	40
3.3. Térbeli indexelés.....	41
3.3.1. Négyzetrács index (grid index).....	41
3.3.2. Négyesfa index.....	43
3.4. Vektoros algoritmusok.....	44
3.4.1. Egyenesszakaszok metszéspontja.....	44
3.4.2. Vonallancok metszése.....	45
3.4.3. Poligonok területe.....	45
3.4.4. Pont-poligon algoritmus.....	46
3.4.5. Sok pont, sok poligon vizsgálata.....	47
3.4.6. Poligon overlay algoritmus.....	48
3.5. Adatbázismodellek: térbeli és leíró adatok összekapcsolása.....	50
3.5.1. Szétválasztott modell.....	50
3.5.2. Integrált modell: tisztán relációs megközelítés.....	54
3.5.3. Integrált modell: objektum-relációs megközelítés.....	55
3.5.4. Integrált modell: térbeli adattípusok.....	56
3.5.5. Adatbázismodellek összefoglalása.....	57
4. RASZTERES TÉRINFORMATIKAI RENDSZEREK.....	58
4.1. Adatbázis kapcsolat.....	59

4.2. Bevezető példa.....	59
4.3. Pixelenkénti műveletek.....	60
4.4. Lokális szomszédsági műveletek.....	61
4.4.1. Konvolúció.....	61
4.4.2. Lejtés és lejtésirány.....	61
4.5. Távolsági szomszédsági műveletek.....	62
4.5.1. Távolság fedvény készítése.....	62
4.5.2. Övezetképzés.....	63
4.5.3. Övezetképzés moduláló fedvényel.....	64
4.5.4. Láthatóság.....	65
4.5.5. Műveletek tartományokon.....	65
5. DIGITÁLIS TEREPMODELLEK.....	68
5.1. TIN előállítás magassági ponthalmazból.....	70
5.2. TIN előállítás szintvonalrajzból.....	71
5.3. Térbeli interpolációs módszerek.....	71
5.3.1. Távolság inverzével súlyozott mozgóátlag.....	72
5.3.2. Polinomiális interpoláció.....	72
5.4. DEM előállítás térbeli interpolációval.....	73
5.4.1. Az Intercon módszer.....	73
5.4.2. Variációs spline interpoláció.....	73
5.5. TIN előállítás DEM-ből.....	76
5.5.1. Fowler-Little algoritmus.....	76
5.5.2. VIP algoritmus.....	77
6. ALKALMAZÁSOK.....	78
6.1. Ingatlan nyilvántartás.....	78
6.2. Közmű nyilvántartás.....	78
6.3. Önkormányzatok.....	78
6.4. Közlekedés, navigáció.....	79
6.5. Honvédelem.....	79
6.6. Környezetvédelem.....	79
6.7. Meteorológia.....	79
6.8. Geomarketing.....	80
6.9. Intézmény üzemeltetés.....	80
7. FEJLESZTŐ CÉGEK, SZOFTVEREK.....	81
7.1. Általános jellemzők.....	81
7.2. Vektoros rendszerek.....	81
7.3. Raszteres rendszerek.....	83
7.4. Hazai fejlesztések.....	84
A. FÜGGELÉK: MicroStation.....	85
A.1. Alapok.....	85
Adattárolás.....	85
Koordinátarendszer, grid.....	86
Parancsvezérlés.....	87
Rétegek.....	87
DGN fájl felépítése.....	87
A.2. Rajzszerkesztés.....	88
A Main eszköztár.....	88
Snapping.....	88
Cell-ek.....	89
Patterning.....	89
A.3. Adatbázis kapcsolat.....	90
Xbase kapcsolat inicializálása.....	90
ODBC kapcsolat inicializálása.....	91
Adatbázis kapcsolat kezelése.....	92
Űrlap (Form) használata.....	93
Displayable attributes (DAS).....	93
Adatfeltöltés NEW linkage módban.....	94
Példa adatbázis kapcsolatra (Foxpro).....	94
A.4. Referenciafájl, digitalizálás.....	95
B. FÜGGELÉK: ArcInfo.....	96

C. FÜGGELÉK: Grafikus formátumok .....	97
TIFF formátum.....	97
GeoTIFF formátum .....	98
DXF formátum.....	99
D. FÜGGELÉK: Polinomiális interpoláció .....	102
E. FÜGGELÉK: Variációs spline interpoláció .....	104
IRODALOM.....	106
Néhány webhely:.....	107

## BEVEZETÉS

Mi a térinformatika? Erre a kérdésre próbálunk választ adni ebben a fejezetben a fontosabb alapfogalmak definiálásával, és egy tipikus alkalmazási példa bemutatásával.

*Térinformatika, geoinformatika:* digitális térképekre épülő informatika. Interdiszciplináris terület, amely elsősorban földrajzi, térképészeti és informatikai szakismeretekre épül.

*Térinformatikai rendszer (térinformációs rendszer):* térképi alapú információs rendszer, amely grafikus (térképi) és nem grafikus (leíró) adatokat együtt, integráltan tud kezelni.

*GIS = Geographical Information System:* földrajzi információs rendszer. Tágabb értelemben megfelel a magyar *térinformatikai rendszer* fogalomnak. Szűkebb értelemben csak a kimondottan *földrajzi adatokra épülő* (pl. környezetvédelmi) rendszereket jelenti, és nem tartalmazza pl. a közműnyilvántartást. A továbbiakban a GIS-t a tágabb értelmezés szerint használjuk.

### Jellemző alkalmazási területek

*LIS = Land Information System:* ingatlan nyilvántartás. Egyrészt az ingatlanok (földrészletek, épületek) térképi rajzát, másrészt a hozzájuk kapcsolódó adatokat (tulajdoni lap) tartalmazza. A LIS térképei jelentik azt a közjogilag hiteles térképi alapot (földmérési alaptérkép), amelyre számos más alkalmazás épül.

*AM/FM = Automated Mapping / Facilities Management:* közmű nyilvántartás. Adott település közműveinek (víz, villany, gáz) vezetékrendszerét és berendezéseit tartalmazza a földmérési alaptérképre illesztve.

További alkalmazásokat a 6. fejezetben mutatunk be.

### Történeti áttekintés

Kezdetben (1960-as évek) csak a nem térbeli (pl. könyvelési) adatokat vitték számítógépre (*gépi adatbázisok*).

Később (1970-es évek) a gépek teljesítményének növekedésével és a grafikai eszközök fejlődésével lehetővé vált a térbeli (térképi) adatok hatékony számítógépes kezelése. Megjelentek a különféle *mérnöki tervező (CAD) rendszerek* és *képfeldolgozó rendszerek* (légifényképek, műholdképek elemzése).

Ezután (1980-as évek) felmerült az igény a grafikus és nem grafikus adatokat integráltan kezelő információs rendszerek iránt, ez vezetett a térinformatika rohamos fejlődéséhez.

Az 1990-es években a térinformatikai rendszerek már személyi számítógépen is elérhetővé váltak, és ez nagyságrendekkel növelte a felhasználók körét.

A kereskedelmi korlátozások miatt 1990 előtt a GIS és CAD rendszerek Kelet-Európában nem, vagy alig voltak elérhetőek, ezért a kelet-európai országok saját fejlesztésekkel próbálkoztak. Ezek a saját fejlesztésű rendszerek részben még ma is használatosak, de a nyugati rendszerek jórészt már kiszorították őket.

Jelenleg Magyarországon a nem térbeli adatokat már szinte mindenütt számítógéppel kezelik (relációs adatbázisok), de a térképek, tervrajzok esetén még előfordul a hagyományos, papíralapú nyilvántartás. A technológiaváltás során a tervező és igazgatási cégeknél a rajzasztalok szerepét fokozatosan CAD és GIS rendszerek veszik át.

### **1. példa:** ingatlan nyilvántartás

Jelenleg Magyarországon a telekkönyvi adatok (tulajdonos, helyrajzi szám, földrészlet területe, stb.) már általában számítógépen vannak, de az ún. földmérési alaptérképeket (amelyek – többek között – a földrészletek pontos határvonalait tartalmazzák, lásd 1. ábra) még részben hagyományosan kezelik. Az ebből adódó hátrányok:

- Különös gondot kell fordítani a pontosságra, mivel a földmérési alaptérkép jelenti az egyedül hiteles adatot, számos építési és tervezési munka is ennek alapján történik. Ezért a térképeket speciális alapanyagon (pl. fémbetétes lap) tárolják, amely évtizedek alatt sem torzul, zsugorodik. Ezzel együtt is a pontosság korlátozott, lényegében a térképen alkalmazott vonalvastagságnak felel meg.

- A változásvezetés nehézkes: a változásokat egy ideig eltérő színnel vezetik rá a térképre, egy idő után azonban a teljes térképet újra kell rajzolni. Az újrarajzolás tovább ronthatja a pontosságot.

- A térképek helyessége csak nehézkesen ellenőrizhető (pl. van-e két azonos helyrajzi szám, egy földrészlet térképen számított területe megfelel-e az adatbázisban szereplő területértéknek, stb.)

- Bizonyos kimutatások igen nehézkesen végezhetőek el. (Például kíváncsiak vagyunk azon földtulajdonosok adataira, akiknek a földjét érinti egy tervezett autópálya útvonala.)

A fenti hátrányok az ingatlan nyilvántartás térinformatikai kezelésével küszöbölhető ki. Ez alapvetően egy digitális térképet és egy hozzá kapcsolt háttéradatbázist jelent, amelyeket a GIS szoftver együtt, integráltan tud kezelni. Mindennek a részleteivel a további fejezetekben fogunk megismerkedni.

**2. példa:** Önkormányzati információs rendszer, óvodai körzetek. A rendelkezésre álló adatok:

- Népeségi adatok: mely lakásban hány óvodás korú gyermek van bejelentve.
- Az egyes óvodák postai címe.

A fenti adatokat tartalmazó adatbázis alapján nehéz eldönteni, hogy például hol szükséges új óvoda, illetve hol van többszörösen ellátott terület. A feladat térinformatikai támogatása a következő lehet:

- Az óvodáskorú gyermekek népsűrűség térképe, amelyen sötétebb ill. világosabb színárnyalat jelöli az egyes terület egységekre eső óvodáskorú gyermekek számát.

- Óvodai körzetek térképe: az egyes óvodák helye körül 500 m-es kör rajzolásával jelképezi az óvoda ellátási körzetét.

A fenti két térkép egymásra vetítve jól mutatja az ellátatlan ill. többszörösen ellátott körzeteket, és megfelelően támogatja a szükséges önkormányzati döntéseket.



1. ábra: 1:2000 méretarányú földmérési alaptérkép (kataszteri térkép) részlete.

# 1. INFORMATIKAI ALAPOK

## 1.1. Raszteres és vektoros adatábrázolás

### 1.1.1. Raszteres adatábrázolás

A képet mátrix formájában tároljuk (2. ábra). Egy mátrixelem szokásos elnevezései: *képpont*, *pixel*, *cella*. A kép típusát alapvetően a *bit-per-pixel érték* határozza meg, vagyis az, hogy egy képpont hány bitből áll. Néhány jellemző típus:

- 1 bites pixelek: bináris kép.
- 8 bites pixelek: monochrom kép, 256 szürkeárnyalat.
- 24 bites pixelek: színes kép, ahol a színek a három alapszín (piros, zöld, kék) keverékeként kódoltak, mindegyik színekomponeensnél 256 árnyalattal (3\*8 bit).
- Multispektrális műholdkép: például 7 sávban, sávonként 8-bites (vagy 16-bites) pixelek.

Előfordulhat, hogy a rasztermátrix nem képi információt hordoz (pl. talajtérkép, terepmodell), ilyenkor a pixelenkénti bitek száma is a fentitől tetszőlegesen eltérő lehet.

*Felbontás (geometriai)*: megadja, hogy egy pixel mekkora területnek felel meg a valóságban (pl. 10 x 10 méter).

```

0 0 0 0 0 0 0 0 0 0 0 0
0 1 0 0 0 0 0 0 0 0 0 0
0 1 1 0 0 0 0 0 0 0 0 0
0 1 0 1 0 0 0 0 0 0 0 0
0 1 0 0 1 0 0 0 0 0 0 0
0 1 0 0 0 1 0 0 0 0 0 0
0 1 0 0 0 0 1 0 0 0 0 0
0 1 0 0 0 0 0 1 0 0 0 0
0 1 0 0 0 0 0 0 1 0 0 0
0 1 1 1 1 1 1 1 1 1 0 0
0 0 0 0 0 0 0 0 0 0 0 0

```

2. ábra: egy háromszög kontúrja raszteres adatábrázolásban

*Jellemző adatformátumok*: TIFF, PCX, BMP, JPEG, stb.

*A TIFF grafikus formátum*. Térinformatikai alkalmazásokban leggyakrabban ezt használják. TIFF = Tagged Image File Format, az 1980-as évek végén kidolgozott raszteres képfomátum. A TIFF fájl felépítése:

- Header (8 byte, részletesebben lásd a Függelékben).
- IFD = Image File Directory: a képet leíró paramétereket tartalmazza (tag-ek, magyarul címkék). A tag-ek többsége opcionális (azaz elhagyható), igen rugalmas képleírást tesznek lehetővé (részletesebben lásd a Függelékben).
- Maga a kép (pixelek sorozata).



*GeoTIFF formátum.* Az 1990-es évek közepén definiált, speciális tag-ekkel bővített, térinformatikai célú TIFF formátum. A GeoTIFF kép olyan programokkal is megjeleníthető, amelyek csak az alap TIFF formátumot ismerik, de ezek természetesen nem tudják értelmezni a speciális címkéket. A geoTIFF formátumot elsősorban *georeferencia* leírására használják: ez lényegében az alkalmazott vetületi rendszer leírását jelenti, amely segítségével az egyes pixeleknek megfelelő vetületi koordináták meghatározhatók.

### 1.1.2. Vektoros adatábrázolás

A képet *rajzelemek* halmazaként (rendszereként) tároljuk, az egyes rajzelemeket koordináta-geometriai eszközökkel írjuk le. Leggyakoribb rajzelem az ún *vektor*, vagyis egy (irányított) egyenesszakasz, amelyet végpontjainak koordinátaival adunk meg:

*LINE*  $x_1, y_1, x_2, y_2$

Például, a 2. ábrán látható háromszög vektoros adatábrázolásban (bal alsó sarok a koordinátarendszer középpontja):

```
LINE 1, 1, 1, 9
LINE 1, 1, 9, 1
LINE 1, 9, 9, 1
```

Példák további rajzelemekre:

*CIRCLE*  $x, y, r$ : Kör  $(x, y)$  középponttal és  $r$  a sugárral.

*TEXT*  $x, y, méret, irány, szöveg$ : felirat a rajzon az  $(x, y)$  pontban, adott méretben és irányban. A felirat tartalmát a "szöveg" ASCII jelsorozat adja meg.

Jellemző adatformátumok: DXF, IGES

A *DXF adatformátum* (Drawing eXchange Format): AutoDesk cég specifikálta és folyamatosan fejleszti. A legtöbb vektoros rendszer tudja importálni ill. exportálni. Szöveges és bináris változata használatos. Az alábbi szekciókból áll:

- HEADER: változók beállítása (koordinátarendszer, stb.)
- TABLES: vonaltípus, réteg, stb.
- BLOCKS: blokk definíciók
- ENTITIES: rajzelemek

A fájlformátum részletesebb leírása a Függelékben található.

### 1.1.3. Összehasonlítás

<i>Raszter</i>	<i>Vektor</i>
A látszati kép	A kép struktúrája
Minden raszterponthoz megadja , hogy ott <i>milyen objektum</i> van	Minden objektumhoz megadja, hogy az a síkon <i>hol</i> van
Pontosság a felbontástól függ	Pontosság a számábrázolástól függ
Nagyításnál: durvább lesz	Nagyításnál nem lesz durvább
Transzformáció: lassú, torzulhat	Transzf. gyors, gyakorlatilag nem torzul
Monitoron közvetlenül megjeleníthető	Megjelenítéséhez rajzolóprogram szükséges
Tárolóterület: képmérettől függ	Tárolóterület a rajz bonyolultságától függ

Megjegyzések:

– *Transzformáció*: ha például egy képet 360 db 1-fokos elforgatással körbe forgatunk, akkor raszterkép esetén számottevő torzulással számolhatunk, vektoros esetben a torzulás nem jelentős.

– *Tárolóterület*: ha adattömörítést alkalmazunk (például ZIP), akkor ugyanazon kép (mondjuk egy vonalrajz) raszteres és vektoros változatának mérete között már nem lesz nagy különbség.

### 1.1.4. Konverziók

*Vektor* → *raszter*: az egyes rajzelemek raszteres képét kell algoritmikusan előállítani, ami viszonylag könnyen megoldható (számítógépes grafika). Vektoros kép monitoron való megjelenítésekor mindig ez történik, mivel a monitor raszteresén dolgozik.

*Raszter* → *vektor*: a raszterképen az egyes objektumokat kell algoritmikusan felismerni, és megfelelő vektoros kóddal helyettesíteni. Ez lényegében alakfelismerési feladat, amelynek automatikus gépi megoldása kielégítő minőségben igen nehéz.

Vegyes adatábrázolás: raszter + vektor egymásra vetítve.

## 1.2. A térinformatika speciális hardware eszközei

A térinformatika elsősorban a grafikus adat be- és kivitel területén igényel speciális eszközöket, ezeket a 3. ábra tekinti át.

	Vektor	Raszter
<i>Bevitel</i>	Digitalizáló tábla	Szkenner, digitális kamera
<i>Kivitel</i>	Tollas plotter	Tintasugaras plotter

3. ábra

### 1.2.1. Beviteli eszközök

*Digitalizáló tábla (tablet):* Egy elektronikusan vezérelt, A3...A0 méretű táblából, és egy egérhez hasonló pozícionáló eszközből (*tábla kurzor*) áll. A tábla kurzor – az egértől eltérően – abszolút pozíciót érzékelő eszköz: mindig pontosan érzékeli, hogy a tábla mely pontján van, akkor is, ha felemelve helyezzük át.

A digitalizáló tábla manuális vektoros adatbevitelt támogat. A digitalizálandó rajzot a táblára rögzítik, majd a tábla kalibrálásával elérik, hogy a rajz négy sarokpontja a képernyőn látható rajzterület négy sarkának feleljen meg. Ezután a tábla kurzorral manuálisan követik a rajz vonalait, és a vonal végpontoknál gombnyomással viszik be a megfelelő koordinátákat.

Az eljárás hátránya, hogy az adatbevitel pontossága és teljessége csak nehezen ellenőrizhető. Ennek ellenére, viszonylagos olcsósága miatt, ez a leggyakrabban használt adatbeviteli eszköz.

*Szkenner (scanner):* optikai leolvasó, raszteres adatot állít elő. Főbb típusok:

– *Síkszkenner:* általában A4 (esetleg A3) méretű. A szkennelendő lapot egy üveglapra kell helyezni, amelyet egy levilágító-érzékelő berendezés soronként letapogat. A síkszkenner igen pontos adatbevitelt biztosít, de nagyobb méretű berendezések igen drágák, ezért a térinformatikában ritkán használatosak.

– *Dobszkenner:* A1, A0 méretű (valójában csak a szélesség korlátozott, a hosszúság nem). A síkszkennerrel ellentétben itt a levilágító-érzékelő berendezés rögzített, és előtte halad el a digitalizálandó lap. Fényes felületű lapok (fóliák) szkennelése esetén a laptovábbítás egyenetlen lehet, ami a szkennelés pontosságát rontja. Bár a dobszkenner is viszonylag drága berendezés, a térinformatikában ezt használják legáltalánosabban.

– *Kéziszkenner (handy scanner):* kb. 10 cm széles sávot visz be. A túl lassú húzást kompenzálja, a túl gyorsat nem. Ma már nemigen használják.

– *Plotterre szerelt kéziszkenner:* sávonként lehet vele nagyobb rajzot, térképet szkennelni. Plotterrel már rendelkező cégek számára olcsó, de nem elég biztonságos megoldás. (A sávok elcsúszhatnak egymáshoz képest.)

*Digitális kamera:* az optika általában mátrix alakban elhelyezett érzékelőkre vetíti a képet, eredményül szürkeárnyaltos vagy színes raszterkép keletkezik. A térinformatikában műholdképek, légifényképek készítésére használatos.

### 1.2.2. Kiviteli eszközök

*Plotter (rajzgép):* A3...A0 méretű rajzok készítésére alkalmas.

– *Tollas plotter:* vektorosan vezérelt berendezés. Két típusa van: a síkplotter, ahol a rajzoló tollat x és y irányban mozgatja a vezérlő elektronika, és a dobplotter, ahol a toll csak x irányban, a papír pedig y irányban mozog. Rendszerint több tollat használ, ezzel biztosít többféle színt ill. vonalvastagságot.

Hátrányok: a tollak gyakran beszáradnak ill. kimerülnek, ilyenkor a félbemaradt rajzot újra kell kezdeni. Dobplotternél a sokszori papírmozgatás után kisebb elcsúszások lehetségesek.

– *Tintasugaras plotter (raszter plotter):* a tintasugaras nyomtatókhoz hasonlóan működik. Raszteresen vezérelt berendezés, ezért a vektoros rajzot nyomtatás előtt raszteresre konvertálja. Ha elég jó a felbontása (a ferde vonalak nem lesznek lépcsősek), akkor egyértelműen jobb rajzminőséget biztosít, mint a tollas plotter. Különleges előnye, hogy raszteres, ill. hibrid raszter-vektor állományokat is lehet vele nyomtatni. Jóval drágább, mint a tollas plotter, ennek ellenére egyre több felhasználó tér át a tintasugaras plotterre.

## 1.3. Raszteres szoftver rendszerek

a). *rajzszerkesztő rendszerek (raszter editorok):* raszteres rajzok létrehozására, módosítására (retusálására) szolgálnak. Ilyen rendszerek például a Windows *Paint* programja, *PhotoShop*, *PaintShopPro*, stb. Speciálisan térképek feldolgozását támogatja például a *MicroStation Descartes*.

b). *képfeldolgozó rendszerek:* digitális képforrásból (kamera, szkennel) származó raszterkép algoritmikus feldolgozására szolgálnak. Jellemző képfeldolgozó műveletek:

- hisztogram számítás,
- konvolúció (zajszűrés, élkiemelés, stb. céljára).
- geometriai transzformációk (lineáris, nemlineáris torzítás)

Egyszerű képfeldolgozó rendszernek tekinthető a Windows alapú *PhotoStyler*. Speciális rendszereket fejlesztettek ki például orvosi képek, műholdképek (meteorológiai képek) feldolgozására.

## 1.4. Vektoros szoftver rendszerek

Tipikusan vektoros szoftverek a *mérnöki tervező (CAD) rendszerek*. (CAD = Computer Aided Design = számítógéppel segített tervezés). A CAD rendszerek alapja rendszerint egy fejlett *rajzszerkesztő rendszer* amely többnyire valamilyen speciális szakterületet támogat. Például:

- Építészeti tervező rendszer
- Gépészeti tervező rendszer
- Áramkörtervező rendszer
- Térképszerkesztő rendszer

Az általános célú CAD rendszerek közül legismertebb az *AutoCAD*.

*Szokásos rajzelemek* (4. ábra):

- *egyenesszakasz*:  $x_1, y_1, x_2, y_2$
- *vonallánc* (*töröttvonal, line string, polyline, „ív”*):  $x_1, y_1, \dots, x_n, y_n$
- *kör*:  $x, y, r$
- *ellipszis*:  $x, y, r_1, r_2, \alpha$  ( $\alpha$ : nagytengely szöge)
- *ív (arc)*:  $x, y, r_1, r_2, \alpha, \beta_1, \beta_2$  ( $\beta_1, \beta_2$ : az ív kezdő- és végpontjának szögei)
- *görbe*:  $g_1, x_1, y_1, \dots, x_n, y_n, g_2$  ( $g_1, g_2$ : irány a végpontokban)
- *B-spline*: kontrollpontok segítségével definiált görbe
- *alakzat (poligon)*:  $x_1, y_1, \dots, x_n, y_n$  ahol  $x_{n+1} = x_1, y_{n+1} = y_1$  (zárt poligon)
- *felirat (text)*:  $x, y$ , méret, irány, szöveg

Minden egyes rajzelemhez *attribútumok* tartoznak: szín, vonaltípus, vonalvastagság, réteg (lásd alább), stb.

### 1.4.1. Rajz strukturálása

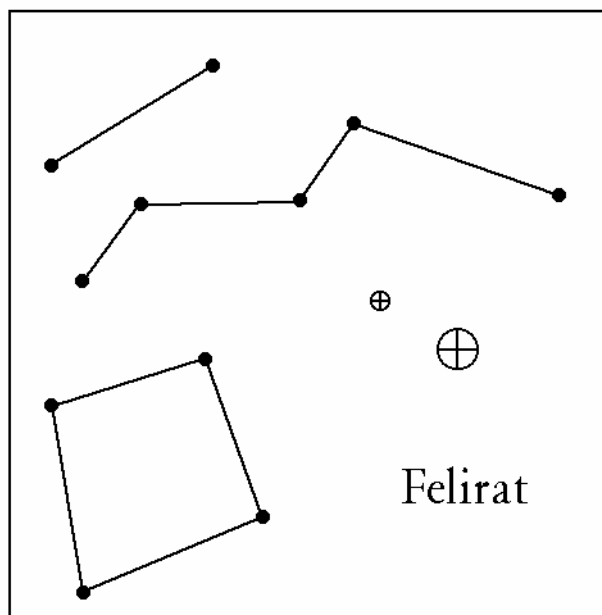
*Rétegekre bontás (fóliázás)*: rajzelemek csoportosítása jelentésük szerint, valamilyen szempontból. Egy réteg többféle rajzelem típust is tartalmazhat. Például egy épület alaprajza az alábbi rétegeket tartalmazhatja:

- falak,
- helyiségek feliratai,
- ajtók és ablakok,
- vízvezetékek és elzárócsapok,
- elektromos vezetékek.

Az egy réteghez tartozó rajzelemek együtt kezelhetők, például közös szín, vonaltípus és vonalvastagság rendelhető hozzájuk. Az egyes rétegek megjelenítése külön-külön ki-bekapcsolható.

*Blokk (cell)*: többször ismétlődő rajzrészlet, jelkulcsi elem kezelésére szolgál, például túristatérképen benzinkút jele (körbe rajzolt T betű), vagy épület homlokzatrajzon ablak, vagy gépészeti rajzon csavar (lásd még a körbe rajzolt kereszt szimbólumot a 4. ábrán). A blokk kezelése két részből áll:

- *blokk definíció*: a blokk egy mintapéldánya, tetszőleges rajzelemek sorozatából áll.
- *blokk hivatkozás*, alakja: (blokknév,  $x, y, \alpha, \text{zoom}$ ). A *blokknév* blokk beillesztését írja elő a rajz ( $x, y$ ) koordinátájú pontjára,  $\alpha$  elforgatási szöggel és *zoom* nagyítási faktorial. (Általánosabb esetben transzformációs mátrix.)



4. ábra: Jellemző rajzelemek (egyenesszakasz, töröttvonal, poligon, felirat) és blokk hivatkozások.

### 1.4.2. Rajzrészlet kijelölése

A kijelölés általában téglalappal, vagy zárt poligonnal történik. A kijelölt rajzelemek meghatározásának lehetséges módjai:

- csak a teljesen belül lévőket jelöljük ki,
- mindent kijelölünk, amelynek valamely részlete belül van,
- a kijelölési határ mentén elvágjuk a rajzelemeket, és csak a belül lévő részek kerülnek kiválasztásra. (Mellékhatás: az elvágott rajzelemek törlődnek, és helyettük több kisebb rajzelem keletkezik.)

### 1.4.3. Pontosságot biztosító eszközök

*Pontos (numerikus) koordináta megadás*

*Pontrács (grid):* egyrészt tájékozódást szolgál, másrészt viszont megkövetelhető, hogy csak grid-re eső pontokat lehessen bevinni.

*Csatolás (snap):* az aktuálisan bevitt pont ráhúzása a legközelebbi rajzelem megfelelő pontjára. Példák:

- Poligon bezárása (a kezdő és záró pont pontos megegyezésének biztosítása).
- Rajz folytatása már berajzolt szakasz végpontjából.
- T-elágazás (meglévő egyenesszakasz közepére való pontos csatlakozás).

A fentieket speciális vonaltípusokra (kettős, szaggatott, stb.) is helyesen hajtja végre a rendszer.

#### **1.4.4. További CAD lehetőségek**

*Kitöltő minták alkalmazása (patterning):* adott terület (például poligon belsejének) kitöltése bizonyos mintázattal. (vonalkázás, pontozás, téglafal minta, stb.).

*Méretezés (dimensioning):* a műszaki dokumentációkban szokásos méretvonalak automatikus felvitele, ill. méretszámok automatikus kalkulálása a rajzi méretek alapján. (A rajz módosításakor nem kell újra méretezni.)

*Parancsnyelv:* minden interaktív műveletnek van egy parancs megfelelője. Parancsok sorozatából paraméterezett segédprogramok készíthetők. Példa: interaktívan megadott két pont között kerítés rajzolása.

*Két monitoros üzemmód:* egyik monitoron a rajz, másikon a menük.

## 2. TÉRKÉPÉSZETI ALAPOK

### 2.1. Térképtípusok

*Térkép:* a Föld felszínén illetve azzal kapcsolatban álló anyagi vagy elvont dolgoknak kicsinyített, általánosított, síkbeli megjelenítése. (A Nemzetközi Térképészeti Szövetség (International Cartographic Association, ICA) meghatározásának megfelelően.)

Megjegyzés: A *map* szó nemcsak térképet, de a matematikában leképezést is jelent. Valóban, a térkép is általában a földfelszín leképezése egy papírlapra meghatározott szabályok szerint.

#### Méretarány

A méretarány a térképi távolság és a valós távolság hányadosa. (Ezt a meghatározást a vetületi rendszereknél majd pontosítjuk.) Ha a térkép méretaránya 1:50.000, akkor a térképen 1 mm a valóságban 50.000 mm-nek, azaz 50 méternek felel meg a Föld felszínén.

A "kisméretarányú" és a "nagy méretarányú" jelzők használata gyakran téves vagy félreérthető, ezért fontos tisztázni:

– *nagy méretarányú* a térkép, ha az 1:m hányados 1:10.000-nél nagyobb (vagyis  $m < 10.000$ ). A térkép részletgazdag, az egyes objektumok relatíve nagy méretben jelennek meg.

– *kisméretarányú* a térkép, ha az 1:m hányados értéke 1:10.000 vagy ennél kisebb (tehát  $m \cdot 10.000$ ). A térkép kevesebb részletet tartalmaz, az egyes objektumok relatíve kisebb méretben jelennek meg.

A méretarány nemcsak azt határozza meg, hogy hogyan ábrázoljuk az objektumokat, de azt is, hogy mit vagyunk képesek ábrázolni. Míg az 1:2000 méretarányú térképen az épületek, lámpaoszlopok önállóan ábrázolhatók, addig az 1:100.000 méretarányú térképen már nem.

*Térképszelvény:* egy összefüggő papírlapon ábrázolt térképrész.

A szelvények továbbosztása általában negyedeléssel történik, például egy 1:4000 szelvénynek négy 1:2000 szelvény felel meg.

#### Térképtípusok

(i) *Általános térképek:* A földfelszín kiválasztott természetes és mesterséges objektumait ábrázolja (domborzat, vízrajz, út-vasút, települések).

– *földmérési térképek, kataszteri térképek:* 1:500 ... 1:10.000 méretarány (1. ábra)

– *topográfiai térképek:* 1:10.000 ... 1:300.000 méretarány

– *földrajzi térképek:* 1:300.000-nél kisebb méretarány

(ii) A *tematikus térkép* valamely földrajzi téma(csoport) közvetítésére szolgál, mint például közművek, a népesség eloszlása, klimatikus viszonyok, áruforgalmi adatok stb. Általában egy általános térkép egyszerűsített változatára épül rá.



Más felosztás szerint:

- A *vonalas térkép* (vektor) az objektumokat szimbólumokkal és (határ)vonalakkal ábrázolja.
- A *fotótérkép* (raszter) légifénykép-felvételek alapján készül. A terep jellemzői a fotótérkép alapján önállóan interpretálhatók, bizonyos jellemzők azonosíthatók feliratok elhelyezésével is. Előállításuk viszonylagosan olcsó.

## 2.2. Koordináta rendszerek, vetületi rendszerek

### 2.2.1. A Föld alakja

A Föld alakja az ún. *geoid*, amelyet úgy kapunk, hogy a világoceánok közepes szintjét gondolatban a kontinensek alatt is folytatjuk.

A geoid durva közelítéssel gömbnek tekinthető. Finomabb közelítéssel egy, a pólusoknál kissé lapult *forgási ellipszoid*, amelynek egyenlítői átmérője kb. 0.3 %-kal nagyobb a sarki átmérőnél. A gömb és a forgási ellipszoid közötti eltérés kb. annyi, mint a földfelszín domborzati változatossága.

*Gauss-gömb*: a földfelszín adott pontjához legjobban simuló gömb.

### 2.2.2. Gömbi geometria

*Jelölés*:  $R$  a gömb sugara.

*Alapfogalmak*:

– *főkör*: a gömb középpontján áthaladó síknak a gömbfelülettel való metszete. Főkörív: a főkör egy szakasza.

A főkörök az egyenesek szerepét játsszák a gömbi geometriában. Eltérés a síkgeometriától, hogy nincsenek párhuzamos egyenesek, bármely két gömbi egyenes (főkör) metszi egymást.

– *főkörív középponti szöge*: a két végpontjából húzott gömbi sugarak által bezárt szög, radiánban mérjük. A főkörív hossza  $R \cdot \alpha$ .

– *két pont távolsága*: a pontokon áthaladó főkör rövidebb ívének hossza. (Két pont között a legrövidebb út.)

– *szög*: két gömbi egyenes bezárt szöge, amelyet a síkjaik hajlásszögével mérünk. (Ugyanezt a szögértéket kapjuk, ha felületi görbék hajlásszögeként definiáljuk a szöget.)

*Gömbi alakzatok*:

– *Euler-féle gömbháromszög*: a gömbfelület három pontját összekötő,  $\pi$ -nél rövidebb három főkörív által határolt terület. Szögei kisebbek  $\pi$ -nél, szögeinek összege viszont nagyobb  $\pi$ -nél. Felszíne:  $F = R^2(\alpha + \beta + \gamma - \pi)$ .

– *gömbkétszög*: két gömbi egyenes által határolt terület. Két főkör a gömböt négy gömbkétszögre osztja. Felszíne  $F = 2R^2\alpha$  (a két főkör szöge egyértelműen meghatározza). (A teljes gömbfelszín  $4R^2\pi$ .)

*Földrajzi fogalmak*:

– *Északi és déli pólus*: a gömb két kitüntetett, átellenes pontja.

– *Meridián*: a pólusokon áthaladó főkör.

- *Egyenlítő*: a meridiánokra merőleges főkör.
- *Loxodroma*: olyan görbe, amely minden meridiánt azonos szögben metsz. A loxodroma ugyan nem a legrövidebb utat adja két pont között, de ha egy jármű loxodroma pályán halad, akkor az iránytűhöz viszonyított haladási irányát nem kell megváltoztatni.

### 2.2.3. Gömbi koordinátarendszerek

Nem törekednek a gömbfelület síkba való kiterítésére, hanem közvetlenül a gömbfelületet írják le.

1. *Geocentrikus*: egy pontot az  $(x, y, z)$  derékszögű koordinátákkal azonosítunk, ahol a koordinátarendszer origója a Föld középpontja.

2. *Földrajzi*: egy pontot a (hosszúság, szélesség) koordinátapárral azonosítunk, ahol
- *hosszúság* ( $\lambda$ ): a pont meridiánjának a Greenwich-i kezdő meridiánnal bezárt szöge. Értéke  $-180^\circ$  (nyugati hosszúság) és  $+180^\circ$  (keleti hosszúság) között változik.
  - *szélesség* ( $\varphi$ ): a pontból az Egyenlítőre bocsátott merőleges szakasz középponti szöge. Értéke  $-90^\circ$  (déli szélesség) és  $+90^\circ$  (északi szélesség) között változik.

*Szélességi kör, paralelkör*: azonos szélességi koordinátájú pontok együttese. (Nem főkör, tehát két pont között nem a legrövidebb utat adja.)

*Meridián*: azonos hosszúsági koordinátájú pontok együttese.

### 2.2.4. Vetületi rendszerek

*Vetületi rendszer*: egy  $V: (\lambda, \varphi) \rightarrow (x, y)$  leképezés, amely a földfelszín minden pontjának a síkbeli Descartes koordinátarendszer egy pontját felelteti meg.

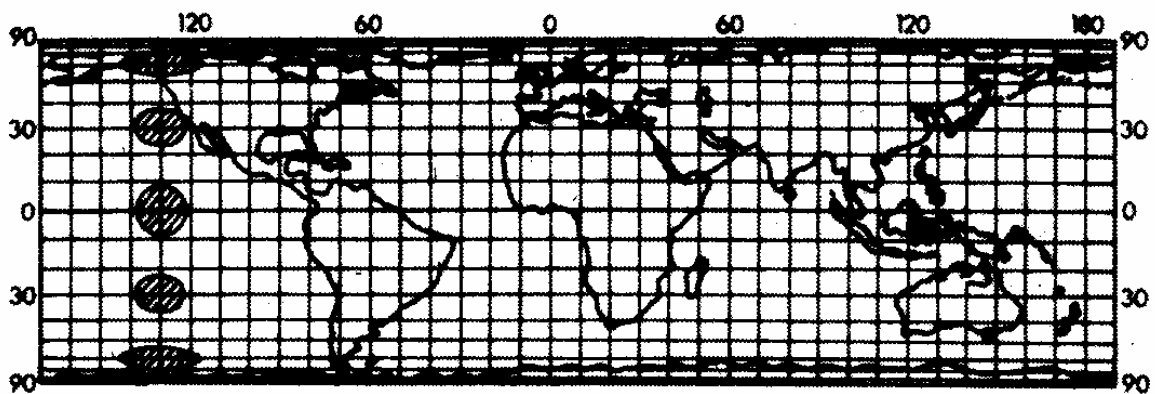
Megjegyzés: a geodéziában a függőleges koordinátákat jelölik  $x$ -szel és a vízszintest  $y$ -nal. Mi azonban a matematikai konvenciót alkalmazzuk, vagyis  $x$  a vízszintes és  $y$  a függőleges koordináta tengely.

*Perspektív vetület*: vetítésugarakkal történő vetítéssel előállítható.

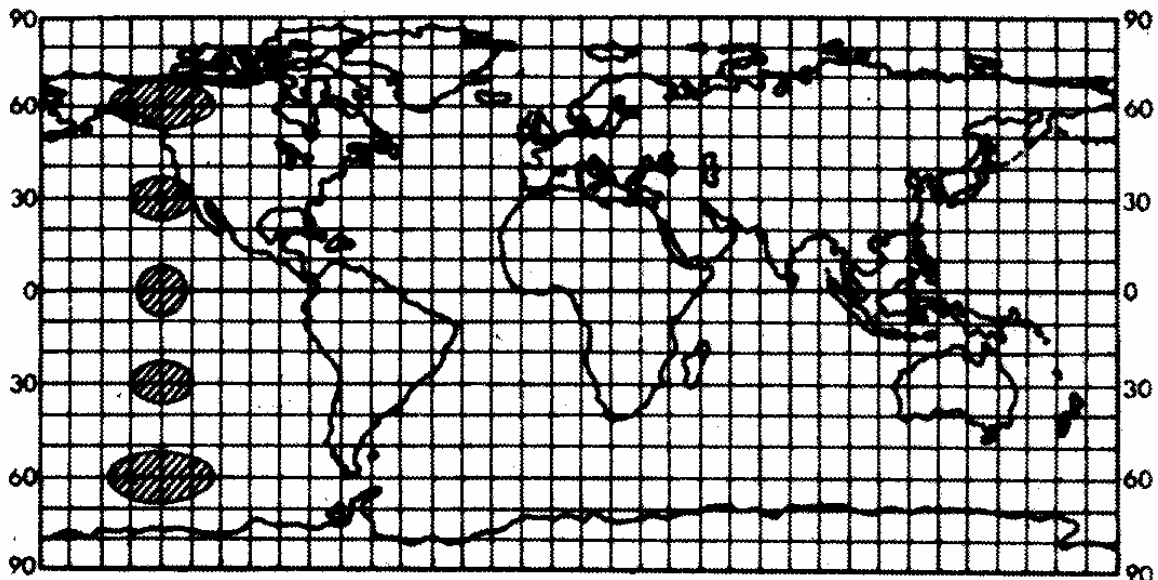
*Nem perspektív vetület*: nem állítható elő vetítésugarakkal.

*Elérendő tulajdonságok (invariánsok)*:

- területtartás (5. ábra),
  - hossztartás (csak egyes vonalak mentén lehetséges) (6. ábra),
  - szögtartás (navigációs célokra) (7. ábra).
- A fenti tulajdonságok természetesen egyidejűleg nem teljesülhetnek.



5. ábra: Területtartó (Lambert-féle) vetület.

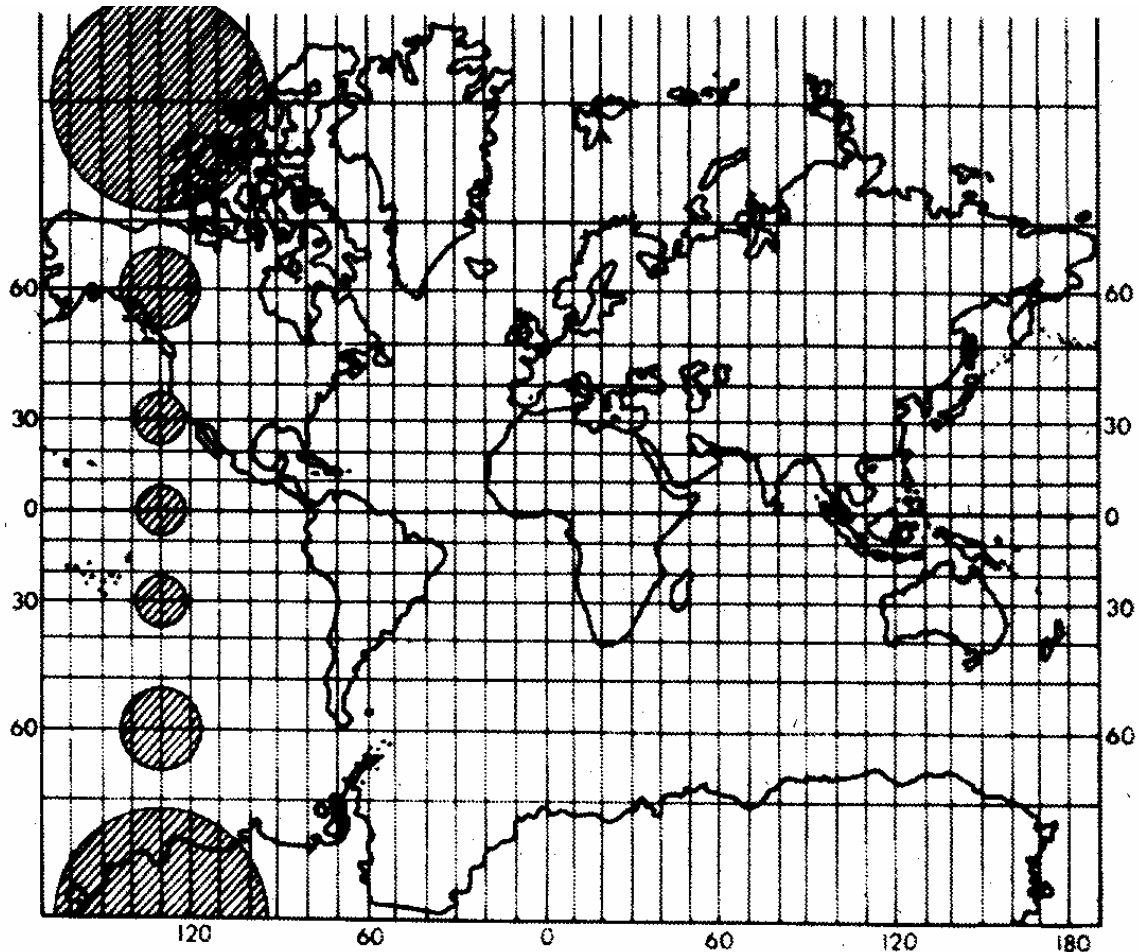


6. ábra: Meridiánban hossztartó vetület.

*Tissot féle indikátrix:* a gömb felszínére rajzolt kisméretű kör. Vetületi torzulások vizsgálatára szolgál: az indikátrixot mozgatjuk a gömb felszínén, és mérjük a képének alakulását (5., 6., 7. ábrák).

*Típusok:*

- *síkvetület:* a gömbfelületet közvetlenül síkra képezzük le.
- *hengervetület:* a gömbfelületet előbb egy hengerfelületre képezzük le, majd azt egy egyenes mentén felhasítva síkba terítjük ki.
- *kúpvetület:* a gömbfelületet előbb egy kúpfelületre képezzük le, majd azt egy egyenes mentén felhasítva síkba terítjük ki.



7. ábra: Szögtartó (Mercator) vetület.

### A méretarány fogalmának pontosítása

Mivel minden vetületi rendszer torzít, így a térképen mért távolságok és a valós távolságok hányadosa helyzettől és iránytól függ, kismértékű szóródást mutat.

Ezért méretarányon a térképen mért hossz és a vetületi hossz hányadosát értjük, ahol vetületi hosszon a földfelszínnek az adott vetületi rendszer szerinti, kicsinyítés nélküli síkbeli képén mért hossz értendő.

A térképen mért távolságokból tehát a méretarány segítségével csak a vetületi távolságokat kapjuk, a valós távolságok meghatározásához az adott vetületi rendszer torzításait is figyelembe kell venni.

### Sztereografikus vetület

Perspektív síkvetület: centrális vetítés a gömb egy C pontjából a gömb azon érintősíkjára, amely a vetítési középponttal átellenes P pontban érinti a gömböt. A C pont kivételével a gömb valamennyi pontját egyértelműen leképezi a síkra. Szögtartó és körtartó

leképezés, de a C ponton átmenő körök egyenesekre képeződnek le. A P pont közelében a torzítás csekély, a C pont felé haladva rohamosan nő (8. ábra).

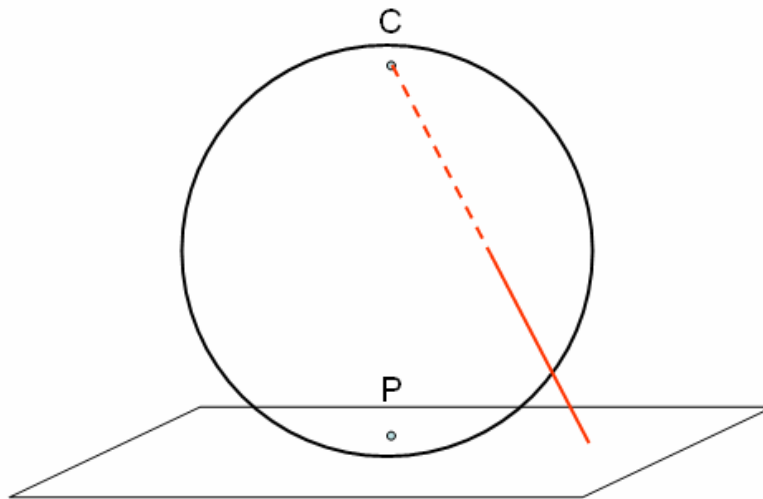
### Mercator vetület

Szögtartó, nem perspektív hengervetület. Egyenletei:

$$x = \lambda$$

$$y = \ln(\operatorname{tg}(\varphi/2 + \pi/4))$$

A szélességi körök vízszintes, a meridiánok függőleges, a loxodrómák általános helyzetű egyenesekbe mennek át. Nem területtartó, a pólusok felé haladva a területek erősen növekednek. (A Tissot-féle indikátrix mindig kör marad, de területe a pólusok felé haladva nő.)

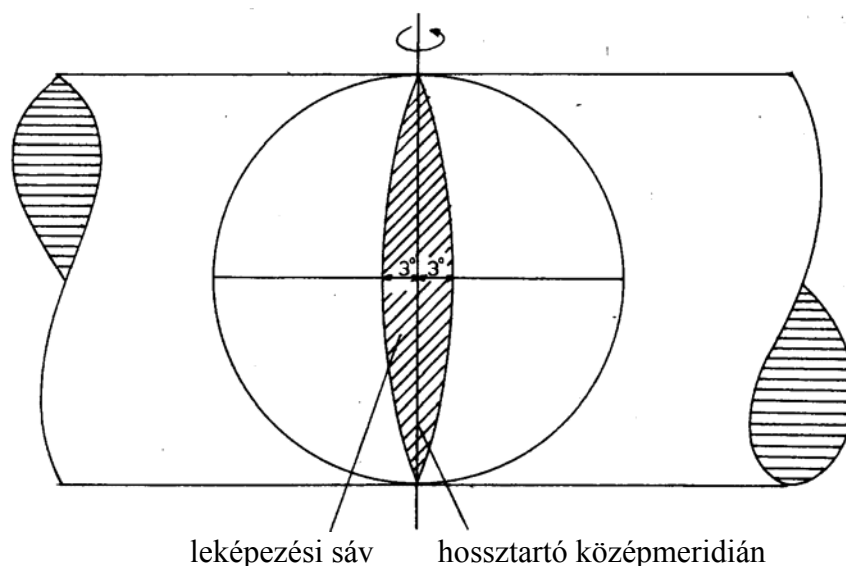


8. ábra: Sztereografikus vetület.

### A Mercator vetület alkalmazásai

*GK = Gauss-Krüger vetület:* Elsősorban Kelet-Európában használatos. A Föld alakját ellipszoiddal modellezi (ún. Kraszovszkij-féle ellipszoid). Az ellipszoid felszínét 6 fokként (nagyobb méretarány esetén 3 ill. 2 fokként) meridiánokkal zónákra (ellipszoid kétszögekre) osztja. Minden egyes zóna esetén egy transzverzális helyzetű elliptikus hengerre Mercator vetítést alkalmaz úgy, hogy a vetítési henger a zóna középmeridiánjánál érinti a felszínt. (9. ábra). A vetület szögtartó, és az érintő meridián mentén hossztartó.

*UTM = Univerzális Transzverzális Mercator vetület:* A GK-hoz hasonló rendszer, a világon általánosan használják. A Föld alakját szintén ellipszoiddal közelíti (ún. Hayford-féle ellipszoid), melynek felszínét 6 fokként meridiánokkal zónákra osztja. A GK-hoz hasonlóan zónánként transzverzális Mercator vetítést alkalmaz, de úgy, hogy a henger a sarkoknál érinti, egyébként metszi a felszínt. A vetület szögtartó, és a két metsző meridián mentén hossztartó.



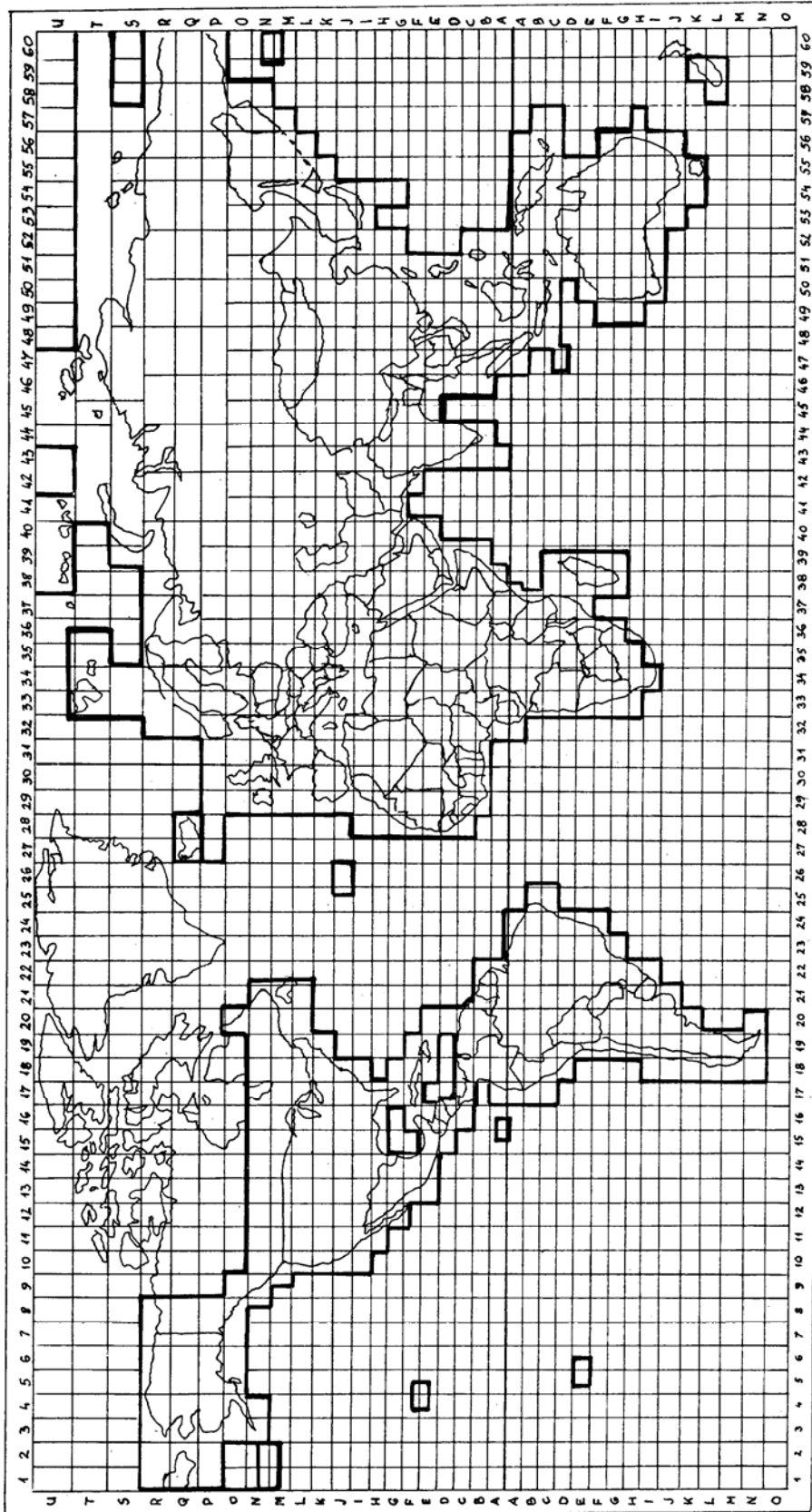
9. ábra: A Gauss-Krüger vetület származtatása.

Az UTM és GK vetületekhez az egész Földre kiterjedő egységes szelvényezés tartozik. A zónákat szélességi övekre osztják: UTM esetén 8, GK esetén 4 szélességi fokként. A zónák találkozásánál fellépő elcsúszásokat átfedésekkel küszöbölik ki. (Pl. ha a térképezendő terület két zóna határára esik, akkor a domináns zóna kiterjesztésével térképezik.)

*IMW (=International Map of the World):* 1:1.000.000 méretarányú, részben UTM, részben Gauss-Krüger vetületű, a Föld teljes felszínét lefedő térképrendszer. Létrehozását 1891-ben határozták el, minden ország a saját területét térképezi. Egy szelvény általában egy  $6^\circ \times 4^\circ$  területet ábrázol. A Földet 2000 szelvény fedi le, ebből 800 a szárazföld (10. ábra). A vetületi rendszerből adódóan a szelvények nem téglalap alakúak.

*DCW (=Digital Chart of the World):* 1:1.000.000 méretarányú, vektoros digitális térkép, az IMW digitális változatának tekinthető.

Magyarországot négy egymilliós szelvény fedi le, ezek jele L-33, L-34, M-33, M-34. A hazai Gauss-Krüger rendszerű (általában topográfiai) térképek az egymilliós szelvények továbbosztásával készültek.



10.ábra: Az egymilliós világtérképmű szelvényezése.

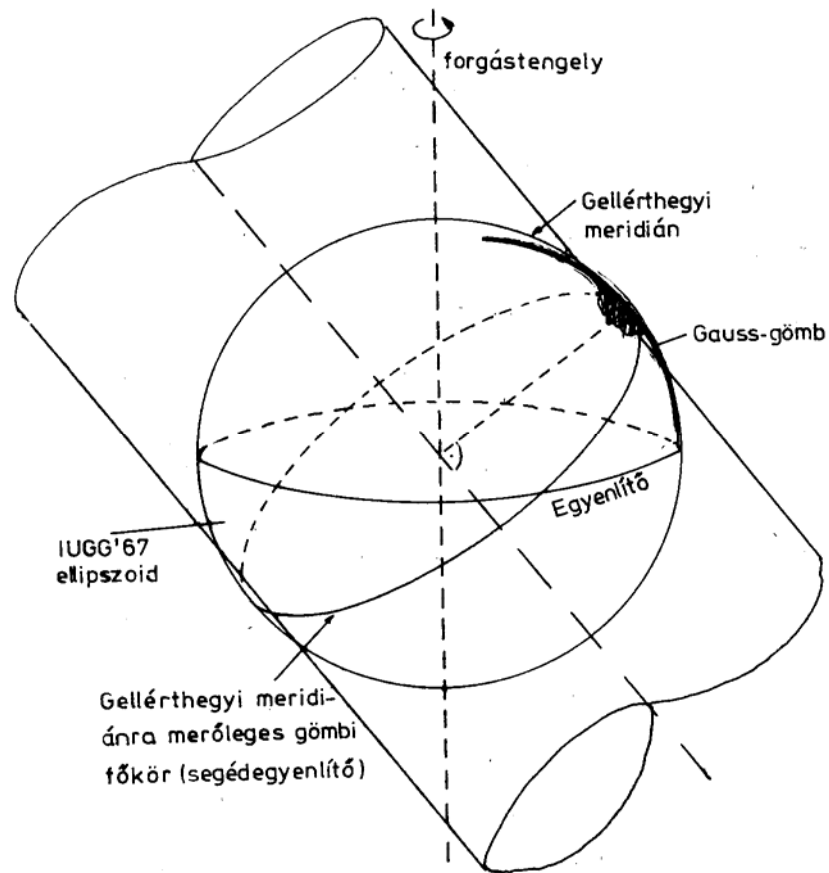
### Budapesti sztereografikus rendszer

A vetítési sík a Gellért-hegy egy meghatározott pontjában érinti a Gauss-gömböt. 127 km sugarú körben 1/10.000-nél kisebb hossztorzulást biztosít. Legnagyobb hossztorzulás Szabolcs-Szatmár megyében lép fel (kb. 4/10.000).

### Egységes Országos Vetületi rendszer (EOV)

1975 óta használatos vetület. Olyan vetületi rendszer, amely Magyarország területét egységesen és minimális torzulással kezeli. Az ellipszoidról előbb Gauss-gömbre, majd hengerfelületre vetítenek (11. ábra). A síkbeli koordinátarendszer függőleges tengelye a Gellérthegyen áthaladó meridiánnak, vízszintes tengelye az ország középvonalánál a gellérthegyi meridiánra merőleges gömbi főkörnek felel meg. A vetítési henger erre a főkörre illeszkedik (11. ábra).

Szögtartó vetület. A hossztorzulás az ország egész területén 1/30.000.000 alatt marad. Koordináta egység = 1 m, a szelvények téglalap alakúak. A koordinátarendszer kezdőpontja a vetítési középponttól 200 km-rel délre, 650 km-rel nyugatra van, így minden koordináta pozitív, és az x, y koordináták sem teveszthetők össze, mert  $x > 400.000 > y$  minden esetben teljesül.





## 11. ábra: Az EOVS vetület származtatása.

*EOVS (Egységes Országos Térképezési Rendszer):* az EOVS-re épülő térképezés. Az országot 83 db 1:100.000 méretarányú szelvény fedi le, ezek továbbosztásával adódnak a nagyobb méretarányú szelvények. A földmérési alaptérképek 1:1000 ill. 1:2000 méretarányban mutatják a beépített területeket, 1:4000 méretarányban lefedik a külterületeket. Szokásos szelvény méret: 50 x 75 cm. Az EOVS un. topográfiai térképsorozata 1:10.000, 1:25.000 ill. 1:100.000 méretarányú térképeket tartalmaz.

## 2.3. Transzformációk

Vetületi rendszerek közötti átszámítás az egyes rendszerek egyenletei alapján történik.

Gyakran találkozunk azonban olyan feladattal, hogy ismeretlen vetületi rendszerű vagy torzított T képet (például szkennelt térképet vagy légifényképet) kell adott vetületi rendszerbeli T' térképpé transzformálni. Ilyenkor *kontrollpontok* megadása szokásos, vagyis a T képen kijelölünk valamely jól azonosítható  $(x_1, y_1), \dots, (x_m, y_m)$  pontokat, amelyekhez meg tudjuk adni, hogy a transzformációnak ezeket az  $(x_1', y_1'), \dots, (x_m', y_m')$  pontokba kell leképeznie. Az eljárás az alábbi lépésekből áll:

- transzformáció típusának kiválasztása,
- transzformáció együtthatóinak számítása a kontrollpontokból,
- transzformáció elvégzése.

A továbbiakban ezeket a lépéseket részletezzük.

### 2.3.1. Transzformáció típusok

*Affin transzformáció:* a leggyakrabban alkalmazott eljárás, lényegében egy eltolással kiegészített lineáris transzformációnak felel meg:

$$\begin{aligned}x' &= a_0 + a_1 \cdot x + a_2 \cdot y \\y' &= b_0 + b_1 \cdot x + b_2 \cdot y\end{aligned}$$

A *Helmert-transzformáció* az affin transzformáció speciális esete, amely eltolás,  $\alpha$  szögű elforgatás és k-szoros nagyítás/kicsinyítés (azaz méretarány változtatás) együttesét jelenti:

$$\begin{aligned}x' &= a_0 + a_1 \cdot x - a_2 \cdot y \\y' &= b_0 + a_2 \cdot x + a_1 \cdot y\end{aligned}$$

ahol  $a_1 = k \cdot \cos \alpha$  és  $a_2 = k \cdot \sin \alpha$ , az eltolást pedig  $(a_0, b_0)$  jelenti.

A *polinomiális transzformációk* az affin transzformáció magasabb fokú általánosításai, általában r-edfokú polinommal adottak, például  $r = 3$  esetén:

$$\begin{aligned}x' &= f_x(x, y) = a_{00} + a_{10}x + a_{01}y + a_{20}x^2 + a_{11}xy + a_{02}y^2 + a_{30}x^3 + a_{21}x^2y + a_{12}xy^2 + a_{03}y^3 \\y' &= f_y(x, y) = b_{00} + b_{10}x + b_{01}y + b_{20}x^2 + b_{11}xy + b_{02}y^2 + b_{30}x^3 + b_{21}x^2y + b_{12}xy^2 + b_{03}y^3\end{aligned}$$

### 2.3.2. Együtthatók meghatározása

Adottak az  $(x_1, y_1), \dots, (x_m, y_m)$  kontrollpontok és  $(x_1', y_1'), \dots, (x_m', y_m')$  képei. Olyan transzformációt keresünk, amely a kontrollpontokat minél pontosabban képezi le. Egy  $r$ -edfokú transzformáció együtthatóinak egyértelmű meghatározásához  $m_r = (r+1)(r+2)/2$  kontrollpont szükséges ( $r = 1, 2, 3, 4, 5$  esetén rendre  $m_r = 3, 6, 10, 15, 21$ ). Ha ennél több kontrollpont van, akkor a transzformáció csak közelítő leképezést biztosít. Ilyenkor a *legkisebb négyzetek módszerét* alkalmazzák, vagyis az  $f$  transzformáció együtthatóit úgy határozzák meg, hogy a hibák négyzetösszege minimális legyen:

$$\Sigma (f_x(x_i, y_i) - x_i')^2 + (f_y(x_i, y_i) - y_i')^2 \rightarrow \text{minimális}$$

Alább az együtthatók meghatározásának általános módját tárgyaljuk, de a konkrétság kedvéért a formulákat az  $r=3$  esetre írjuk fel. Tehát az alábbi formulákban az  $a_i$  és  $b_i$  együtthatókat keressük:

$$\begin{aligned} x' &= a_{00} + a_{10}x + a_{01}y + a_{20}x^2 + a_{11}xy + a_{02}y^2 + a_{30}x^3 + a_{21}x^2y + a_{12}xy^2 + a_{03}y^3 \\ y' &= b_{00} + b_{10}x + b_{01}y + b_{20}x^2 + b_{11}xy + b_{02}y^2 + b_{30}x^3 + b_{21}x^2y + b_{12}xy^2 + b_{03}y^3 \end{aligned}$$

Tekintsük most az alábbi  $m \times m_r$ -es  $A$  mátrixot ( $m_r$  az együtthatók száma,  $r = 3$  esetén  $m_r = 10$ ):

$$A = \begin{bmatrix} 1 & x_1 & y_1 & x_1^2 & x_1 y_1 & y_1^2 & x_1^3 & x_1^2 y_1 & x_1 y_1^2 & y_1^3 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_m & y_m & x_m^2 & x_m y_m & y_m^2 & x_m^3 & x_m^2 y_m & x_m y_m^2 & y_m^3 \end{bmatrix}$$

Legyen  $D = (A^T \cdot A)^{-1} \cdot A^T$ , ekkor az együtthatók számítása:

$$\begin{bmatrix} a_{00} \\ a_{10} \\ \vdots \\ a_{m_r} \end{bmatrix} = D \cdot \begin{bmatrix} x_1' \\ x_2' \\ \vdots \\ x_m' \end{bmatrix} \qquad \begin{bmatrix} b_{00} \\ b_{10} \\ \vdots \\ b_{m_r} \end{bmatrix} = D \cdot \begin{bmatrix} y_1' \\ y_2' \\ \vdots \\ y_m' \end{bmatrix}$$

A fentiek részletes levezetését a Függelék tartalmazza.

### 2.3.3. A transzformáció számítása

Egy T input adatstruktúrából egy T' output adatstruktúrát kell képezni. Más eljárást kell követnünk vektoros ill. raszteres adatok esetén.

*Vektoros adatok* esetén az adatstruktúrában tárolt minden x, y koordinátaértéket a megfelelő  $f_x(x, y)$  és  $f_y(x, y)$  értékekkel helyettesítünk. Amennyiben az outputként egy meghatározott kivágatot (téglalap alakú területet) képezünk, úgy gondoskodni kell a kivágatból kilógó rajzelemek vágásáról.

*Raszteres adatok* esetén a transzformáció *inverzével* célszerű számolni. Itt ugyanis a T' mátrix elemein haladunk végig, és minden (x', y') koordinátájú pixel ősenek T-beli (x, y) koordinátáit számítjuk. Ez a koordináta általában nem egész értéként adódik, ilyenkor valamilyen algoritmussal el kell dönteni, hogy milyen pixel értéket választunk (átmintázás). Három jellemző módszer:

a). Legközelebbi szomszéd (nearest neighbor) választása: az (x, y) valós koordinátájú ponttal szomszédos négy egész koordinátájú pixel közül a legközelebbit választjuk. Ez a gyakorlatban azt jelenti, hogy az x, y valós számokat egész értékre kerekítjük (ha a törtrész 0.5-nél kisebb, akkor lefelé, egyébként felfelé kerekítünk). Az eredeti és transzformált kép színmélysége (ill. szürkefokozatok száma) megegyezik.

b). Bilineáris interpoláció: ha a (x, y) pont a  $P_1, P_2, P_3, P_4$  rácspontok közé esik, akkor az (x, y) ponton át húzott vízszintes és függőleges egyenesekkel az egységnégyzetet négy téglalpra osztjuk, a  $P_i$  pont  $p_i$  értékét a vele szembe eső téglalap  $t_i$  területével súlyozzuk ( $p = \sum_i p_i t_i$ ). A transzformált kép színmélysége (ill. szürkefokozatok száma) nagyobb lehet, mint az eredeti képé volt, például bináris képből szürkeárnyalatos keletkezik.

c). Köbös konvolúció: az (x, y) pont  $4 \times 4$  pixeles környezete alapján számolunk.

A köbös konvolúció számításának módja: legyen  $i_0 = \text{int}(x)$ ,  $j_0 = \text{int}(y)$ , ahol „int” egész részt jelent: A pixel interpolált értéke

$$p = \sum_{i=-1..2} \sum_{j=-1..2} P(i_0+i, j_0+j) * f(d(i_0+i, j_0+j) - i)$$

ahol

$P(i, j)$  az (i, j) pixel értéke az input képen,

$d(i, j)$  az (i, j) pixel távolsága az (x, y) ponttól,

az  $f(x)$  függvény értéke

$$(a+2)|x|^3 - (a+3)|x|^2 + 1 \quad \text{ha } |x| < 1$$

$$a|x|^3 - 5a|x|^2 + 8a|x| - 4a \quad \text{ha } 1 < x < 2$$

$$0 \text{ egyébként}$$

Megjegyzés: az ERDAS rendszernél  $a = -0.5$  konstanst alkalmaznak, más rendszereknél más értéket választhatnak.

## 2.4. A térképezés alapjai

*Térképezés (térkép készítés):* a Földre vonatkozó adatok mérése, összegyűjtése, rendszerezése grafikus ábrázolás céljára. A térképezés módjai:

- terepfelmérés,
- távérzékelés.

### 2.4.1. Terepfelmérés

*Geodézia (földmérés):* A Föld alakjával és méreteivel, felületének és egyes részeinek felmérésével, valamint földrajzi helymeghatározással foglalkozó tudomány.

(Ne tévesszük össze a geográfia (földrajz) és a geológia (földtan) fogalmával!)

#### Helymeghatározás

*Vízszintes mérés:* egy földfelszíni pont földrajzi koordinátáit határozza meg (szélesség, hosszúság). Országokra, kontinensekre kiterjedő méréseket a forgási ellipszoidra vonatkoztatják, 50 km<sup>2</sup>-nél kisebb területek esetén a méréseket gömbre, egész kis terület (pl. egy település) esetén síkfelületre vonatkoztatják.

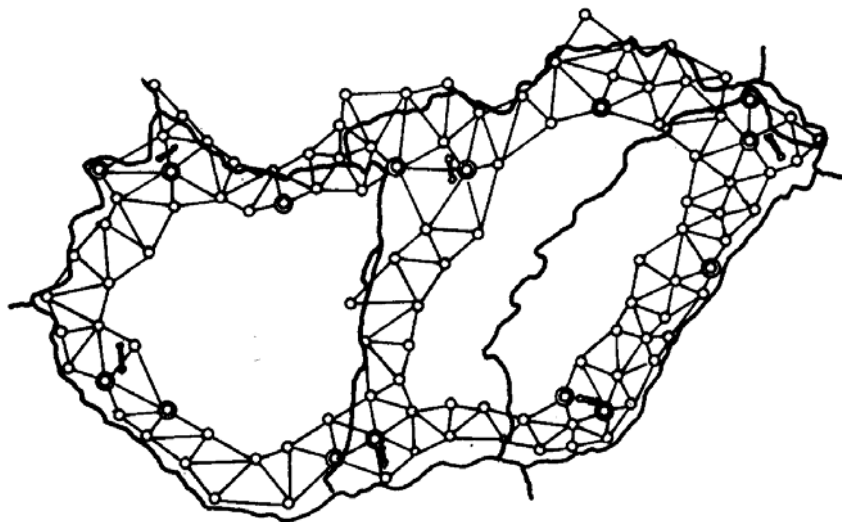
*Magasságmérés:* a földfelszíni pontnak a geoidtól mért távolságát határozza meg (tengerszint feletti magasság).

*Földmérési alappontok:* ismert koordinátájú, fizikailag állandósított pontok. Háromszögrácsot alkotnak, az oldalhossz

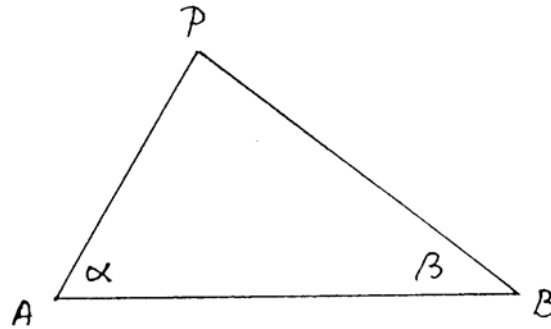
- elsőrendű pontok esetén 30 km (12. ábra),
- másodrendű pontok esetén 15 km,
- harmadrendű pontok esetén 7 km,
- negyedrendű pontok esetén 2 km.

#### Hagyományos mérési módszerek:

A számos mérési módszer közül itt csak a *háromszögelést* emeljük ki, amikor például a terepen egy P pont koordinátáinak meghatározása az ismert koordinátájú A, B pontokban mért  $\alpha = \text{PAB}$  és  $\beta = \text{ABP}$  szögek segítségével történik (13. ábra). Szögmérésre általában *teodolitot* használnak.



12. ábra: Magyarország elsőrendű háromszögelési hálózata.



13. ábra: Háromszögeléssel történő földmérés elve.

#### GPS alapú mérés:

GPS = *Global Positioning System*: műholdak segítségével történő helymeghatározás. A rendszer alapvetően 24 db NAVSTAR műholdból áll (az USA hadseregének tulajdona), a Föld bármely pontján legalább 4 mindig látható. A Föld felszínén működtetett GPS vevő készülék a Föld bármely pontján képes meghatározni a helyzetét a műholdakról vett jelek alapján.

A mérés elve a háromszögelés (13. ábra) térbeli változatának tekinthető, amelyet a GPS vevőben egy feldolgozó szoftver végez el a műholdak által sugározott idő- és pályaelem adatok alapján. Az egyszerűbb vevők csak *kódmérést* végeznek, vagyis a műhold által sugárzott kódot fejtik meg. A pontosabb vevők *fázismérést* is végeznek, vagyis a beérkezett műholdjel egész ciklusait és fázisát is mérik.

A GPS háromdimenziós, globális koordinátarendszerben adja a koordinátákat, ezt át kell számolni az aktuálisan használt vetületi rendszerbe.

Az Amerikai Védelmi Hivatal 1990-ben bevezette az SA (= Selective Availability) rendszert, amelynek eredményeképpen a műholdak mesterségesen elrontott jeleket sugároznak, ezzel csökkentve a polgári célú GPS mérési pontosságát. A teljes pontosság csak katonai célokra (és kiválasztott polgári felhasználók számára) volt elérhető. Ma már az SA-t megszüntették.

Többféle GPS mérési módszer van:

- *Abszolút* helymeghatározás esetén csak egy vevővel mérnek.
- *Relatív* (vagy *differentiális*) helymeghatározás esetén két, egymástól legfeljebb 10 km távolságra lévő GPS vevővel dolgoznak: az egyik egy ismert koordinátájú ponton áll (*referencia állomás*), a másik pedig a mérendő ponton (*felhasználó*). Ez a módszer lényegesen pontosabb eredményt szolgáltat, mint az egy vevővel történő (abszolút) helymeghatározás.

Beszélhetünk továbbá statikus és kinematikus mérésről:

- *Statikus* esetben a mérés teljes időtartama alatt a vevő egy helyben áll.
- *Kinematikus* esetben a vevő mérés közben folyamatosan mozog (pl. járművön).

A koordinátaszámítás is kétféleképp történhet:

- *Valós idejű* (*real time*) mérés esetén a vevő azonnal, a mérés helyszínén határozza meg a pont helyzetét.

– *Utófeldolgozás* esetén a vevő csak letárolja a mérések adatait, amelyekből utólag egy számítógépes program határozza meg a tényleges koordinátákat. Az utófeldolgozás olcsóbb műszerrel gyorsabb mérést tesz lehetővé.

A GPS két fő alkalmazási területe a navigáció és a geodézia.

*Navigáció:* hajó, repülőgép helyzetének meghatározásához egyetlen vevőt használnak, amely mozgás közben folyamatosan mér (abszolút, kinematikus mérés). Pontossága 10-20 m. Ennél pontosabb helymeghatározáshoz két vevő szükséges (valós idejű, relatív mérés), például városi közlekedésben gépjármű aktuális helyének digitális térképen való megjelenítéséhez alkalmazható. Sajátos alkalmazást jelentenek a *járműpark figyelő rendszerek*: itt minden jármű saját GPS-vevővel rendelkezik, aktuális pozícióját folyamatosan beküldi egy diszpécser központba, ahol az egy digitális térképen megjelenik.

*Geodézia:* általában statikus, relatív mérést alkalmaznak. A pontosság néhány cm, de egy pont méréséhez hosszabb idő szükséges, ami utófeldolgozással csökkenthető. A GPS-szel történő mérés költsége kb. fele a hagyományos módszernek, és számos egyéb előnnyel is jár (pl. a pontok között nem szükséges összelátás, időjárási viszonyok (pl. köd) kevésbé zavarják).

A mérési idő csökkenthető az ún. *fél-kinematikus módszerrel*. Ennél mind a referencia állomás, mind a vevő folyamatosan végez méréseket, de amíg a referencia állomás egy helyen áll, addig a felhasználó pontról pontra jár, és minden ponton csak rövid ideig mér. Mindez csak nyitott terepen érvényes, ahol megszakítás nélkül, folyamatosan vehetők a műholdak GPS jelei (nincs takarás, pl. épületek vagy fák miatt). Ha a műhoddal a kapcsolat egy pillanatra megszakad, a mérést előlről kell kezdeni.

GPS vevőkészülékek ára széles skálán mozog, az olcsó készülékektől a milliós értékű berendezésekig.

### **Térképkészítés**

*Hagyományos:* A terepen vázlatrajz készítése, a szükséges pontok koordinátáinak bemérése, mérési jegyzőkönyv készítése. A mérési jegyzőkönyv alapján rajzasztalon készül a pontos térkép.

*Számítógéppel segített:* a terepen a bemért koordinátákat hordozható számítógépbe viszik, például relációs adatbázis formájában, amely a ponthoz tartozó objektum típusát, ill. az objektumok kapcsolatát is tartalmazza. Például egy rekord lehet: " $x_1, y_1, x_2, y_2$ , telekhatár". A mérési adatbázisból egy alkalmas illesztőprogram vektoros térképi adatstruktúrát állít elő, amely CAD rendszerrel kezelhető.

### **2.4.2. Távérzékelés**

*Távérzékelés:* légi- és űrfelvételek készítése ill. feldolgozása. Célja: térképkészítés, vagy speciális elemzések végzése (pl. mezőgazdasági termésmennyiség becslés).

A műholdképek rendszerint *több sávban* készülnek. Ez azt jelenti, hogy az elektromágneses spektrumban sávokat különítenek el, és ugyanazon területről több képet készítenek az egyes sávokban visszaverődött fényt (sugarakat) rögzítve.

*Fotogrammetria:* légifényképek kiértékelése helymeghatározás, távolság- és magasságmérések céljából.

### Egyképes (sík) fotogrammetria

Főleg síkvidéken alkalmazzák. Gyakran a légifényképet a meglévő térképre vetítve annak aktualizálására használják, ehhez a légifényképet a térképpel fedésbe kell hozni. A kamera dőlésszögéből és a perspektívikus torzításból eredő hibát optikai berendezéssel, vagy digitális légifényképen számítógépes algoritmussal lehet korrigálni. Nagyobb területet ábrázoló kép (műholdkép) esetén a transzformációnál a térképészeti vetületet is figyelembe kell venni.

A számítógépes képtranszformáció gyakori módja a *kontrollpontokra* épülő eljárás. Itt olyan tereptárgyakat választanak ki, amelyek a légifényképen és a térképen egyaránt jól azonosíthatók (pl. jellegzetes épületek, tornyok), és ezeket manuálisan egymáshoz rendelik. Vagyis,  $n$  kontrollpont esetén a légifénykép adott  $P_1, \dots, P_n$  pontjait a térképen adott  $Q_1, \dots, Q_n$  pontoknak kell megfeleltetni. A kontrollpontok alapján affin vagy polinomiális transzformáció alkalmazható. Az affin transzformáció pontosabb, ugyanakkor lényegesen gyorsabb, és kevésbé érzékeny a kontrollpontok hibás megadására.

Dombos, hegyes terepen egyképes fotogrammetria csak akkor alkalmazható, ha rendelkezésre áll a terület domborzatmodellje. Ez utóbbin olyan modellt értünk, amely segítségével bármely  $(x, y)$  koordinátájú pontban ki lehet számítani a  $z$  terepmagasságot. A domborzatmodell segítségével *ortofotó* (merőleges vetítésű sugarakkal előállított kép) készíthető. Az eljárás egy lehetséges módja: a térképre egy rácshálózatot illesztünk, és minden egyes rácsponthoz kiszámítjuk, hogy az hová esik a légifényképen (Kraus, 1998). Ezután kontrollpont alapú transzformációval a torzított rácsot (légifénykép) szabályos rácsba visszük át (ortofotó).

Magasságmérés – pl. torony esetén – egyképes fotogrammetria esetén is lehetséges, ha a nadirpontot (felvétel alatti pontot) ismerjük (14. ábra).

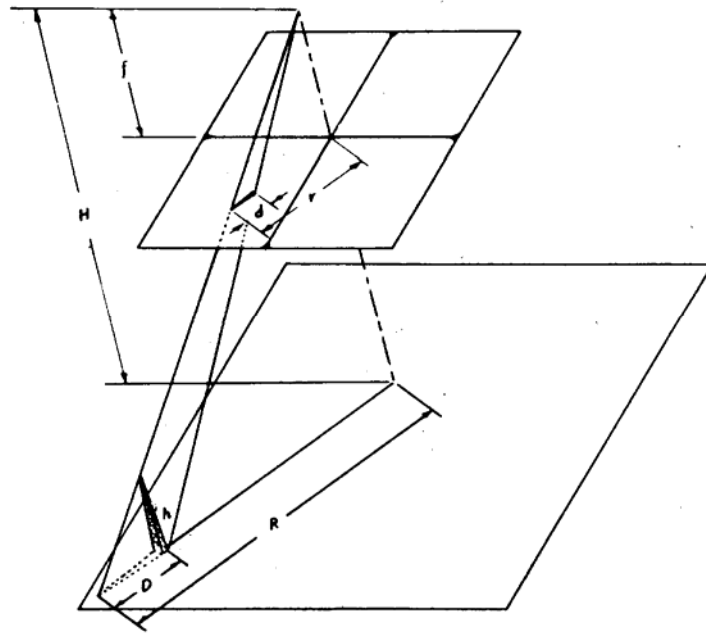
$h$  – a tárgy magassága

$H$  – a repülési magasság

$d$  – a fényképen a tárgy aljának és tetejének távolsága

$r$  – a fényképen a tárgy tetejének a nadirponttól mért távolsága

$$\frac{h}{H} = \frac{D}{R} = \frac{d}{r} \quad \text{innen} \quad h = \frac{d \cdot H}{r}$$



14. ábra: Magasságmérés egyképes fotogrammetriával.

### **Kétképes (tér) fotogrammetria:**

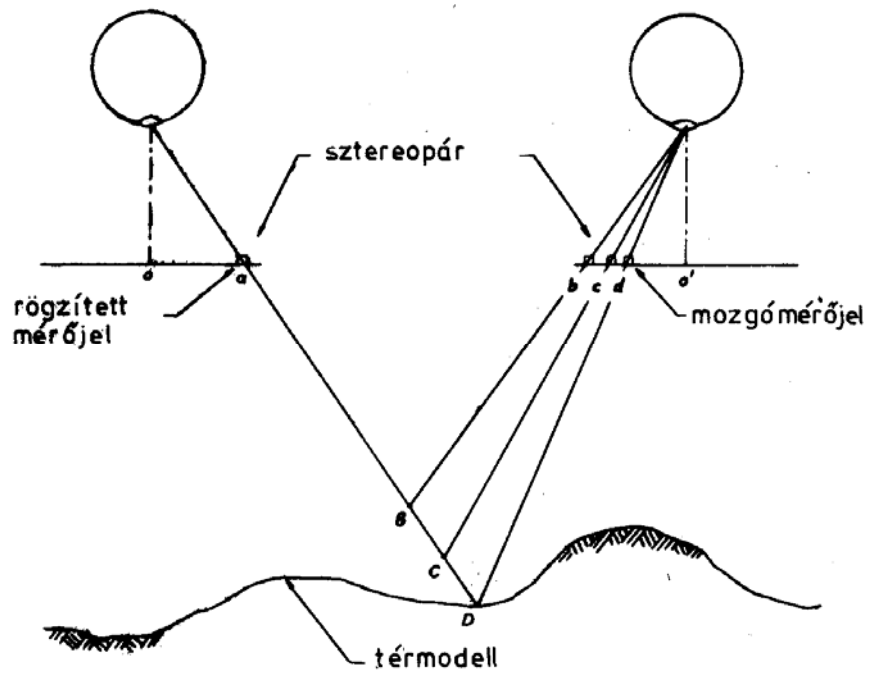
Sztereo képpáron az egymásnak megfelelő objektumok összerendelésével meghatározhatók azok (magassági) koordinátái. Feldolgozás történhet:

- Hagyományos optikai berendezéssel (sztereoszkóp, sztereoplotter). A feldolgozás az emberi térlátás képességének kihasználására épül: A műszerbe épített mérőjel állításával (15. ábra) határozzák meg az egyes objektumok magasságát. Elsősorban domborzat térképezésére használják (szintvonalas térkép).

- Sztereo számítógép monitorral (pl. Intergraph Image Station rendszer). A monitor gyorsan váltakozva jeleníti meg a sztereo képpár bal ill. jobb komponensét. Ehhez egy speciális szemüveg tartozik, amely azonos frekvenciával kapcsolja ki-be a bal és jobb szemoldalt. A berendezés segítségével a felhasználó térben látja a képet, és a sztereoszkóphoz hasonló elven tudja meghatározni a tereptárgyak magasságát.

- Automatikusan: olyan szoftverrel, amely a sztereo képpárból algoritmikusan 3D pontrácsot számol.





15. ábra: A mérőjel elve.

Domborzatmodell előállításának legkorszerűbb módja a *lézerszkennő*. A berendezés repülőgépről lézersugárral tapogatja le a felszínt, és igen pontos terepmodellt ad. Ez egyben hátránya is lehet: mivel észleli a növényzetet és minden épületet, tereptárgyat, ezért nem a geológiai értelemben vett domborzatot állítja elő.

## 3. VEKTOROS TÉRINFORMATIKAI RENDSZEREK

### 3.1. A modellalkotás folyamata

Ahhoz, hogy a létező világ jelenségeit és folyamatait a számítógépre le tudjuk képezni, modellalkotásra van szükség. Ennek három szintjét szokták megkülönböztetni (általában, nem csak a térinformatikában):

*Elvi modell:* ezen a szinten a számunkra fontos entitásokat, kapcsolatokat és folyamatokat próbáljuk megragadni. Az adatbázisok világában erre a célra szolgál az *egyed-kapcsolat modell*, az információs rendszereknél pedig az *SSADM* (Structured Systems Analysis and Design Method) vagy az *UML* (Unified Modeling Language). Ez a modellezési szint még független a konkrét implementációtól. Az elvi modell gyakran több hierarchiaszintre osztható.

*Logikai modell:* lényegében absztrakt adatstruktúrák modellezési szintje, a modell alapegységeit gyakran *objektumoknak* nevezik. Adatbázisok esetén rendszerint *relációs adatmodellt* használnak (relációsémák, elsődleges kulcsok és idegen kulcsok), amelyre SQL nyelvű leírás alkalmazható.

*Fizikai modell:* a tényleges gépi adatkezelés szintje. Ezt a szintet az egyes alkalmazói szoftverek többé-kevésbé eltakarják a felhasználó elől.

A vektoros térinformatikát továbbiakban a *logikai modell* szintjén vizsgáljuk. Az alábbi fogalmakat használjuk:

– *Térbeli adatbázis (spatial database):* részben vagy egészben térbeli (térképi) vonatkozású adatok rendszere. Egy alkalmazás által kezelt valamennyi (térbeli és nem térbeli) adat együttesét nevezzük térbeli adatbázisnak.

– *Fedvény (angolul coverage):* tematikusan összetartozó térbeli objektumok együttese. A CAD rendszerbeli rétegfogalom általánosításának tekinthető.

A térbeli objektumok általában két fő komponensből állnak:

– *térbeli komponens*, amelyet grafikusán jelenítünk meg (például telek határvonalai),  
 – *leíró komponens*, amelyet rendszerint táblázatosan jelenítünk meg (például telek adatai).

A továbbiakban először a térbeli komponens vektoros kezelését vizsgáljuk (3.2, 3.3, 3.4. fejezet), majd a két komponens együttes kezelését biztosító adatbázismodelleket tekintjük át (3.5. fejezet).

## 3.2. Vektoros adatmodellek

### 3.2.1. Alapvető objektumtípusok

Az objektumokat térbeli megjelenésük szerint az alábbi csoportokba sorolhatjuk:

(i) *pontszerű (0D) objektum*: térbeli helye általában  $x$ ,  $y$  koordinátával adott. Két típusa van:

– kis méretű objektum: az adott méretarány mellett túl kicsi a grafikus ábrázoláshoz (pl. lámpaoszlop). Megjelenítés: meghatározott jelkulcsi jelöléssel.

– csomópont: vonalas objektumok találkozási helyét jelöli (pl. útelágazás). Rendszerint nincs külön grafikus megjelenítése.

(ii) *vonalas (1D) objektum*: általában vonallánccal adott:  $x_1, y_1, \dots, x_n, y_n$ . (Pl. vasútvonal.) Megjelenítés: adott színnel és vonaltípussal.

Elnevezés: vonalon a továbbiakban általában vonalláncot értünk, amely speciális esetként az egyenesszakaszt is magában foglalja.

(iii) *területi (2D) objektum*: általában poligonnal adott:  $x_1, y_1, \dots, x_n, y_n$ . (Pl.: telek.) Megjelenítés: adott kitöltő mintázattal és/vagy színnel.

Referencia pont: a 2D objektum egy kijelölt pontja, amelyhez pl. felirat rendelhető. Konvex poligon esetén a súlypontot célszerű választani, konkáv esetben azonban ez kívül eshet a poligonon.

#### A méretarány kérdése

A digitális térkép lényegében egy térbeli adatbázis, amely tetszés szerinti nagyításban térképként megjeleníthető ill. nyomtatható, így a méretarány jelentősége itt megváltozik. Egy digitális térkép méretaránya alapvetően két dologra határoz meg:

– *mely objektumtípusokat* tartalmaz az adatbázis (pl. kis méretarányú térkép nem tartalmaz épületeket),

– az adott objektumok *milyen adattartalommal tárolódnak* (pl. egy települést kis méretarány esetén pontszerűen (0D objektum), nagyobb méretarány esetén poligonnal (2D objektum) ábrázolunk).

Ez azt jelenti, hogy egy kis méretarányú digitális térképet nincs értelme nagyobb méretarányban nyomtatni, hisz a szükséges adattartalom hiányzik belőle.

Ha egy területet erősen különböző méretarányokban kell kezelni, akkor két megoldási lehetőség kínálkozik:

(i) Több különálló, egymásra épülő digitális térképet készítünk. Például egy városi GIS rendszer esetén készíthető egy kis méretarányú áttekintő térkép, amely csak a kerületek körvonalait és a főbb utakat tartalmazza, míg a nagy méretarányú térképen már az épületek körvonalai, házszámok, stb. is látható. A módszer hátránya, hogy ha pl. egy kerület határa megváltozik, akkor a változást két különálló adatstruktúrán kell átvezetni.

(ii) A méretarány változtatásával fokozatosan ki/bekapcsoljuk az egyes rajzi rétegek megjelenítését. Például egy országos úthálózat esetén kis méretaránynál csak az országos főútvonalak rétegét kapcsoljuk be, majd a méretarány növelésével fokozatosan bekapcsoljuk a másodrendű és harmadrendű utakat, végül a földutakat is.

### 3.2.2. Spagetti modell

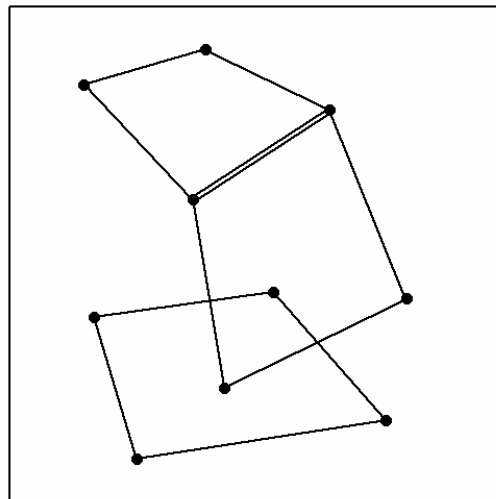
Objektumok egyszerű halmaza, az objektumok között nincs hivatkozási kapcsolat. Ilyen a CAD rendszerek és az egyszerűbb térinformatikai rendszerek adatstruktúrája.

Az adatstruktúra előnye, hogy könnyen kezelhető, egyszerű az adatok karbantartása. Ezért az előnyért viszont számos hátránnyal kell fizetni:

a) Az egymást metsző vonalak metszéspontjában nem feltétlenül van csomópont (a vonalak ilyenkor "nem tudják", hogy metszik egymást, lásd 16. ábra).

b) A szomszédos poligonok (például telkek) határvonala kétszer tárolódik, ami egyrészt redundanciát jelent, másrészt módosításkor zavarokat okozhat.

A fenti jellegű problémák miatt az adatintegritás ellenőrzése, elemzések elvégzése nehézkes és lassú.



16. ábra: a spagetti modell.

Példa: telekosztás.

1. megoldás: egyszerű válaszvonal behúzással. A területszámítás lényegében lehetetlen, mivel a csatlakozó pontoknál nem szakad meg az eredeti vonal.

2. megoldás: snapping vonaloztással. A területszámítás lehetséges, de körülményes.

### 3.2.3. Topológikus modellek

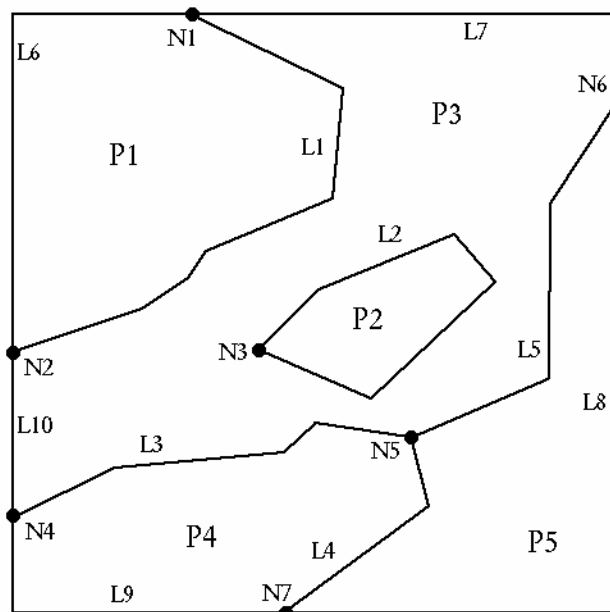
Ha az adatstruktúra nem csak a rajzi objektumokat, hanem azok térbeli kapcsolódási struktúráját (azaz a topológiát) is tartalmazza, akkor topológikus adatmodellről beszélünk. Az ilyen modelleknél általában minden rajzelemnek egyedi azonosítója (*identifier*, röviden *id*) van, ennek segítségével az egyes rajzelemek egymásra hivatkozhatnak.

A következőkben három jellegzetes topológikus adatstruktúrát ismertetünk: a tartománytérképet, a hálózatot és a folytonos felületek leírására használható TIN modellt.

### 3.2.4. Tartománytérkép (folttérkép)

Egy adott területet diszjunkt tartományokkal (foltokkal) hézagmentesen fedünk le (pl. talajtérkép, megyetérkép). Két tartomány határvonalát 1D objektumként, az egyes tartományokat 2D objektumként tároljuk (17. ábra).

Megjegyzés: A gyakorlatban előfordulhat, hogy az ábrázolandó 2D objektumok átfedik egymást (pl. ha a különböző időpontokban történt erdőtüzek területeit ábrázoljuk). Ekkor a metsző területeket önálló 2D objektumként felvéve nyerünk tartománytérkép struktúrát.



17. ábra: tartománytérkép. A csomópontokat  $N_i$ , a vonalakat  $L_i$ , a poligonokat  $P_i$  jelöli.

A tartományok szigeteket tartalmazhatnak, amelyek területe nem tartozik a tartományhoz – hiszen a diszjunkttság csak így teljesül. A szigetek megkülönböztetett figyelmet igényelnek mind az adatstruktúra, mind az algoritmusok tekintetében.

Alább egy tipikus tartománytérkép adatstruktúrát mutatunk be, ehhez hasonlót használ az ArcInfo rendszer is. A 17. és 18. ábrák konkrét példát mutatnak az adatstruktúrára.

NODE: csomópont tömb, egy rekordjának felépítése:

id : csomópont azonosító  
 x, y : koordináták  
 [attribútumok]

LINE: határvonal tömb, egy rekordjának felépítése:

id : vonal azonosító  
 node1 : kezdő csomópont azonosítója  
 node2 : záró csomópont azonosítója  
 [lpoly : baloldali poligon azonosítója]  
 [rpoly : jobboldali poligon azonosítója]  
 $x_1, y_1, \dots, x_n, y_n$  : töréspontok koordinátái  
 [attribútumok]

POLYGON: tartomány tömb, egy rekordjának felépítése:

id; : tartomány azonosító  
 line<sub>1</sub>, ..., line<sub>n</sub> : határvonalak azonosítói  
 [attribútumok]

A fenti adatstruktúrára a 18. ábra ad példát. Figyeljük meg a sziget leírását!

<i>NODE:</i>	<i>id</i>	<i>x</i>	<i>y</i>			
	N1	x <sub>1</sub>	y <sub>1</sub>			
	N2	x <sub>2</sub>	y <sub>2</sub>			
	N3	x <sub>3</sub>	y <sub>3</sub>			
	N4	x <sub>4</sub>	y <sub>4</sub>			
	N5	x <sub>5</sub>	y <sub>5</sub>			
	N6	x <sub>6</sub>	y <sub>6</sub>			
	N7	x <sub>7</sub>	y <sub>7</sub>			
<i>LINE:</i>	<i>id</i>	<i>node1</i>	<i>node2</i>	<i>lpoly</i>	<i>rpoly</i>	<i>x<sub>1</sub>, y<sub>1</sub>, ..., x<sub>n</sub>, y<sub>n</sub></i>
	L1	N1	N2	P3	P1	...
	L2	N3	N3	P3	P2	...
	L3	N4	N5	P3	P4	...
	L4	N5	N7	P5	P4	...
	L5	N5	N6	P3	P5	...
	L6	N1	N2	P1	P0	...
	L7	N1	N6	P0	P3	...
	L8	N6	N7	P0	P5	...
	L9	N4	N7	P4	P0	...
	L10	N2	N4	P3	P0	...
<i>POLYGON:</i>	<i>id</i>	<i>line<sub>1</sub>, ..., line<sub>n</sub></i>				
	P1	L1, L6				
	P2	L2				
	P3	L1, L7, L5, L3, L10, L2				
	P4	L3, L4, L9				
	P5	L4, L5, L8				

18. ábra. A 17. ábrán látható tartománytérképet leíró adatstruktúra

*Megjegyzések:*

– Az lpoly és rpoly azonosítók a tartományterképek hatékony algoritmikus kezelését szolgálják. Pontos jelentésük: az adott határvonal az lpoly és rpoly tartományokat választja el, és pedig ha a node1 kezdőpontból haladunk a node2 záró pont felé, akkor lpoly bal oldalon, rpoly pedig jobb oldalon fekszik.

– Ha a poligon szigete(ke)t tartalmaz, akkor a sziget határvonalait is fel kell venni a POLYGON rekord listájára.

*Példa:* telekosztás. Összetett folyamat, amelyet a felhasználótól bekért adatok alapján egy programmodul (makró) oldhat meg.

**3.2.5. Hálózat**

0D és 1D típusú objektumok rendszere (pl. úthálózat, csőhálózat). Jellemző típusai:

**a) Geometriai hálózat:** a térbeli viszonyokat pontosan leképezi. Lehetséges felépítése:

NODE: csomópont tömb, egy rekordjának felépítése:

id : csomópont azonosító  
 x, y : koordináták  
 e<sub>1</sub>,...,e<sub>n</sub> : kiinduló élek azonosítói  
 [attribútumok, pl. van-e közlekedési lámpa]

LINE: vonallánc tömb, egy rekordjának felépítése:

id : él azonosító  
 node<sub>1</sub> : kezdő csomópont azonosítója  
 node<sub>2</sub> : záró csomópont azonosítója  
 x<sub>1</sub>, y<sub>1</sub>,...,x<sub>n</sub>, y<sub>n</sub> : töréspontok koordinátái  
 [attribútumok, pl. forgalom iránya]

A vonalláncok csomópont nélkül is keresztezhetik egymást (pl. felüljáró). A tartományterképnél ez nem volt megengedett.

**b) Logikai hálózat:** absztrakt gráf, térbeli információ nélkül. Lehetséges felépítése:

NODE: csomópont tömb, egy rekordjának felépítése:

id : csomópont azonosító  
 e<sub>1</sub>,...,e<sub>n</sub> : kiinduló élek azonosítói  
 [attribútumok]

EDGE: él tömb, egy rekordjának felépítése:

id : él azonosító  
 node<sub>1</sub> : kezdő csomópont azonosítója  
 node<sub>2</sub> : záró csomópont azonosítója  
 [attribútumok, pl. az él hossza]

**c) Félig geometriai hálózat:** a logikai és geometriai hálózat kombinációja. Lehetséges felépítése:

NODE: csomópont tömb, egy rekordjának felépítése:

id : csomópont azonosító  
 x, y : koordináták  
 e<sub>1</sub>,...,e<sub>n</sub> : kiinduló élek azonosítói  
 [attribútumok]

EDGE: él tömb, egy rekordjának felépítése:

id : él azonosító  
 node<sub>1</sub> : kezdő csomópont azonosítója  
 node<sub>2</sub> : záró csomópont azonosítója  
 [attribútumok, pl. az él tényleges hossza]

*Példa félig geometriai hálózatra:* vasúthálózat. A Szeged-Kiskunfélegyháza szakasz egyetlen él, annak ellenére, hogy a valóság hú megjelenítéshez töröttvonalal kellene leírni. Kérdés, hogyan tartjuk nyilván a közbülső állomásokat és egyéb pályamenti objektumokat? Erre megoldás a lineáris címzés módszere.

**Lineáris címzés módszere:** egy élen belül objektumok azonosítása. Például, ha egy útszakaszon híd van, ezt kétféleképp jelölhetjük az adatbázisban:

a). Az útszakasznak megfelelő élt 3 élre bontjuk, és a középsőhöz 'híd' attribútumot rendelünk. Ekkor az élek erősen elszaporodhatnak.

b). A hidakat és egyéb pályamenti létesítményeket külön táblában tartjuk nyilván:

Objektumok (obj\_id, edge\_id, dist1, dist2)

ahol *obj\_id* az objektum azonosítója, *dist1* a kezdetének, *dist2* a végének az *edge\_id* él kezdő-pontjától mért távolsága. Ekkor nem szaporodnak el az élek, és nem lassul a feldolgozás, ha a hidakra (és más, éleken lévő objektumokra) nem vagyunk kíváncsiak.

*Lekérdezések, elemzések hálózatokon:*

- Pl. két pont között mi az optimális útvonal adott súlyú és magasságú jármű számára.
- Városi tömegközlekedés optimalizálása (járatok útvonala, sűrűsége, átszállásszám minimalizálása, stb.)

### 3.2.6. Folytonos felület

Egy  $f(x, y)$  függvénnyel leírható folytonos felület (pl. domborzat, hőmérséklet) vektoros ábrázolására két mód nyílik:

a). Izovonalas ábrázolás. (Izovonal: töröttvonal, amelyhez attribútumként egy adott számérték tartozik.) Szokásos ábrázolási módjai:

LINE  $x_1, y_1, \dots, x_n, y_n, f$ : vonallánc attribútummal, vagy

LINE  $x_1, y_1, z_1, \dots, x_n, y_n, z_n$ : 3D-vonallánc (ez  $z_1 = \dots = z_n$  miatt redundáns, ugyanakkor 3D-modellezésnél előnyös lehet ez a megoldás).

b). TIN (= Triangulated Irregular Network): háromszögrács (részletesen lásd a Digitális terepmodellnél). Példaként két lehetséges tárolási módot mutatunk be:

(i) Háromszögenkénti tárolás: a TRIANGLE és NODE fájlkból áll.

A NODE fájl egy rekordjának felépítése:

– id: szögpont azonosító száma,

–  $x, y, z$ : szögpont koordinátái ( $z$  az  $f(x, y)$  függvény adott pontbeli értéke).

A TRIANGLE fájl egy rekord felépítése:

– id: a háromszög azonosító száma,

– node<sub>1</sub>, node<sub>2</sub>, node<sub>3</sub>: a három szögpont azonosító számai,

– tr<sub>1</sub>, tr<sub>2</sub>, tr<sub>3</sub>: a szomszédos háromszögek azonosító számai.



(ii) Szögpont-szomszédság szerinti tárolás: csak egy NODE fájlból áll, ahol egy rekord tartalma:

- id: szögpont azonosító száma,
- x, y, z: szögpont koordinátái,
- $node_1, \dots, node_n$ : szomszédos (éllel összekötött) szögpontok azonosító számai.

### 3.3. Térbeli indexelés

Gyakran van szükség adott térbeli feltételnek eleget tevő (például megjelenítésnél adott téglalapba eső, vagy snappingnél adott környezetbe eső) rajzelemek kiválasztására. Ilyenkor az összes rajzelem végigellenőrzése nyilván elfogadhatatlanul lassú lenne. A térbeli indexelés célja, hogy az ellenőrzendő rajzelemek számát nagyságrendekkel csökkentse. A kiválasztás általában két lépésből áll:

- *Előszűrés*: a szóba jöhető rajzelemek kiválasztása térbeli index segítségével.
- *Kiválasztás*: a feltételnek eleget tevő rajzelemek kiválasztása egyenkénti ellenőrzéssel.

Ha például összesen 100 000 rajzelem van, és ebből kell 20-at kiválasztani, akkor előszűréssel kiválasztunk – mondjuk – 100-at (ez gyors eljárás), majd ezekből kiválasztunk 20-at egyenkénti ellenőrzéssel (ez lassú, de csak 100 elemet kell vizsgálni, és nem százezret).

Indexelés nélkül még a térkép megjelenítése is elfogadhatatlanul lassú lehet, ugyanis minden kirajzoláskor az összes rajzelemet ellenőrizni kell, hogy bele esik-e a kirajzolandó ablakba. Emiatt – nagy állományok esetén – lassú lesz a kirajzolás akkor is, ha csak egy egészen kis kivágot szeretnénk kinagyítani a képernyőn. Mivel CAD rendszerek esetén nem jellemzők a nagy állományok, ezért indexelés sem feltétlenül szükséges. GIS rendszereknél azonban az indexelés elengedhetetlen.

Az indexelési módok tárgyalásánál feltesszük, hogy minden rajzelemnek egyedi azonosítója van.

#### 3.3.1. Négyzetrács index (grid index)

A teljes térkép területét  $n \times m$  négyzetből álló ráccsal fedjük le (a 19. ábrán  $n = m = 3$ ). Minden négyzethez egy indexlistát rendelünk, amely az adott négyzetbe – részben vagy egészben – beleeső rajzelemek indexeit tartalmazza (20. ábra). Ha egy rajzelem több négyzetben is szerepel, akkor szükségképpen több indexlistán fog szerepelni.

Gépi adatstruktúra szintjén egy kétsoros tömböt alkalmazhatunk, ahol az első sor a listaelemeket, a második sor pedig a láncoló "next" pointereket tartalmazza. A (20. ábrán a kilenc lista az 1., ..., 9. oszlopokban kezdődik. A listák végét  $-1$  jelzi.

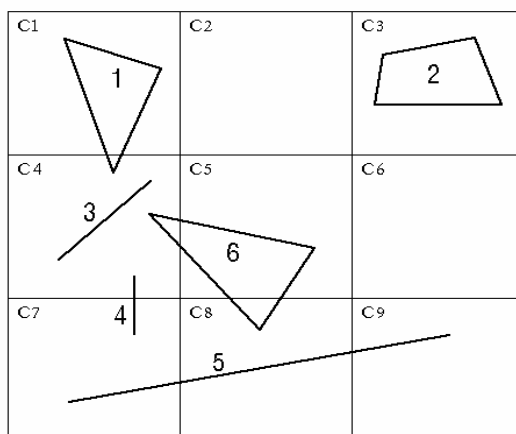
Kirajzoláskor ellenőrizni kell, hogy a képernyőn megjelenítendő kivágot mely négyzetekbe esik (előszűrés), és csak az ezeknek megfelelő indexlistákon kell végigmenni (kiválasztás). A kirajzolás annál gyorsabb lesz, minél kisebb kivágot akarunk megjeleníteni. Eleget nagy nagyítás esetén valós idejű scrollozás is lehetővé válik.

Az indexlisták redundánsak, vagyis egy rajzelem több listán is szerepelhet.

Megjegyzések:

– Amikor azt vizsgáljuk, hogy egy rajzelem beleesik-e egy grid négyzetbe, elegendő a rajzelem befoglaló téglalapját vizsgálni. Így ugyan előfordulhat, hogy egy rajzelem olyan négyzet listájára is felkerül, amelybe valójában nem esik bele (például a 19. ábrán szereplő 6. rajzelemet fel kellene venni a C7 négyzet listájára is). Ez azonban nem okoz gondot, mivel az indexelés csak előszűrést végez, az utána következő ellenőrzésnél az ilyen rajzelemek kiesnek.

– A rajz módosításakor nem szükséges a listákról törölni a törölt/módosított rajzelemekre való hivatkozásokat, elegendő csak az új/módosított rajzelemeket felvenni. Az így benmaradó hibás hivatkozások az előszűrés utáni ellenőrzésnél kiesnek.



19. ábra. Négyzetrács indexelés.

C1	C2	C3	C4	C5	C6	C7	C8	C9
R1		R2	R1	R6		R4	R5	R5
			R3			R5	R6	
			R4					
			R6					

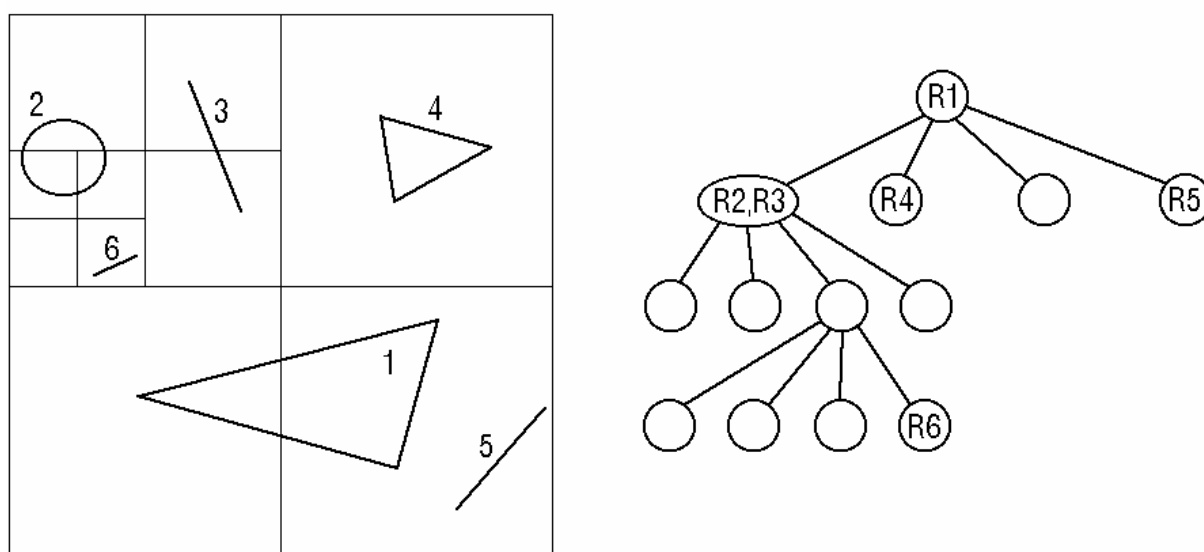
	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>	<b>10</b>	<b>11</b>	<b>12</b>	<b>13</b>	<b>14</b>
<b>R1</b>	-1	R2	R1	R6	-1	R4	R5	R5	R3	R4	R5	R6	R6	
	-1	-1	10	-1	-1	12	14	-1	11	13	-1	-1	-1	

20. ábra. Indexlisták a 19. ábra szerinti négyzetrács indexhez (fent) és a gépi adatstruktúra (lent). Az  $i$ -edik rajzelem azonosítóját  $R_i$  jelöli.

### 3.3.2. Négyesfa index

Alapelv: a teljes rajzterületet alkotó téglalapot négy egyenlő részre osztjuk, majd az egyes negyedeket tovább negyedeljük, stb. Így egy fastruktúra keletkezik, amelynek gyökere a teljes rajzterületet reprezentálja, szögpontjai pedig a negyedeléssel kapott egyes szegmenseket.

A négyesfa sokféleképpen felépíthető, pl. attól függően, hogy pontszerű vagy területi objektumok tárolására kívánjuk használni (lásd az irodalomjegyzékben Samet, 1989.) Itt csak egy jellemző megoldást mutatunk be. Minden rajzelemet a négyesfa egy (és csak egy) szögpontjához rendeljük: ahhoz a szögponthoz, amelyhez tartozó szegmensbe a rajzelem befoglaló téglalapja teljes egészében belefér, de annak egyik negyedébe sem fér már bele.



21. ábra. Négyesfa index

Adatstruktúra:

A négyesfa  $\text{NODE}(n1, n2, n3, n4, \text{rajzelemlista})$  felépítésű elemek sorozata lehet, ahol

- $n1, \dots, n4$ : a leszármazott node-okra mutató pointerek,
- *rajzelemlista*: az adott szögponthoz tartozó rajzelemek azonosítói.

Négyesfára rajzelem felfűzése: először a gyökérre helyezzük. Ha valamelyik negyedben elfér, akkor egy szinttel süllyesztjük, stb.

Ablak kirajzolásnál gyökérből indulunk, de csak azokra a leszármazottakra megyünk tovább, amelyek metszik az ablakot.

Megjegyzés: a négyesfa index itt leírt változata redundanciamentes, vagyis minden rajzelem csak egyszer szerepel rajta.

### 3.4. Vektoros algoritmusok

#### 3.4.1. Egyenesszakaszok metszéspontja

Az  $A = (a_x, a_y)$  és  $B = (b_x, b_y)$  pontokat összekötő, valamint a  $C = (c_x, c_y)$  és  $D = (d_x, d_y)$  pontokat összekötő egyenesszakaszok  $Z = (x, y)$  metszéspontját számítjuk. Először az A és B, valamint a C és D pontokon áthaladó egyenesek metszéspontját határozzuk meg. A megoldandó egyenletrendszer:

$$(b_y - a_y) * x - (b_x - a_x) * y = a_x * b_y - a_y * b_x$$

$$(d_y - c_y) * x - (d_x - c_x) * y = c_x * d_y - c_y * d_x$$

Szakaszok metszéspontjának meghatározásakor  $Z = (x, y)$ -t csak akkor fogadjuk el, ha A és B közé, ill. C és D közé esik:

$$(x - a_x) * (x - b_x) \leq 0$$

$$(y - a_y) * (y - b_y) \leq 0$$

$$(x - c_x) * (x - d_x) \leq 0$$

$$(y - c_y) * (y - d_y) \leq 0$$

*Ha sok egyenesszakasz metszését* kell elvégezni (például ha spagetti adatstruktúrán topológiát építünk), akkor  $n$  egyenesszakasz esetén  $n * (n-1) / 2$  metszészvizsgálatot kellene végezni. Ugyanakkor a vizsgált egyenesszakaszok gyakran távol vannak egymástól, így eleve nem lehet metszés. Ezért célszerű a számítást úgy gyorsítani, hogy először a két egyenesszakasz befoglaló téglalapját határozzuk meg: ha a befoglaló téglalapok diszjunktak, akkor az egyenesszakaszok sem metszhetik egymást. Az AB szakasz P befoglaló téglalapját a bal alsó sarok  $(p_{x1}, p_{y1})$ , és a jobb felső sarok  $(p_{x2}, p_{y2})$  koordinátaival adjuk meg, hasonlóan a CD szakasz Q befoglaló téglalapjára:

$$p_{x1} = \min(a_x, b_x) \quad q_{x1} = \min(c_x, d_x)$$

$$p_{y1} = \min(a_y, b_y) \quad q_{y1} = \min(c_y, d_y)$$

$$p_{x2} = \max(a_x, b_x) \quad q_{x2} = \max(c_x, d_x)$$

$$p_{y2} = \max(a_y, b_y) \quad q_{y2} = \max(c_y, d_y)$$

A befoglaló téglalapok akkor és csak akkor diszjunktak, ha

$$p_{x2} < q_{x1} \text{ vagy } q_{x2} < p_{x1} \text{ vagy } p_{y2} < q_{y1} \text{ vagy } q_{y2} < p_{y1}$$

Térbeli indexeléssel a számítás tovább gyorsítható, mivel ekkor a befoglaló téglalapok vizsgálatát is csak a közeli egyenesszakaszokra kell elvégezni.

### 3.4.2. Vonalláncok metszése

Ha az L és M vonalláncok n ill. m egyenes szakaszból állnak, akkor  $n \cdot m$  metszésvizsgálatot kellene végezni. A számításigény csökkenthető befoglaló téglalapok vizsgálatával:

– először a teljes vonalak befoglaló téglalapjait határozzuk meg. Az  $L = (x_1, y_1, \dots, x_n, y_n)$  vonal befoglaló téglalapja a  $(p_{x1}, p_{y1}, p_{x2}, p_{y2})$  négyessel adható meg, ahol

$$p_{x1} = \min(x_1, \dots, x_n)$$

$$p_{y1} = \min(y_1, \dots, y_n)$$

$$p_{x2} = \max(x_1, \dots, x_n)$$

$$p_{y2} = \max(y_1, \dots, y_n)$$

hasonlóan adódik a  $(q_{x1}, q_{x2}, q_{y1}, q_{y2})$  befoglaló téglalap.

– Megvizsgáljuk, hogy a befoglaló téglalapok metszik-e egymást. Ha ugyanis a téglalapok diszjunktak, akkor a vonalak sem metszhetik egymást. A téglalapok akkor és csak akkor diszjunktak, ha

$$p_{x2} < q_{x1} \text{ vagy } q_{x2} < p_{x1} \text{ vagy } p_{y2} < q_{y1} \text{ vagy } q_{y2} < p_{y1}$$

– A számításigény tovább csökkenthető, ha a vonalakat monoton szakaszokra osztjuk. Egy  $x_i, y_i, \dots, x_j, y_j$  vonalszakaszt monotonnak nevezünk, ha mind az x, mind az y koordináták monoton nőnek vagy csökkennek. Ha pl. egy monoton növe és monoton csökkenő szakasz metszését vizsgáljuk, ezek legfeljebb egyszer metszhetik egymást, a metszéspont megtalálása után tehát további metszésvizsgálat fölösleges.

### 3.4.3. Poligonok területe

Tekintsük az  $x_1, y_1, \dots, x_n, y_n$  pontok által meghatározott zárt poligont ( $x_{n+1} = x_1, y_{n+1} = y_1$ ). Feltételezzük, hogy a poligon önmagát nem metszi.

Bocsássunk függőleges egyenest minden szögpontról az x tengelyre, és számítsuk ki minden élhez az él, a függőleges egyenesek és az x tengely által bezárt trapéz (előjeles) területét (22. ábra):

$$T_i = (x_{i+1} - x_i) * (y_{i+1} + y_i) / 2$$

A poligon (előjeles) területe az egyes trapézok területének összegeként adódik:

$$T = \sum_{i=1}^n T_i$$

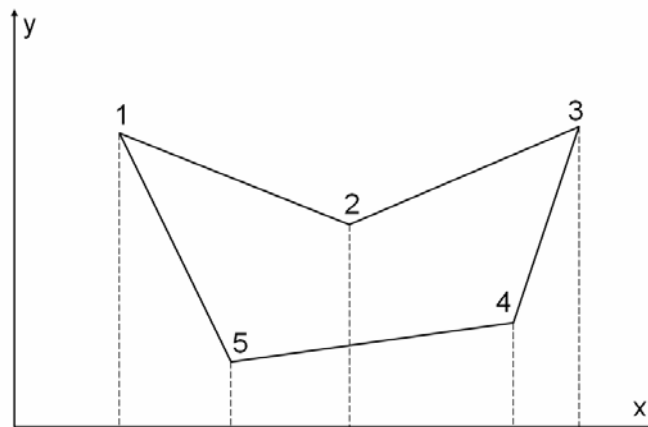
A fenti formula tetszőleges konvex vagy konkáv poligon esetén helyes eredményt ad, feltéve, hogy minden y érték pozitív. Amennyiben negatív y értékek is fellépnek, az y koordinátákra  $y_i' = y_i - \min(y_1, \dots, y_n)$  eltolást kell alkalmazni.

Ha a poligont óramutató járása szerint járjuk körül, akkor pozitív, egyébként negatív terület adódik. A tényleges területérték tehát  $|T|$ .

Ha sok poligon területét kell egyidejűleg meghatározni, akkor tartományterkép adatstruktúra esetén a következő megoldás célszerű:

- Minden vonalhoz egy (előjeles) területértéket számolunk a fenti trapéz módszerrel.
- Az egyes poligonok területét a határoló vonalak területének összegeként kapjuk, ügyelve arra, hogy ha az adott poligon egy vonalnak bal poligonja, akkor a vonalhoz rendelt terület  $(-1)$ -szeresével számolunk.

Ez az eljárás helyes eredményt ad lyukak (szigetek) esetén is, sőt akkor is, ha a poligonok vonallistája rendezetlen!



22. ábra: Poligon területének számítása.

#### 3.4.4. Pont-poligon algoritmus

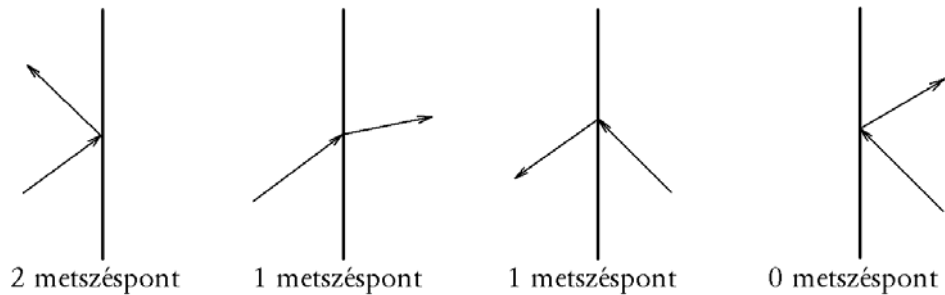
Feladat: el kell dönteni, hogy egy adott  $(u, v)$  pont egy  $x_1, y_1, \dots, x_n, y_n$  poligon belsejében van-e ( $x_{n+1} = x_1, y_{n+1} = y_1$ ). A poligon szigetet is tartalmazhat.

*Megoldás elve:* rajzoljunk a pontból függőlegesen felfelé egy félegyenest! Ha ez páratlan számú helyen metszi a poligont, akkor a pont belül van, egyébként kívül van. Ez akkor is igaz, ha a poligon sziget(ek)et tartalmaz.

*Problematikus esetek kezelése:*

(i) Ha a poligon egy éle függőleges, akkor nem számítunk metszéspontot (akkor sem, ha éppen fedésben van a félegyenessel).

(ii) Ha a félegyenes a poligon egy szögpontján halad át, akkor csak a balról érkező ill. balra távozó élt tekintjük metszéspontnak (23. ábra).



23. ábra

*Algoritmus:* Ha az alábbi algoritmus lefutása után a 'val' változó értéke 1, akkor a pont belül, ha -1, akkor kívül van, ha 0, akkor a határon van:

```

val := -1
for i=1 to n
if xi+1 != xi then                                // nem függőleges él, lásd (i) eset
  if (xi+1-u)*(u-xi) >= 0 then                    // u ∈ [xi, xi+1]
    if xi+1 != u or xi <= u then                // nem jobbról jön, lásd (ii) eset
      if xi != u or xi+1 <= u then            // nem jobbra megy, lásd (ii) eset
        b := (yi+1-yi)/(xi+1-xi)          // iránytangens
        metsz := yi + b*(u-xi)
        if metsz > v then val := val*(-1)
        if metsz = v then val := 0 // határon van
      end if
    end if
  end if
end if
end if
next i

```

### 3.4.5. Sok pont, sok poligon vizsgálata

Tartománytérkép esetén nyilvánvaló, hogy egy adott pont egy és csak egy poligonba eshet bele. Tegyük fel, hogy adott  $Z_1, \dots, Z_n$  pontokról el kell dönteni, hogy melyik pont melyik poligonba esik. Az algoritmust a korábban tárgyalt tartománytérkép adatstruktúrán mutatjuk be.

Itt is minden pontból függőlegesen felfelé félegyenest indítunk, és minden egyes vonallal megvizsgáljuk a metszéspontját (metszéspontjait). A  $Z_1, \dots, Z_n$  pontokhoz az  $m_1, \dots, m_n$  és  $p_1, \dots, p_n$  segéd tömböket vesszük fel, ahol  $m_i$  a legelső metszéspont  $y$  koordinátáját,  $p_i$  pedig a megfelelő poligon azonosítót fogja tartalmazni. Kezdetben minden  $p_i$  definiálatlan,  $m_i$  értéke „végtelen”.

Az algoritmus az  $L_1, \dots, L_m$  vonalakon megy végig:

```

For L = L1, ..., Lm
  xmin := L minimális x-koordinátája
  xmax := L maximális x-koordinátája
  ymin := L minimális y-koordinátája
  ymax := L maximális y-koordinátája
  for i=1 to n      // végig a pontokon, Zi koordinátái (xi, yi)
    if (xmin ≤ xi ≤ xmax) then
      if (yi < ymax and ymin < mi)
        metsz := Zi-ből induló függőleges félegyenes L-lel való
                legalsó metszéspontjának y-koordinátája
        if pi = -1 or mi > metsz then
          mi := metsz
          pi := alsó_poligon
        end if
      end if
    end if
  next i
next L

```

ahol "alsó\_poligon" a metszéspont alatti poligon azonosítója, amelyet a vonal lpoly, rpoly paramétereiből tudunk meghatározni.

Amikor az összes vonallal végeztünk, a Z<sub>i</sub> ponthoz a p<sub>i</sub> tömbelem tartalmazza a legalsó metszéspont alatti poligon azonosítóját, vagyis azét, amely a pontot tartalmazza. A pontok x koordináta szerinti rendezésével az algoritmus tovább gyorsítható.

### 3.4.6. Poligon overlay algoritmus

*Példa.* Tekintsünk két tartományterképet: egy *talajterképet*, amelyen az egyes tartományok adott talajminőségű területeket kódolnak, és egy *kataszteri térképet*, amely földrészleteket tartalmaz. Meg kell határozni az egyes földrészletek értékét, amelyet a rá eső különböző talajminőségű területek súlyozott összegével számíthatunk. Ehhez az alábbi feladat megoldása szükséges, amelyet poligon overlay (poligon fedvényezés) műveletnek neveznek.

*Feladat.* Adott két, diszjunkt tartományokkal lefedett fedvény:

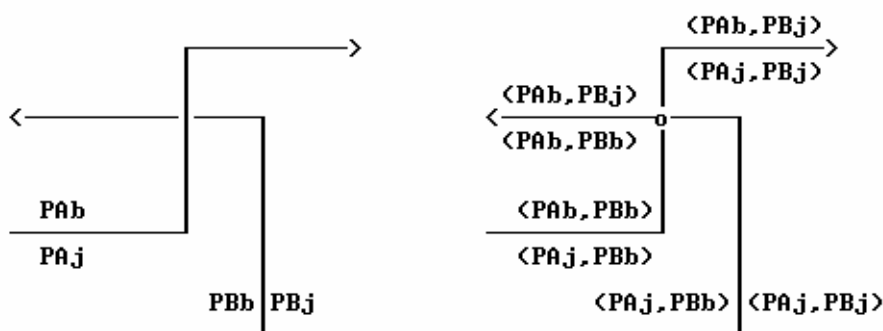
$$\begin{aligned}
 A &= A_1 \cup \dots \cup A_n \\
 B &= B_1 \cup \dots \cup B_m
 \end{aligned}$$

ahol az egyes tartományokhoz rendre az  $a_1, \dots, a_n$  ill.  $b_1, \dots, b_m$  attribútumok tartoznak. A metszet fedvényt kell képezni, vagyis olyan  $C = C_1 \cup \dots \cup C_z$  fedvényt, ahol a lefedő tartományok szintén diszjunktak, és  $C_k$  akkor és csak akkor szerepel C-ben, ha van olyan  $A_i$  és  $B_j$ , hogy a metszetük éppen  $C_k$ -val egyenlő. Ekkor  $C_k$ -hoz az  $(a_i, b_j)$  attribútumpárt kell rendelni.

*Algoritmus:* Az input és output fedvényeknél egyaránt a tartományterkép vonal-poligon adatstruktúráját tételezzük fel. Tehát adott az A fedvényt leíró  $A_{\text{node}}$ ,  $A_{\text{line}}$ ,  $A_{\text{poly}}$  és a B fedvényt leíró  $B_{\text{node}}$ ,  $B_{\text{line}}$ ,  $B_{\text{poly}}$  adatstruktúrák, ezekből kell létrehozni a megfelelő  $C_{\text{node}}$ ,  $C_{\text{line}}$ ,  $C_{\text{poly}}$  adatstruktúrákat.

Ha az  $A_i$  és  $B_j$  poligonokat az  $i, j$  sorszámokkal azonosítjuk, akkor az  $A_i$  és  $B_j$  metszésével előálló C-beli poligon azonosítója legyen az  $(i, j)$  pár. (A node és line elemek azonosítóinak megadására nincs megkötés.)





24. ábra.  $P_{Ab}$  és  $P_{Aj}$  A-beli bal- és jobboldali poligonazonosítók. Hasonlóan,  $P_{Bx}$  B-beli, ( $P_{Ax}$ ,  $P_{By}$ ) pedig C-beli poligonazonosító

Az algoritmus lépései:

1.  $C_{node} := A_{node} \cup B_{node}$ .
2.  $C_{line} := A_{line} \cup B_{line}$ . (Itt az  $lpoly$  és  $rpoly$  értéke még az A és B fedvényekre vonatkozik.)

3. Képezzük valamennyi A-beli vonal metszéspontjait valamennyi B-beli vonallal. (A vonalak koordináta szerinti rendezésével, és befoglaló téglalapok vizsgálatával elérhető, hogy nem kell minden A-beli vonalat minden B-belivel összevetni.) Minden egyes metszésnél az alábbiakat kell elvégezni:

- A  $C_{node}$  tömbbe felvenni a metszéspontot.
- A  $C_{line}$  megfelelő vonal elemeit helyettesíteni kell a megfelelő részvonallakkal.
- Minden újonnan keletkezett részvonálhoz meghatározzuk a bal és jobboldali poligon azonosítókat, a metszésben részt vett az A,B-beli vonalak bal/jobbs poligon azonosítóinak felhasználásával (lásd ábra).

4.  $C_{poly}$  létrehozása:  $C_{line}$ -on végighaladunk, és minden egyes vonalnál  $lpoly$  és  $rpoly$  értékét vizsgáljuk:

- Ha a megfelelő poligonok még nem szerepeltek  $C_{poly}$ -ban, akkor felvesszük azokat.
- $C_{poly}$ -ban a megfelelő poligon vonal-listájára felvesszük az adott vonalra való hivatkozást.

A fenti eljárás eredményeként fokozatosan kitöltődik a  $C_{poly}$  tömb.

Hangsúlyozzuk, hogy a fentiekben csak az algoritmus alapötleteit vázoltuk, annak tényleges megvalósításához még számos speciális esetet le kell kezelni.

*Néhány speciális eset:*

- Egy  $A_i$  és egy  $B_j$  poligon metszete több poligonból álló poligonhalmaz is lehet.
- Ha egy vonalszakasz az A és B fedvényen egyaránt szerepel (pl. folyó partvonala egy talaj és egy növénytakaró térképen), ez speciális kezelést igényel. *További gyakorlati probléma:* ha a digitalizálási pontatlanság miatt a közös vonalak egymástól többé-kevésbé eltérnek, akkor számos új metszéspont és tartomány keletkezhet a poligon overlay algoritmusnál. *Megoldás:* bizonyos tolerancia küszöbnél közelebb álló vonalszakaszokat a rendszer azonosnak tekinthet.

### 3.5. Adatbázismodellek: térbeli és leíró adatok összekapcsolása

A térinformatikai rendszerek lényege, hogy a grafikus (térképi) és nem grafikus (leíró) adatokat együtt, integráltan tudják kezelni. Ennek eredményeként például az alábbi lehetőségek birtokába jutunk:

(i) *Rajz* → *adatbázis lekérdezés*: a rajzon grafikus eszközökkel kijelölünk egy rajzelemet vagy rajzelemek egy csoportját, eredményül a kapcsolt leíró adatok listáját kapjuk. (Például adott poligonba eső telkek tulajdonosai.)

(ii) *Adatbázis* → *rajz lekérdezés*: hagyományos adatbázis-lekérdezéssel kijelölünk egy rekordcsoportot, eredményül a rajzon a kapcsolt rajzelemek eltérő színnel jelennek meg. (Például a 80 évnél idősebb tulajdonosok telkei.)

(iii) *Rajz feliratozása az adatbázisból*. (Például a térképen megjelenő helyrajzi számok az adatbázisból kerülnek frissítésre.)

A térképi és leíró adatok együttes kezelésének főbb módjait tekintjük át a következőkben.

#### 3.5.1. Szétválasztott modell

A reprezentálandó adatokat két csoportra bontjuk:

- *grafikus adatok*: térképen ábrázolandók,
- *leíró adatok*: táblázatokban tárolhatók.

A szoftverek fejlődése során külön rendszerek jöttek létre a rajzi és táblázatos adatok kezelésére, így általában két független szoftver között kell kapcsolatot teremteni:

– *Grafikus rendszer (Graphics System = GS)*: a vektoros rajzot kezeli, például CAD rendszer.

– *Adatbázis-kezelő rendszer (Database Management System = DBMS)*: a szöveges (táblázatos) adatokat kezeli.

A kapcsolatteremtésről rendszerint a GS gondoskodik, a DBMS a GS-től függetlenül működik.

A leíró adatok egyed-kapcsolat diagrammal modellezhetjük és (relációs) adatbázisként kezelhetjük.

A grafikus adatokat rétegekbe és rétegcsoporthoz szokták rendezni.

– *réteg*: hasonló jellegű, jelentésű rajzelemek együttese, amelyek általában azonos színnel jelennek meg a térképen.

– *rétegcsoporthoz*: több, tartalmilag összetartozó réteg együttese.

Gyakran többszintű hierarchiát alkalmaznak, például réteg, altéma, téma, teljes térkép.

#### Példa:

<i>Rétegcsoporthoz</i>	<i>Réteg</i>
– alaptérkép:	– épületek
	– telkek
	...

- vízmű:                   – nyomócső hálózat
- szennyvíz hálózat
- ...
- elektromos:           – magasfeszültségű hálózat
- transzformátor állomások
- ...

A szétválasztott modellnél *fedvénynek* nevezzük a tematikailag összetartozó rétegek és a hozzájuk kapcsolódó adattáblák együttesét.

A grafikus és leíró adatok közti kapcsolat rajzelem-adatrekord összekapcsolással valósul meg (általában egy rajzelemhez egy adatrekord kapcsolódik). *Erős kapcsolásról* beszélünk, ha a kapcsolt rajzelem törlése a megfelelő adatrekord törlését is maga után vonja, *gyenge kapcsolás* esetén ez nem történik meg.

Az alábbiakban két jellemző kapcsolási módot mutatunk be.

### Rajz és adatbázis közötti kapcsolat

*1. módszer:* vektoros rajzelemek bővítése adatbázis link-ekkel. (A *MicroStation* rendszer alkalmazza.) A kapcsolat elve (25. ábra):

– A DBMS oldaláról: minden kapcsolandó táblát bővíteni kell egy LINK nevű mezővel, amely kezdetben egy folyamatos rekord sorszámot tartalmazhat (25. ábra).

– A GS oldaláról: minden rajzelemhez tetszőleges számú adatbázis link csatolható, egy link a kapcsolt tábla nevét és a rekord sorszámát tartalmazza, egy további bit pedig megmondja, hogy erős vagy gyenge kapcsolásról van szó.

<i>Rajz file (GS)</i>	<i>Adattáblák (DBMS)</i>	
rajzelem	R (A <sub>1</sub> , ..., A <sub>n</sub> , LINK)	S (B <sub>1</sub> , ..., B <sub>m</sub> , LINK)
link=(R, 3)	a <sub>11</sub> ... a <sub>1n</sub> 1	b <sub>11</sub> ... b <sub>1m</sub> 1
rajzelem	a <sub>21</sub> ... a <sub>2n</sub> 2	b <sub>21</sub> ... b <sub>2m</sub> 2
rajzelem	a <sub>31</sub> ... a <sub>3n</sub> 3	b <sub>31</sub> ... b <sub>3m</sub> 3
link=(R, 1)	...	...
link=(S, 2)		
rajzelem		
link=(S, 2)		
...		

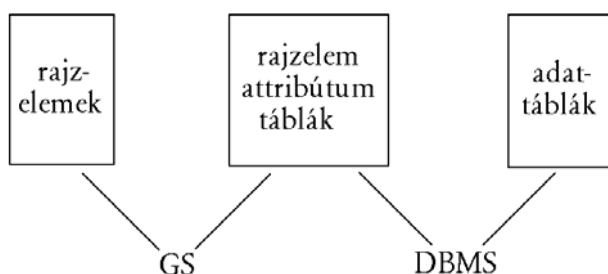
25. ábra: adatbázis kapcsolat link-eken keresztül.

*2. módszer:* összekapcsolás rajzelem-azonosítók segítségével. (Az *ArcInfo* rendszer alkalmazza.) A kapcsolat elve:

– A GS oldaláról: minden rajzelemnek egyedi azonosítója van. A GS minden rajzelem típushoz egy speciális adattáblát, úgynevezett *rajzelem-attribútumtáblát* generál, amelynek minden egyes rekordja egy rajzelem leírását tartalmazza.

– A DBMS oldaláról: az adatbázis táblái a rajzelem-attribútumtáblákhoz a szokásos relációs módon kapcsolódnak (külső kulcsokkal).

Ennél a kapcsolatnál tehát a GS kezeli a rajz file-t és a rajzelem-attribútumtáblákat, a DBMS pedig a rajzelem-attribútumtáblákat és a további táblákat (26. ábra).



26. ábra: adatbázis kapcsolat rajzelem-attribútumtáblákon keresztül.

*Példa:* Tegyük fel, hogy egy vállalati leltárt nyilvántartó adatbázist ki akarunk egészíteni a vállalat irodaépületének tervrajzával (27. ábra), és meg akarjuk jelölni, hogy melyik tárgy melyik helyiségben található.

iroda	iroda	mosdó	iroda	iroda
folyosó				
tanácsterem	iroda			raktár

27. ábra

Az egyes termeket határoló, poligon típusú rajzelemekhez a GS egy *POLYGON* (*id*, *teremid*, *ter*, *ker*) rajzelem-attribútum-táblát hoz létre, ahol:

- *id*: poligon azonosítója,
- *teremid*: a poligon felhasználói azonosítója
- *ter*: poligon területe,
- *ker*: poligon kerülete.

Az adatbázisban a berendezési tárgyakat egy *LELTÁR* (*lszám*, *megnev*, *érték*, *dátum*, *teremid*) táblában tartjuk nyilván, melynek mezői:

- *lszám*: a tárgy leltári száma,
- *megnev*: a tárgy megnevezése,
- *érték*: a tárgy beszerzési értéke,
- *dátum*: a beszerzés dátuma,
- *teremid*: a terem azonosítója, amelyben található.

A *POLYGON* és *LELTÁR* táblák között a *teremid* mezők képeznek kapcsolatot.

### **Példa szétválasztott modellre**

Egyszerűsített ingatlan-nyilvántartást veszünk alapul, amely földrészleteket (telkeket), épületeket és tulajdonosaikat kezeli.

*Leíró adatok (E-K modell, relációs modell):*

Egyed típusok:

- TULAJDONOS (tulazon, név, lakcím)
- TELEK (helyrajzsorszám, terület)
- ÉPÜLET (postacím, alapterület, szintszám)

Kapcsolatok:

- telek-tulajdonos: TELEKTUL (helyrajzsorszám, tulazon, hányad)
- épület-tulajdonos: ÉPÜLETTUL (postacím, tulazon, hányad)

*Grafikus adatok:*

- *Telek réteg:* földrészlet-poligonok, telek helyrajzi számok.
- *Épület réteg:* épületpoligonok, házszámok.
- *Vízrajz réteg:* tavak, vízfolyások
- *Domborzat réteg:* szintvonalak

*Rajz-adatbázis kapcsolat:*

- a telek poligonok és a telek rekordok között,
- az épület poligonok és az épület rekordok között.

*Fedvények:*

- TELEK fedvény: a *Telek* réteg és a TELEK adattábla együttese.
- ÉPÜLET fedvény: az *Épület* réteg és az ÉPÜLET adattábla együttese.
- VÍZRAJZ fedvény.
- DOMBORZAT fedvény.

### **A szétválasztott modell értékelése**

A modell alapvető jellemzője, hogy a térbeli és leíró adatokat elkülönítve kezeli. Ebből adódnak az előnyei és hátrányai is.

*Előnyök:*

- Nagy múltú, fejlett grafikus (GS) és adatbázis-kezelő (DBMS) szoftverek összekapcsolását teszi lehetővé.
- Gyakran a grafikus és leíró adatok külön keletkeznek (rendszerint a leíró adatok már régen adatbázisban vannak, amikor a digitális térképek elkészülnek), így ezek utólagos összekapcsolása természetes megoldás lehet.
- Bizonyos, a témához szorosan nem kapcsolódó rajzi információk könnyebben kezelhetők réteg szemlélettel (pl. ingatlan-nyilvántartásnál vízrajz, domborzat).

*Hátrányok:*

- A térbeli adatok esetén elveszítjük az adatbázis-funkcionalitást (biztonsági mechanizmusok, naplózás, adat-rekonstrukció, stb.)
- A grafikus és leíró adatokat elkülönült kezelése miatt a két adatbázis elszakadhat egymástól.

Ez utóbbi elkerülésére

- erős kapcsolat alkalmazható (ahol indokolt),
- mentéskor a grafikus és leíró adatfájlokat mindig együtt kell menteni,
- a grafikus rendszernek időnként ellenőriznie kell a kapcsolatokat (például az eltérésekről hibalistát adhat).

### 3.5.2. Integrált modell: tisztán relációs megközelítés

A szétválasztott modell hátrányai eltűnnek, ha a grafikus és leíró adatokat egyaránt relációs adatbázisban tároljuk. Ebben az esetben a korábbi példában szereplő *Telek* fedvény a következő relációsémákkal írható fel:

TELEK (helyrajzszám, terület, poligonid)

POLIGON (poligonid, sorszám, pontid)

PONT (pontid, x, y)

*A TELEK adattábla:*

<i>Helyrajzszám</i>	<i>Terület</i>	<i>Poligonid</i>
1121	250	47
3655	400	48
2276	1300	33

*A POLIGON adattábla:*

<i>Poligonid</i>	<i>Sorszám</i>	<i>Pontid</i>
47	1	11
47	2	12
47	3	13
48	1	12
48	2	13
48	3	14
48	4	15
33	1	11
33	2	16
33	3	17
33	4	18

*A PONT adattábla:*

<i>Pontid</i>	<i>X</i>	<i>Y</i>
11	220	110
12	310	115
13	307	250
14	442	250
15	435	105
16	156	110
17	150	244
18	220	238

28. ábra. A *Telek* fedvény tisztán relációs megvalósítása

A megfelelő adattáblák a 28. ábrán láthatók. A *sorszám* attribútum határozza meg a poligonok szögpontjainak sorrendjét. A 3655 helyrajzi számú telek határvonalának koordinátáit az alábbi módon kérdezzük le:

```
SELECT sorszám, x, y FROM Telek, Poligon, Pont
WHERE helyrajziszám = 3655 AND Telek.poligonid = Poligon.poligonid
AND Poligon.pontid = Pont.pontid
ORDER BY sorszám;
```

Ennél a modellnél *fedvényen* térbeli adatokat tartalmazó adattáblák együttesét értjük.

Az integrált megközelítés *előnye*, hogy a grafikus és leíró adatokat egy közös adatbázisban tároljuk, így azok nem szakadhatnak el egymástól, és a teljes adatbázis-funkcionalitás érvényesül valamennyi adatra. *Hátrány* viszont, hogy a térbeli adatok kezelése nehézkessé válik: lekérdezéskor például ismernünk kell a térbeli adatok pontos tárolási struktúráját. Az integrált modell tehát javításra szorul.

### 3.5.3. Integrált modell: objektum-relációs megközelítés

Az objektum-relációs megközelítést az SQL3 szabvány rögzíti, és számos DBMS (például Oracle, PostgreSQL) alkalmazza. Lényege: a relációs modellt *absztrakt adattípusokkal* (ADT = *Abstract Data Type*) egészítjük ki. Az ADT-khez a felhasználó metódusokat definiálhat, és ha az ADT-t ezeken keresztül kezeli, függetlenítheti magát az ADT belső felépítésétől.

*Fedvénynek* nevezhetünk minden olyan relációsémát (esetleg relációsémák együttesét), amely térbeli ADT-t tartalmaz.

Térbeli ADT definiálásához az objektum-relációs modell számos eszközt biztosít, ezek közül itt csak a *beágyazott táblát* emeljük ki: ennek lényege, hogy egy relációséma egy attribútuma maga is relációséma lehet.

*Példa:* A *Telek* relációséma a következőképp definiálható:

Telek (helyrajziszám, terület, Polygon(sorszám, x, y))

ahol a *Polygon* beágyazott tábla, amely sorrendben egy poligon szögpontjainak koordinátáit tartalmazza. Minden egyes telekhez külön kis poligon-tábla tartozik. Továbbá, definiálhatunk egy TelekTerület (helyrajziszám) metódust, amely a helyrajzi számmal megadott telek területét adja vissza.

A objektum-relációs modell *előnyei*:

- Az ADT-k lehetővé teszik a térbeli adatok intelligens kezelését.
- A térbeli és leíró adatok egységes rendszerben kezelhetők, a teljes adatbázis-funkcionalitás rendelkezésre áll.

*Hátrányt* jelent viszont, hogy a fejlesztőnek magának kell definiálni a térbeli ADT-eket és metódusokat, és ezt a különböző fejlesztők még azonos DBMS esetén is valószínűleg eltérő módon teszik meg. Felmerül tehát az igény beépített térbeli adattípusok létrehozására.

### 3.5.4. Integrált modell: térbeli adattípusok

Egyre több DBMS támogat térbeli adattípusokat (MySQL, PostgreSQL, Oracle Spatial). Az egyes rendszerek adattípusai és kezelésük eltérő, ennek részleteit nem tárgyaljuk. Itt példaként az alábbi adattípusokat tételezzük fel:

- *Point*, azaz koordinátaival adott pont.
- *LineString*, azaz vonallánc.
- *Polygon*, azaz zárt vonallánc által határolt terület.
- *MultiPolygon*, amely poligonok halmazát jelenti.

Néhány művelet, amelyek a fenti adattípusokra definiálhatók:

- *Area (polygon)*: a poligon területét adja.
- *PointInPolygon (point, polygon)*: értéke *igaz*, ha a pont a poligon belsejében van, egyébként *hamis*.
- *Intersection (polygon, polygon)*: értéke *multipolygon*, amely a két poligon metszetét adja. (Ugyanis ha egyik vagy mindkét poligon konkáv, akkor a metszet különálló poligonok halmaza is lehet.) A visszaadott érték NULL, ha a poligonoknak nincs közös része.

*Példa:* A *Telek* relációséma a következőképp definiálható:

TELEK (helyrajzszám, terület, geo)

ahol a *geo* attribútum típusa *polygon*. Mindez SQL-ben:

```
CREATE TABLE Telek
( helyrajzszám CHAR(10) PRIMARY KEY,
  terület INTEGER,
  geo POLYGON );
```

Az alábbi lekérdezés azon telkek helyrajzi számát listázza, amelyeknél hibás területérték szerepel az adatbázisban:

```
SELECT helyrajzszám FROM Telek
WHERE Area(geo) <> terület;
```

Az alábbi lekérdezés azon telkek térbeli adatait válogatja le (például megjelenítéshez), amelyek területe kisebb egy adott értéknél:

```
SELECT geo FROM Telek
WHERE terület < 200;
```

*Példa.* Tegyük fel, hogy a *Kút* (id, típus, geo) relációsémában kutakat tartunk nyilván, a *geo* attribútum típusa *point*. Az alábbi lekérdezés térbeli összekapcsolást (spatial join) hajt végre: azon telkek helyrajzi számát listázza, amelyeken nyilvántartott kút van:

```
SELECT helyrajzszám FROM Telek, Kút
WHERE PointInPolygon (Kút.geo, Telek.geo);
```

*Példa.* Tekintsük az alábbi talajtérkép-fedvényt:

Talaj (talajid, talajnév, geo)



ahol *geo* típusa *multipolygon*. Az alábbi lekérdezés lényegében egy poligon-overlay műveletet hajt végre a *Telek* és *Talaj* fedvények között, a lekérdezés eredménye egy tartománytérkép fedvény lesz:

```
SELECT helyrajziszám, talajid, Intersection (Telek.geo, Talaj.geo)
FROM Telek, Talaj
WHERE Intersection (Telek.geo, Talaj.geo) IS NOT NULL;
```

### 3.5.5. Adatbázismodellek összefoglalása

*Szétválasztott modell:* a meglévő alkalmazások jelentős része ilyen (MicroStation, ArcInfo).

*Integrált megoldások:*

- Tisztán relációs modell: elvi jelentőségű, gyakorlatilag nem használják.
- Objektum-relációs modell: elvétve alkalmazzák.
- Térbeli adattípusok: terjedőben (Oracle Spatial, PostgreSQL, PostGIS, MySQL).

*A GIS alkalmazásfejlesztők általában az alábbi stratégiát követik:*

– Az alkalmazás tervezésekor rendszerint objektum-orientált szemlélettel dolgoznak (például *Telek* objektum, UML.)

– A kivitelezési szakaszban – az adott alkalmazást és a szoftverpiaci kínálatot is figyelembe véve – általában szétválasztott modellt, vagy térbeli adattípusokra épülő integrált megoldást választanak. (Például szétválasztott modell esetén a *Telek* objektumot *TELEK réteggel* és *Telek adattáblával* helyettesítik. Integrált modell esetén egyetlen *Telek tábla* lesz, amely térbeli oszlopot tartalmaz.)

## 4. RASZTERES TÉRINFORMATIKAI RENDSZEREK

Általában a természeti környezet leírására szolgálnak. Folytonos változású jelenségek jól modellezhetők a segítségükkel (pl. domborzat, légszennyezés, meteorológia, talajminőség, növénykultúrák, népsűrűség, stb.).

Egy adott terület leírására általában több egymásra helyezett raszter réteget használnak. Egy réteg egy adott jellemző leírására szolgál.

*Fedvény:* egy vagy több, tartalmilag összetartozó réteg, az esetleges kapcsolódó adattáblákkal.

Raszteres adat előállítás:

- távérzékelés (műholdkép, légifénykép)
- szkennelés
- vektor-raszter konverzió
- diszkrét pontokban mért értékekből interpolációval (ezzel majd a Digitális Terepmodellnél foglalkozunk részletesen)

Az egyes raszterpontokhoz tartozó számértékek jelentése lehet

- mennyiségi jellemző, pl. domborzatnál terepmagasság.
- minőségi jellemző, pl. növénykultúra, talaj fizikai minősége:
  - 1 – homok,
  - 2 – vályog,
  - 3 – agyag.

*Felbontás (geometriai):* egy raszterpont a valós világban mekkora területnek (négyzetnek) felel meg. Az adott raszterponthoz tartozó számérték a folytonos jellemző átlagértékének felel meg.

*Megjelenítési módok:*

– Minden raszterpont értékhez más szín. A raszterkép mellett a képernyőn általában színregiszter adja a jelmagyarázatot. Például:

1 (homok):	szürke
2 (vályog):	világosbarna
3 (agyag):	sötétbarna

– Folytonosan változó érték esetén folytonos színskála (pl. domborzatnál (zöld...barna skála)

– Perspektivikus megjelenítés (elsősorban domborzatnál): perspektivikus háló, esetleg takart vonalak törlésével, változtatható nézőpontból.

– Több réteg együttes megjelenítése: pl. domborzat és légifénykép réteg megjelenítése perspektivikusan.

## 4.1. Adatbázis kapcsolat

Raszteres rendszereknél az adatbázis kapcsolat jelentősége kisebb. Az alábbiakban két jellemző példát említünk.

*Pixelértékhez adatrekord kapcsolása.* Tekintsünk egy talajtérképet, ahol a rajz minden pixele adott talajminőséget kódol (például 1: homok, 2: vályog, 3: agyag). Az egyes talajtípusok vegyi, mechanikai, stb. adatait egy adattáblában tároljuk, amely táblát bővítünk a talajtérképen használt számkóddal. Ezáltal minden egyes raszterpont kapcsolatba kerül a megfelelő adatbázis rekorddal.

*Adatrekordhoz pixel koordináta kapcsolása.* Ebben az esetben valamely raszteres objektum referenciapontjának koordinátáit helyezik el a kapcsolt adatrekordban.

Például, ha ingatlan nyilvántartás esetén csak szkennelt raszteres térképek állnak rendelkezésre, a TELEK adattábla a következőképp épülhet fel:

TELEK (hrs, terület, térkép, x, y)

ahol *térkép* a megfelelő térképszelvényt tartalmazó raszteres állomány neve,  $(x, y)$  pedig az adott raszter állományon az adott telek referencia pontjának (pixel)koordinátái.

Korrektebb megoldást kapunk a szkennelt szelvények geokódolásával. Vegyünk fel egy TERKÉP (szelvényszám, x1, y1, x2, y2) táblát, amelynek egy rekordja egy adott térképszelvény bal alsó és jobb felső sarkának geodéziai koordinátáit tartalmazza! Ezek után a TELEK táblában  $(x, y)$  geodéziai koordináták lehetnek, és a *térkép* attribútum akár el is hagyható.

## 4.2. Bevezető példa

*Feladat:* a fakitermelésre alkalmas területek kiválasztása. A megfelelő terület az alábbi jellemzőkkel rendelkezik:

- erdei fenyő (lucfenyő nem felel meg),
- megfelelő a talajminőség (a nem megfelelő talaj nem bírja el a gépeket, a fakitermelés elfogadhatatlan környezeti károkat okoz),
- nem lehet 500 m-nél közelebb tó vagy vízfolyás (az erózió rontja a vízminőséget),

Rendelkezésre álló raszteres adatok (felbontás: 1 pixel = 100 méter):

- T: talajminőség térkép. A 3-as és 5-ös érték a megfelelő.
- N: növénykultúra térkép. A 2-es érték felel meg az erdei fenyőnek.
- V: vízrajzi térkép. 0 = szárazföld, 1 = víz (tó, folyó, stb.).

*Megoldás:*

– A V rétegen az 1 értékű pixeleket terjesszük ki 5 lépésen keresztül (dilatációs művelet, egy lépése: 1-re vált minden olyan pixel, amelynek van 1-es szomszédja). Így kapjuk a V1 réteget.

– A T, N, V1 rétegekből pixelenkénti művelettel állítjuk elő az E eredményréteget:

$E_{i,j} := (T_{i,j} = 3 \text{ OR } T_{i,j} = 5) \text{ AND } N_{i,j} = 2 \text{ AND } (\text{NOT } V1_{i,j})$

– Ahol  $E = 1$ , ott lehet fakitermelést végezni.

A továbbiakban a raszteres térinformatikai rendszerek jellemző műveleteit tekintjük át. Egy művelet általában egy  $A$  fedvényből egy  $B$  fedvényt állít elő.

### 4.3. Pixelenkénti műveletek

Egy  $Z$  fedvényből egy  $Z'$  fedvényt képezünk, ahol  $z'_{i,j}$  értéke csak  $z_{i,j}$  értékétől függ.

(i) *Formulával definiált művelet*:  $z'_{i,j} := f(z_{i,j})$ . A formulától függően információvesztés lehet.

Példa:  $z'_{i,j} := 2 * z_{i,j} + 3$ .

(ii) *Átkódolás*: minden lehetséges  $Z$ -beli pixelértékhez megadjuk a megfelelő  $Z'$ -beli értéket. Ha különböző  $Z$ -beli értékekhez különböző  $Z'$ -beli értékek tartoznak, akkor nincs információvesztés.

Példa: A  $Z$  fedvényben minden raszterpontban az utolsó erdőtüz évszámát tároljuk. Az évszám értékeket nagyság szerint rendezzük (pl. 0 = nem volt erdőtüz, 1931, 1964, 1982), majd ezeket rendre a 0, 1, 2, 3 értékekkel helyettesítjük  $Z'$ -ben. Ezzel tárolóhelyet takarítunk meg. A tényleges évszámértékek egy kapcsolódó adattáblában tárolhatók:

0	0
1	1931
2	1964
3	1982

(iii) *Intervallumba sorolás*: Értékintervallumokhoz rendelhetünk új értékeket (pl. 0...499 lesz az 1, 500...999 lesz a 2, 1000-nél nagyobb érték 3 lesz az átkódolás után). Akkor alkalmazhatjuk, ha az eredeti réteg pontról pontra változó értékeket tartalmaz (pl. terepmagasság, csapadékmennyiség). Az értékosztályokat definiáló intervallum határok itt is egy kapcsolt adattáblában tárolhatók. A tárolóhely megtakarítás most azonban információvesztéssel jár!

(iv) *Osztályba sorolás (clusterezés)*: a kép pontjait a  $C^{(1)}$ , ...,  $C^{(r)}$  osztályokba soroljuk. Ha  $z_{i,j}$  a  $C^{(k)}$  osztályba sorolódik, akkor az eredmény fedvényben  $z'_{i,j}$  értéke  $k$  lesz.

Példa: 16 sávós műholdkép alapján terméseredmény becslés. Egy pixel értéke egy  $z = (z_1, \dots, z_{16})$  vektor. Az osztályok a  $C^{(i)} = (c^{(i)}_1, \dots, c^{(i)}_{16})$  osztály reprezentáns pontok ( $i=1, \dots, r$ ) segítségével adottak, például  $C^{(1)}$  az erdőt,  $C^{(2)}$  a szántót jellemző tipikus pixel érték, stb. A  $z$  pontot a  $C^{(1)}, \dots, C^{(r)}$  osztályok közül  $C^{(k)}$ -ba soroljuk, ha valamely  $d$  távolságfüggvény szerint a  $d(z, C^{(k)})$  távolság minimális. Például  $d(z, C^{(k)}) = |z_1 - c^{(k)}_1| + \dots + |z_{16} - c^{(k)}_{16}|$  módon számítható.

(v) *Több réteg közötti művelet*: az  $Z_1, \dots, Z_n$  fedvényekből állítunk elő egy  $Z'$  fedvényt például aritmetikai, logikai, max, min, stb. műveletek felhasználásával.

Példa: a fakitermelési feladatban

$E_{i,j} := (T_{i,j} = 3 \text{ OR } T_{i,j} = 5) \text{ AND } N_{i,j} = 2 \text{ AND } (\text{NOT } \forall i,j)$

## 4.4. Lokális szomszédsági műveletek

Az új fedvény értékeit a bemenő fedvény pontjainak szomszédjai határozzák meg:  $z'_{i,j} = f(z_{i,j} \text{ környezete})$ .

### 4.4.1. Konvolúció

Egy  $W$   $(2n+1) \times (2n+1)$ -es ablak-mátrixot mozgatunk a raszter pontjai fölött:

$$W = \begin{bmatrix} w_{-n,-n} & \cdots & w_{-n,n} \\ \vdots & & \vdots \\ w_{n,-n} & \cdots & w_{n,n} \end{bmatrix}$$

A szokásos ablak  $3 \times 3$  elemből áll ( $n = 1$ ). Az ablak közepére eső raszterpont új értékét az ablakba eső értékek súlyozott középértékéből képezzük:

$$z'_{i,j} = \sum_{u=-n}^n \sum_{v=-n}^n w_{u,v} z_{i+u,j+v}$$

Példák:

– Simítás (helyi részletek eltüntetése) az alábbi ablakkal:

$$\begin{array}{ccc} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{array}$$

illetve általában:

$$z'_{i,j} = \frac{1}{(2n+1)^2} \sum_{u=-n}^n \sum_{v=-n}^n z_{i+u,j+v}$$

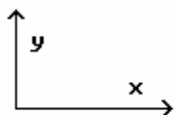
– Élkiemelés (helyi részletek kiemelése) az alábbi ablakkal:

$$\begin{array}{ccc} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{array}$$

### 4.4.2. Lejtés és lejtésirány

Tekintsük a  $h(x, y)$  függvényt, amely minden  $(x, y)$  pontban megadja a tengerszint feletti magasságot. Gradiens vektor:  $(\partial h / \partial x, \partial h / \partial y)$ , ebben az irányban változik a  $h(x, y)$  függvény a leggyorsabban (a szintvonalra merőleges, az emelkedés iránya).

Legyen a  $Z$  fedvény a  $h(x, y)$  függvény diszkrét változata, vagyis  $z_{i,j}$  az  $(i, j)$  pontnak megfelelő négyzethez tartozó átlagos terepmagasságot adja.



$z[i, j-1]$	$z[i, j]$
	$z[i+1, j]$

Digitális gradiens: ( $D_x$ ,  $D_y$ ) ahol

$$D_x[i, j] = z[i, j] - z[i, j-1]$$

$$D_y[i, j] = z[i, j] - z[i+1, j]$$

$$\text{Lejtőmeredekség fedvény (D): } d_{i,j} = \sqrt{D_x[i, j]^2 + D_y[i, j]^2}$$

Lejtésirány fedvény (A):  $\alpha_{i,j} = \arctg(D_y[i, j]/D_x[i, j])$  alapján számítható:

$D_x$	$D_y$	$\alpha$
0	> 0	90°
0	< 0	-90°
> 0	0	0°
< 0	0	180°
> 0	≠ 0	$\arctg(D_y/D_x)$
< 0	≠ 0	$\arctg(D_y/D_x) + 180^\circ$

Alkalmazás: erózió és vízlefolyás modellezése, vagy árnyékolt domborzat megjelenítés.

*Lejtőkategória térkép:* a D lejtőmeredekség fedvényt intervallumba sorolással tartománytérképpé alakítjuk (pl. az egyes tartományok: 0...3, 3...6, 6...10, 10...20, 20...35, 35-nél nagyobb fokos lejtés).

*Lejtőkitettségi térkép:* az A lejtésirány fedvényt intervallumba sorolással tartománytérképpé alakítjuk (pl. az egyes tartományok: É, ÉK, K, DK, D, DNy, Ny, ÉNy fekvésű lejtők).

## 4.5. Távolsági műveletek

### 4.5.1. Távolság fedvény készítése

Az új fedvényen minden pixel értéke egy adott cellacsoporttól mért távolság lesz.  
Alkalmazás: pl. szennyezett területtől való távolság.

Távolság fogalma: a pontos euklideszi távolságot különféle módokon közelítik:

– 4 szomszédos távolság (manhattan-távolság, city block distance): két cella távolsága a minimális 4-szomszédos lépésszám, amellyel egyikből a másikba el lehet jutni.

– 8 szomszédos távolság: hasonlóan, az átlós szomszédokat is figyelembe véve.

– egyéb közelítések.

A továbbiakban 4-szomszédos távolságot tételezünk fel.

*Iterációs algoritmus:* szomszédról szomszédra terjesztéssel és inkrementálással történik.  $n \times n$  méretű fedvényt tételezünk fel. Kiinduláskor 0 értékűek a szennyezett területhez tartozó pixelek, a többi pixel plusz végtelen értékű (pontosabban, a számítógépen ábrázolható legnagyobb számnál eggyel kisebb), ezt az értéket X-szel jelöljük. Egy iterációs lépés egy pixellel terjeszti a távolságértékeket, a  $k$ -edik lépésben  $Z^{(k)}$ -ből  $Z^{(k+1)}$ -et számolunk ( $Z = Z^{(1)}$ ):

```
for i=1 to n
for j=1 to n
 $z^{(k+1)}_{i,j} := \min(z^{(k)}_{i,j}, z^{(k)}_{i,j-1}+1, z^{(k)}_{i,j+1}+1, z^{(k)}_{i-1,j}+1, z^{(k)}_{i+1,j}+1)$ 
```

Az eljárást meg lehet szakítani, ha már nincs több X értékű cella. Legrosszabb esetben  $n$  iterációs lépés szükséges, ekkor az időigény  $O(n^3)$ .

*Lineáris idejű algoritmus:* a  $Z$  fedvényen kétszer kell végigmenni (magában a  $Z$  tömbben számolunk), az összes időigény  $O(n^2)$ :

1. Előre haladó fázis:

```
for i=2 to n
for j=2 to n
 $z_{i,j} := \min(z_{i,j}, z_{i-1,j}+1, z_{i,j-1}+1)$ 
```

2. Visszafelé haladó fázis:

```
for i=n-1 to 1
for j=n-1 to 1
 $z_{i,j} := \min(z_{i,j}, z_{i+1,j}+1, z_{i,j+1}+1)$ 
```

#### 4.5.2. Övezetképzés

Az övezet (sáv, angolul buffer zone) úgy szemléltethető, mint az objektum térbeli kiterjesztése adott távolsággal. Az eredmény egy fedvény lehet az alábbi értékekkel:

- 2, ha az eredeti objektumon belül vagyunk
- 1, ha az övezeten belül vagyunk
- 0, ha az objektumon és az övezeten is kívül vagyunk

Alkalmazások: zajzóna az utak mentén, védőtávolság a veszélyes berendezések körül.

*Iterációs övezetképző algoritmus:* kezdetben az objektum pontjai 2 értékűek, minden más pont 0. Egy iterációs lépés, amely  $Z^{(k)}$ -ből egy  $Z^{(k+1)}$  fedvényt számol:

```
for i=1 to n
for j=1 to n
if  $z^{(k)}_{i,j}=0$  AND ( $z^{(k)}_{i,j-1} > 0$  OR  $z^{(k)}_{i,j+1} > 0$  OR  $z^{(k)}_{i-1,j} > 0$  OR  $z^{(k)}_{i+1,j} > 0$ )
then  $z^{(k+1)}_{i,j} := 1$ 
```

Ha ezt  $r$  iterációs lépésen keresztül ismétljük, akkor  $r$  széles övezet keletkezik. Időigény:  $O(r \cdot n^2)$ .

*Lineáris idejű algoritmus:* távolság fedvényt készítünk, amelyet  $r$  értékével küszöbölünk.

### 4.5.3. Övezetképzés moduláló fedvénnyel

A kiterjesztés mértékét befolyásolhatja egy másik fedvény, amely folytonosan változó értékeket tartalmaz (moduláló fedvény). (Pl. moduláló fedvénnyel modellezhetők a zajterjedést gátló tereptárgyak.) Ez befolyásolja az övezet szélességét – szűkülést okoz a nagy moduláló értékek helyén.

Egy objektumon kívüli  $z_{i,j}$  pont kezdőértéke nem 0, hanem  $-m_{i,j}$ , ha  $m_{i,j}$  az adott ponthoz tartozó moduláló érték. A számítás:

```
for i=1 to n
for j=1 to n
if  $z^{(k)}_{i,j} < 1$  AND ( $z^{(k)}_{i,j-1} > 0$  OR  $z^{(k)}_{i,j+1} > 0$  OR  $z^{(k)}_{i-1,j} > 0$  OR  $z^{(k)}_{i+1,j} > 0$ )
then  $z^{(k+1)}_{i,j} := z^{(k)}_{i,j} + 1$ 
```

*Példa:* autópálya mentén zajzóna számítása. A zajterjedést gátló építményeket moduláló fedvény kódolja (29. ábra).

Az autópálya fedvény	Moduláló fedvény	Egyesített fedvény
. . . 2 . . . . .	. . . . .	. . . 2 . . . . .
. . . 2 . . . . .	. . . . .	. . . 2 . . . . .
. . . 2 . . . . .	. . . . .	. . . 2 . . . . .
. . . 2 . . . . .	. . . . 3 . . . . .	. . . 2 . -3 . . . . .
. . . 2 . . . . .	. . . . 3 . . . . .	. . . 2 . -3 . . . . .
. . . 2 . . . . .	. . . . 3 . . . . .	. . . 2 . -3 . . . . .
. 2 2 . . . . .	. . . . 3 . . . . .	. 2 2 . -3 . . . . .
. 2 . . . . .	. . . . .	. 2 . . . . .
2 2 . . . . .	. . . . .	2 2 . . . . .
2 . . . . .	. . . . .	2 . . . . .
1. lépés	2. lépés	3. lépés
. 1 2 1 . . . . .	1 1 2 1 1 . . . . .	1 1 2 1 1 1 . . . . .
. 1 2 1 . . . . .	1 1 2 1 1 . . . . .	1 1 2 1 1 1 . . . . .
. 1 2 1 . . . . .	1 1 2 1 1 . . . . .	1 1 2 1 1 1 . . . . .
. 1 2 1-3 . . . . .	1 1 2 1-2 . . . . .	1 1 2 1-1 . . . . .
. 1 2 1-3 . . . . .	1 1 2 1-2 . . . . .	1 1 2 1-1 . . . . .
. 1 2 1-3 . . . . .	1 1 2 1-2 . . . . .	1 1 2 1-1 . . . . .
1 2 2 1-3 . . . . .	1 2 2 1-2 . . . . .	1 2 2 1-1 . . . . .
1 2 1 . . . . .	1 2 1 1 . . . . .	1 2 1 1 1 . . . . .
2 2 1 . . . . .	2 2 1 1 . . . . .	2 2 1 1 1 . . . . .
2 1 . . . . .	2 1 1 . . . . .	2 1 1 1 . . . . .
4. lépés	5. lépés	6. lépés
1 1 2 1 1 1 1 . . .	1 1 2 1 1 1 1 1 . .	1 1 2 1 1 1 1 1 1 .
1 1 2 1 1 1 1 . . .	1 1 2 1 1 1 1 1 . .	1 1 2 1 1 1 1 1 1 .
1 1 2 1 1 1 1 . . .	1 1 2 1 1 1 1 1 . .	1 1 2 1 1 1 1 1 1 .
1 1 2 1 . 1 . . . . .	1 1 2 1 1 1 1 . . .	1 1 2 1 1 1 1 1 . .
1 1 2 1 . . . . .	1 1 2 1 1 1 . . . . .	1 1 2 1 1 1 1 . . .
1 1 2 1 . . . . .	1 1 2 1 1 . . . . .	1 1 2 1 1 1 . . . . .
1 2 2 1 . . . . .	1 2 2 1 1 1 . . . . .	1 2 2 1 1 1 1 . . .
1 2 1 1 1 1 . . . . .	1 2 1 1 1 1 1 . . . . .	1 2 1 1 1 1 1 1 . .
2 2 1 1 1 1 . . . . .	2 2 1 1 1 1 1 . . . . .	2 2 1 1 1 1 1 1 . .
2 1 1 1 1 . . . . .	2 1 1 1 1 1 . . . . .	2 1 1 1 1 1 1 . . .

29. ábra: Övezetképzés moduláló fedvénnyel. A '.' karakter 0 értéket jelent.



#### 4.5.4. Láthatóság

Adott egy fedvényünk magassági adatokkal és egy nézőpont, ki kell számítanunk az adott pontból látható területeket. Az eredmény egy fedvény, ahol 0 a nem látható, 1 a látható pontok értéke. Alkalmazás: távközlési berendezések helyének tervezése, a tájképet zavaró objektumok (pl. gyárkémény) helyének megtervezése. Algoritmus: az adott nézőpontból sugárirányban kiinduló metszetsvonalak mentén vizsgálni, hogy a nézőpontot a célponttal összekötő egyenes metszi-e a terepfelszínt.

#### 4.5.5. Műveletek tartományokon

Az azonos számértékekből álló összefüggő területet *tartomány*nak (*foltnak*) nevezzük. Az összefüggőséget 4-szomszédság alapján definiáljuk.

Átkódolás (connected component labelling): Minden tartomány egyedi azonosítót kap, minden képpont a tartományban átveszi a tartomány azonosító számát. Az alábbi példában a szemléletesség kedvéért tartománykódként az A, B, C, D, E betűjeleket használtuk, valójában ezek helyett gépi adatábrázolásban az 1, 2, 3, 4, 5 számok szolgálhatnak:

3	3	1	1	A	A	B	B
1	3	1	2	B	A	B	C
1	1	1	2	B	B	B	C
3	3	3	1	D	D	D	E

*Lineáris idejű algoritmus:*

1. Sorfolytonosan haladunk a fedvényen, egyszerre két sort figyelünk: az aktuálisat és a felette lévő. Értelemszerűen képezzük az output fedvény elemeit, az első két sor feldolgozása után az alábbi adódik:

A	A	B	B
C	A	B	D

Itt még nem tudhatjuk, hogy a B-vel és C-vel jelölt tartományok valójában megegyeznek, ezért jelöltük különböző betűvel. A folytatás:

A	A	B	B
C	A	B	D
C	C	C	D

Itt vesszük észre, hogy a B és C tartományok megegyeznek, ezért feljegyezzük: B=C. A folytatás:

A	A	B	B
C	A	B	D
C	C	C	D
E	E	E	F

2. Újra végigmegyünk a mátrixon, és a feljegyzésnek megfelelően (esetünkben B=C) lecseréljük a betűket.

A	A	C	C
C	A	C	D
C	C	C	D
E	E	E	F

3. Eredménymátrix átkódolása folyamatos kódkiosztásra, vagyis A, C, D, E, F helyett A, B, C, D, E:

A	A	B	B
B	A	B	C
B	B	B	C
D	D	D	E

A fenti algoritmus 1. lépésének pontos leírása: Felveszünk egy N elemű T tömböt a kódolás tárolására. (Feltesszük, hogy legfeljebb N különböző tartomány van a fedvényen.) Ha a  $T[i]$  elem értéke pozitív, akkor az  $i$  azonosítási számú tartomány pixel értékét adja, ha  $T[i]$  negatív, akkor  $|T[i]|$  egy másik T-beli elemre mutató pointert jelent. Egy  $i$  tartománykódhoz tartozó pixel értéket úgy kapjuk meg, hogy  $T[i]$ -ből kiindulva addig haladunk a negatív pointer hivatkozásokat követve, amíg pozitív elemet nem találunk. Belátható, hogy a T tömbbeli tartománykódok fastruktúrákat alkotnak, minden egyes fastruktúra gyökerénél egy pozitív elem van. Jelölje  $m$  a T első szabad helyét, kezdetben  $m=0$  (T üres).

Sorfolytonosan járjuk be a mátrixot. Egyszerre mindig két sort figyelünk, az aktuálisat és a megelőzőt. Az aktuális sorban azonos értékű összefüggő pixelcsoportokat vizsgáljuk, és megfelelő tartománykóddal helyettesítjük.

– Ha az aktuális pixel csoporttal szomszédos azonos értékű pixel nincs az előző sorban, akkor új azonosítási számot osztunk ki:  $T[m]$ -be beírjuk az aktuális pixel értéket, a pixeleket pedig  $m$ -mel helyettesítjük, majd  $m := m+1$ .

– Ha az előző sorban vannak az aktuális pixel csoporttal azonos értékű pixelek, és ezekhez ugyanazon azonosítási szám tartozik, akkor az aktuális pixel csoportot ezzel az azonosítási számmal helyettesítjük.

– Ha az előző sorban vannak az aktuális pixel csoporttal azonos értékű pixelek, de ezekhez különböző – mondjuk X és Y – azonosítási szám tartozik, akkor az egyik azonosítási számmal – mondjuk X-szel – helyettesítjük az aktuális pixel csoportot, és az X-hez illetve Y-hoz tartozó fákat egyesítjük T-ben. Ez utóbbi azt jelenti, hogy  $T[X]$ -ből és  $T[Y]$ -ből kiindulva megkeressük a fák  $X1$  és  $Y1$  gyökerét, majd  $T[X1] := Y1$  hivatkozással a fákat összekapcsoljuk.

A fenti példa esetén az alábbi T tömb és tartománykódolás alakul ki:

	A	B	C	D	E	F	
	3	C	1	2	3	1	
3	3	1	1				A A B B
1	3	1	2				C A B D
1	1	1	2				C C C D
3	3	3	1				E E E F

*A tartomány területe* (pixelben): a tartományba eső pixelek darabszáma.

*A tartomány kerülete* (pixelben): a határoló cellák külső élének száma.

Megjegyzés: a fenti módon számított terület és kerület nagyban függ a tartományoknak a raszterhez viszonyított irányától. Ha azonban a vizsgált terület határának nincs kitüntetett iránya, ez a hiba nem jelentős.

*A tartomány alakja:* Az egyik leggyakoribb meghatározási módja az alakmérésnek, ha a kerületet elosztjuk a tartomány területének négyzetgyökével. Elosztva ezt a számot 3.54-el olyan mértéket kapunk ( $K/(\sqrt{T} \cdot 3,54)$ ), ami a körnél 1 (a lehető legtömörebb alak), 1.13 a négyzetnél, és növekvő szám a hosszú, keskeny alakzatok esetében. Az alakmérést például tájökölógiai vizsgálatoknál alkalmazzák (Mezősi és tsai, 1993).

A tartományokat egy relációs adattáblában tarthatjuk nyilván, amely pl. az alábbi mezőket tartalmazza:

tartomány azonosító:	A	B	C	D	E
eredeti pixel érték:	3	1	2	3	1
terület:	3	7	2	3	1
kerület:	8	16	6	8	4
alak:	1.3	1.7	1.2	1.3	1.13
egyéb attribútumok	...	...	...	...	...

Példa tartományokkal kapcsolatos speciális műveletre: Határozzuk meg minden pont távolságát az öt tartalmazó tartomány határának legközelebbi részétől, majd ez legyen minden pont értéke. (A határt úgy definiálhatjuk, mint olyan pontokat, amelyek szomszédságában lévő pontoknak más értéke van.) A kapott pixelérték a pont 'stabilitását' jellemzi, vagyis azt, hogy milyen mértékben veszik körül azonos értékű pontok.

## 5. DIGITÁLIS TEREPMODELLEK

A Föld felszínének leírására szolgáló számítógépes modelleket *digitális terepmodell*nek (*DTM = Digital Terrain Model*) nevezik.

Általában feltételezik, hogy a felszín egy kétváltozós  $h(x, y)$  függvénnyel leírható, ahol  $x, y$  a felszín egy adott pontjának koordinátái,  $h(x, y)$  pedig az adott pontban mért (tengerszint feletti) magasság. Ezzel az ún. 2.5 dimenziós modellezési technikával bizonyos felszíni képződményeket (pl. kihajló sziklákat) csak közelítően lehet leírni, könnyű kezelhetősége folytán mégis ezen modell alkalmazása vált általánossá.

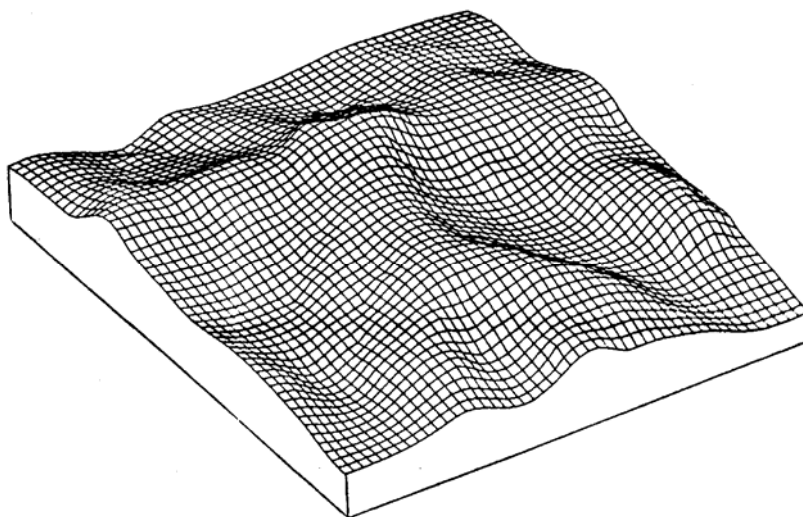
A  $h(x, y)$  függvényt "majdnem mindenütt" folytonosan differenciálhatónak tételezzük fel. A kivételes helyeket a DTM előállításakor külön jelölni kell:

- szakadásvonal (tereplépcső):  $h(x, y)$  nem folytonos.
- törésvonal:  $h(x, y)$  deriváltja nem folytonos.

Alapvetően raszteres és vektoros DTM-et különböztethetünk meg:

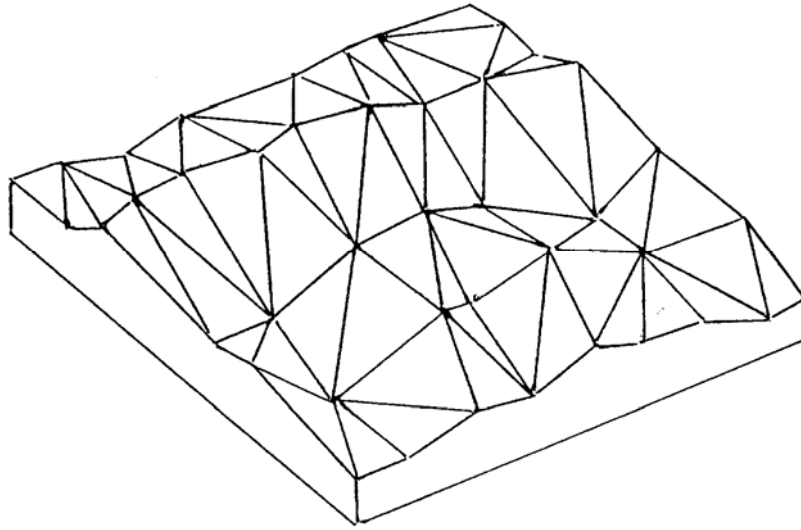
a). *DEM = Digital Elevation Model*: raszteres DTM, ahol a  $h(x, y)$  függvényt egy  $H[i, j]$  mátrixszal közelítjük (30. ábra). Jellemző paraméterei:

- felbontás: egy raszterpontnak megfelelő négyzet alakú terület oldalhossza. Tipikus érték: 20 m.
- pontosság: a magasságérték legkisebb egysége, kisméretarányú modelleknél általában 1 méter.



30. ábra: Raszteres terepmodell (DEM) 3D ábrázolása.

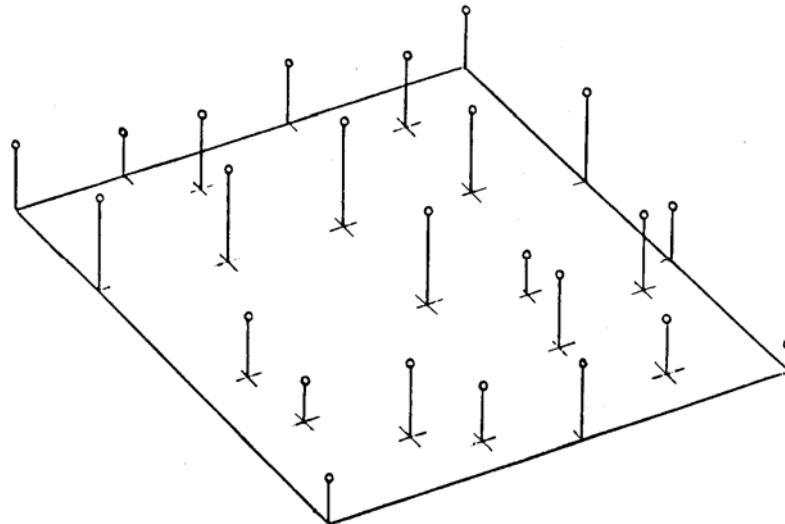
b). *TIN = Triangulated Irregular Network*: vektoros DTM, ahol a felszínt szabálytalanul elhelyezett háromszöglapokkal közelítjük (31. ábra). A szögpontok magasságértéke és összekapcsolási struktúrája kerül tárolásra. A háromszögek elhelyezése a terepviszonyoktól függ (alföldön nagyméretű, hegyvidéken a domborzatot követő, kisebb háromszögek alkalmazhatók). A lehetséges adatstruktúrákat a vektoros adatmodelleknél ismertettük.



31. ábra: Vektoros terepmodell (TIN) 3D ábrázolása.

A DTM előállításához az alábbi forrásadatok használatosak:

1. Mért magassági pontok: terepi felméréssel a jellemző tereppontok magasságát megméri (32. ábra).
2. Távérzékelés: sztereo légifénykép pár feldolgozása fotogrammetriai úton vagy szoftverrel.
3. Szintvonalas térkép: meglévő papírtérkép állomány felhasználása.



32. ábra: Egyedi magasságpontok.

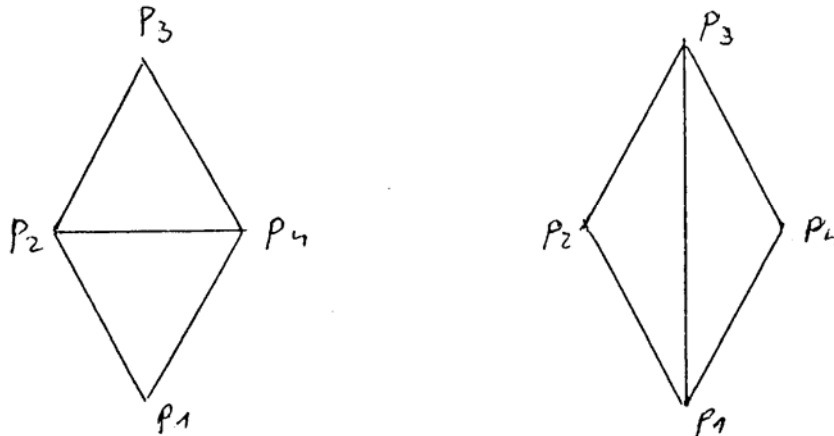
## 5.1. TIN előállítása magassági ponthalmazból

Adott  $(x, y, z)$  pontok halmaza, és az a kérdés, hogyan kössük össze ezeket egy TIN-háromszögrácsá. A szokásos eljárást Delaunay-triangularizációnak nevezik, ezt ismertetjük alább.

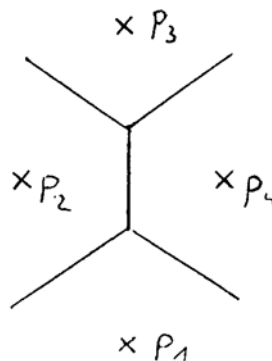
A háromszögek kitűzésénél az  $(x, y, z)$  pontoknak csak az  $(x, y)$  síkbeli vetületeit vizsgáljuk. A cél az, hogy minél "kövérebb" háromszögeket alkossunk (amelynek szögei minél közelebb állnak a 60 fokhoz), ugyanis a háromszög belső pontjai ekkor kerülnek a legközelebb az adott értékű csúspontokhoz (33. ábra). Az alábbi két definíció egyenértékű:

1. *definíció:* Egy ponthalmazon három pont *Delaunay háromszöget* alkot, ha a rajtuk áthaladó kör nem tartalmaz további szögpontot. Egy ponthalmaz triangularizációját Delaunay-triangularizációnak nevezzük, ha bármely háromszög Delaunay-háromszög.

2. *definíció:* Osszuk fel a területet tartományokra úgy, hogy minden pontot a hozzá legközelebbi szögponthoz sorolunk (Voronoi diagram). Tehát minden szögponthoz tartozik egy tartomány, ún. Voronoi poligon vagy Thiessen poligon (34. ábra). A ponthalmaz egy triangularizációját Delaunay-triangularizációnak nevezzük, ha pontosan azokat a szögpontokat köti össze él, amelyek Voronoi poligonjai szomszédosak.



33 ábra: Négy szögpont helyes és helytelen triangularizációja.



34. ábra: Négy szögpont Voronoi poligonjai.

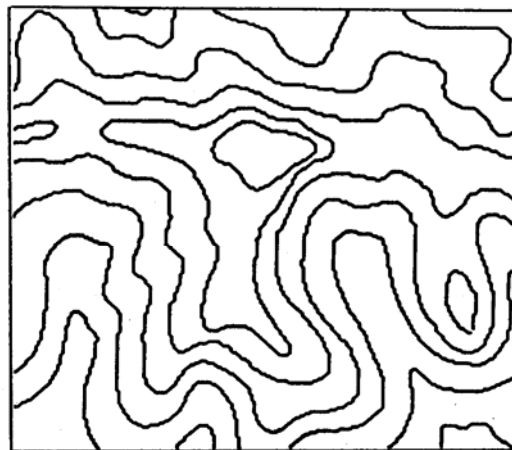
*Triangularizációs algoritmus:* Kössük össze a két, egymáshoz legközelebbi szögpontot (ez szükségképpen Delaunay él). Ezután olyan harmadik pontot keresünk, hogy a rajtuk áthaladó kör ne tartalmazzon további pontot. Folytassuk az eljárást, mindig a legközelebbi pontok vizsgálatával.

## 5.2. TIN előállítása szintvonalrajzból

*Az eljárás lépései:*

1. Szintvonalrajz szkennelése.
2. Raszterkép manuális javítása (szakadások és összeérések megszüntetése, speciális jelkulcsi elemek törlése).
3. Vektorizálás (automatikusan vagy manuálisan).
4. Szintvonalakhoz magasság érték rendelése. Az eljárás félautomatikusan történhet, például párhuzamosan futó szintvonalkötegeknél csoportos értékadással.
5. Háromszögek kijelölése a szintvonalak között Delaunay eljárással (5.1. fejezet), a háromszögek egy (vagy két) oldalát a szintvonal adja.

*Az eljárás hátránya,* hogy lapos részeket generál (például hegycsúcsot körülvevő zárt szintvonalnál, vagy a 35. ábra közepén látható hegyhátnál). Ezen hátrányok pontsűrítéssel küszöbölhetők ki: a kritikus helyeken újabb pontokat veszünk fel, például térbeli interpolációval (lásd alább).



35. ábra: Digitalizált, vékonyított szintvonalrajz.

## 5.3. Térbeli interpolációs módszerek

Legyen adott az  $f(x, y)$  függvény értéke az  $(x_1, y_1), \dots, (x_m, y_m)$  pontokban, legyenek ezek az értékek  $d_1, \dots, d_m$ . Szeretnénk a függvény értékét további pontokban – például adott

felbontású négyzetrács pontjaiban, vagy TIN sűrítése céljából – interpolációval becsülni. Néhány fontosabb eljárást ismertetünk.

### 5.3.1. Távolság inverzével súlyozott mozgóátlag

A módszer lényege, hogy a meghatározandó  $P$  pont környezetében kiválasztanak  $n$  ismert magasságú pontot ( $P_1, \dots, P_m$ ), és ezek  $d_1, \dots, d_m$  magasságértékéből átlagolják a  $P$  pont  $z$  magasságértékét. Jellemzően a

$$z = (d_1/t_1 + \dots + d_m/t_m) / (1/t_1 + \dots + 1/t_m)$$

formula használatos, ahol  $t_i$  a  $P_i$  pontnak  $P$ -től való távolságát jelenti. A távolság inverzével való súlyozás nyilván a közelebbi pontoknak ad nagyobb szerepet, az  $(1/t_1 + \dots + 1/t_m)$  tényezőnek pedig normáló szerepe van (ha  $d_1 = \dots = d_m$ , akkor  $z$  ezt a közös értéket kell hogy adja).

Szintvonalrajz esetén a módszer a következőképp alkalmazható. Indítsunk  $m$  egyenest a  $P$  pontból  $360/m$  fokos szöginkrementummal, és legyenek  $P_1, \dots, P_m$  azon szintvonalpontok, amelyeket ezen egyenesek a  $P$  pontból indulva elsőként metszenek! Nevezzük ezeket a  $P$ -ből látható szintvonalpontoknak. Az így kiválasztott pontokra végezzük az interpolációt.

Az eljárás hátránya, hogy lokális minimumok és maximumok csak az adott pontokban (szintvonalakon) léphetnek fel. Ez elsősorban zárt görbét alkotó szintvonalak (például hegycsúcsok) esetén okoz hibát, ugyanis a görbén belül konstans (lapos) felület keletkezik, mivel az eljárás minden irányban azonos magasságértékeket érzékel. További hátrány, hogy az eljárás "nem lát át" a szintvonalakon, ezért a szintvonalaknál a terepen természetellenes törés keletkezhet.

### 5.3.2. Polinomiális interpoláció

Legyen adott az  $f(x, y)$  függvény értéke az  $(x_1, y_1), \dots, (x_m, y_m)$  pontokban, legyenek ezek az értékek  $d_1, \dots, d_m$ . Az  $f$  függvényt a legkisebb négyzetek módszerével szeretnénk közelíteni, vagyis olyan  $p(x, y)$   $r$ -edfokú polinomot találni, amelyre az

$$E = \sum_i (p(x_i, y_i) - d_i)^2$$

összeg minimális. A módszer lényegét a koordináta transzformációknál (2.3. fejezet) már tárgyaltuk, a megoldást az ott leírtakhoz hasonlóan nyerjük (az eljárás részletes levezetését a Függelék tartalmazza).

A polinomiális interpoláció a következőképp értékelhető:

- Az eddigi megközelítésekkel szemben itt lokális minimumok és maximumok nem csak az adott pontokban léphetnek fel, tehát realiztikusabb eredmény várható.
- Szintvonalrajz esetén az eljárás meglehetősen számításigényes, az adott pontok (szintvonalpontok) nagy száma miatt.
- Az eljárás nem veszi figyelembe a terep fizikai jellemzőit. Ebben a tekintetben majd a variációs spline interpoláció jelent előrelépést (lásd később).



## 5.4. DEM előállítása térbeli interpolációval

Források: szintvonalas térkép és/vagy mért magassági pontok. A szintvonalrajz előkészítése hasonlóan történhet, mint ahogy a TIN előállításánál láttuk.

### 5.4.1. Az Intercon módszer

Az IDRISI térinformatikai szoftver Intercon modulja az alábbi egyszerű interpolációs módszert alkalmazza, kimondottan szintvonalrajzra:

– Vízzintes, függőleges és átlós irányú metszeteket készít a generálandó DEM felbontásának megfelelően.

– Minden metszet mentén lineáris interpolációval határozza meg az ismeretlen pontok magasságát, és minden ponthoz a – metszet mentén való – lejtés értékét.

– A fentiek szerint a DEM minden pontjához több magasság- és lejtésérték keletkezik. Ezek közül azt a magasságértéket választja az algoritmus, amelyhez a legnagyobb lejtés tartozik.

Lokális maximumok és minimumok itt is csak az adott pontokban léphetnek fel (a hegycsúcsok tehát laposak maradnak). További hátrány, hogy szintvonalak megszakadása esetén erős torzulások léphetnek fel, márpedig a gyakorlatban nehéz garantálni a szakadásmentes szintvonalrajzot.

### 5.4.2. Variációs spline interpoláció

Alábbiakban a legkorszerűbbnek tekinthető interpolációs módszert ismertetjük. Az eljárás két fő fázisból áll:

1. *Előkészítés:* kezdeti DEM mátrixot (lásd ábra) hozunk létre (szkennelt szintvonalrajz esetén a vonalak vékonyításával, vektorizált szintvonalrajz esetén raszterizálással).

2. *Térbeli interpolációval* meghatározzuk a szintvonalak közé eső pixelek magasságértékét.

X	X	200	X	X	240	X	X	X	X
X	200	X	X	X	240	X	X	X	X
X	200	X	X	X	240	X	X	X	280
200	X	X	X	240	X	X	X	280	X
X	X	X	240	X	X	X	280	X	X
X	240	240	X	X	X	280	X	X	X
240	X	X	X	X	X	280	X	X	X
X	X	X	253	X	X	X	280	X	X
X	X	X	X	X	X	X	X	280	280

36. ábra. Kezdeti DEM mátrix részlete. X definiálatlan pontot, a 253 érték egyedi magasságpontot jelöl.

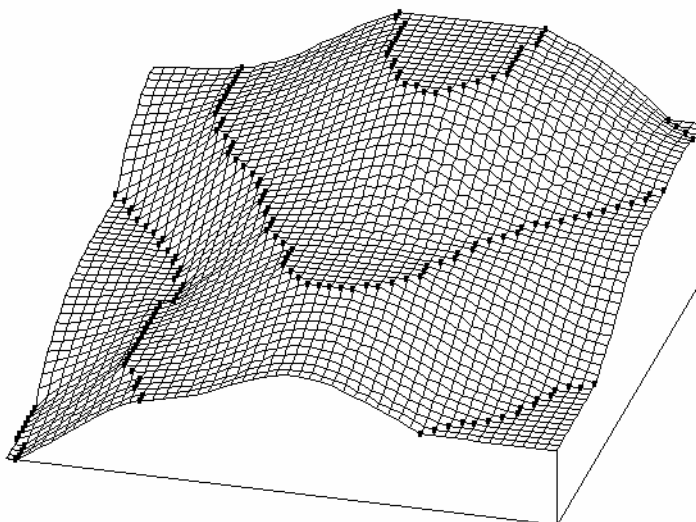
A térbeli interpoláció elvét először folytonos esetben tárgyaljuk.

Olyan  $f(x, y)$  függvényt keresünk, amely az adott  $(x_1, y_1), \dots, (x_m, y_m)$  pontokban a megfelelő  $d_1, \dots, d_m$  értékeket veszi fel, és "minimális energiájú" felületet képez. A felszín energiájának mérésére kétféle modell használatos:

– *Membrán modell* esetén az

$$E_f^1 = \iint (f_x^2 + f_y^2) dx dy \quad (1)$$

energiafüggvényt használjuk, ahol  $f_x$  az  $x$  szerinti parciális deriváltfüggvényt jelöli, hasonlóan  $f_y$ . A membrán modell bizonyos értelemben minimális felszínű felületet ad, így megengedi, hogy a szintvonalaknál törések lépjenek fel, a zárt szintvonallal határolt hegycsúcsok pedig laposak lesznek, amennyiben magassági pont nincs hozzájuk megadva (37. ábra).

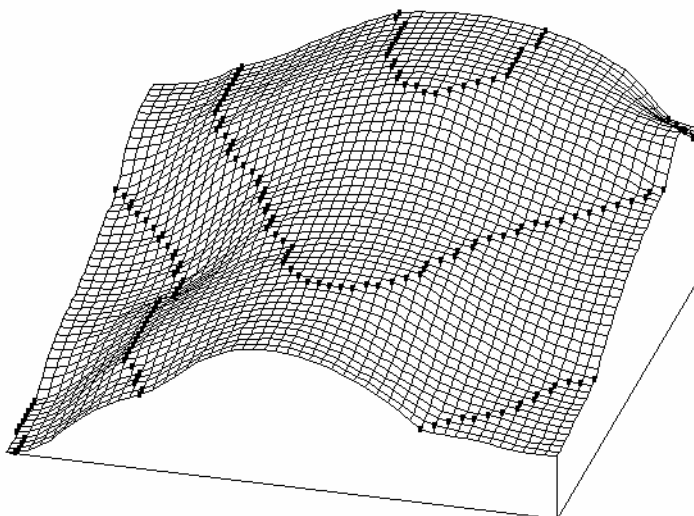


37. ábra. Membrán modell

– *vékonylemez modell (thin plate model)* esetén az

$$E_f^2 = \iint (f_{xx}^2 + 2f_{xy}^2 + f_{yy}^2) dx dy \quad (2)$$

energiafüggvényt használjuk, ahol  $f_{xx}$ ,  $f_{xy}$  és  $f_{yy}$  a megfelelő másodrendű parciális deriváltakat jelöli. Ez a modell nem viseli el az éles töréseket, ezért – szemben a membrán modellel – a szintvonalaknál sima átmenetet alkot, a hegycsúcsok feldomborodnak és a mélyedések besüllyednek (38. ábra). Ezért egyértelmű, hogy *terepmodellezésre a vékonylemez modell alkalmas*.



38. ábra. Vékonylemez modell

A fenti integrálok diszkrétizálásával megmutatható (lásd a Függelékben), hogy membrán modell előállításához a kezdeti DEM mátrixon ismételt konvolúciót kell végrehajtani a

$$\begin{array}{ccc} & 1/4 & \\ 1/4 & 0 & 1/4 \\ & 1/4 & \end{array}$$

maszkkal mindaddig, amíg a mátrix be nem konvergál. Vékonylemez modell előállításához szintén ismételt konvolúció végzendő a

$$\frac{1}{32} \begin{bmatrix} & & -1 & & \\ & -2 & 8 & -2 & \\ -1 & 8 & 12 & 8 & -1 \\ & -2 & 8 & -2 & \\ & & -1 & & \end{bmatrix}$$

maszkkal. Amíg azonban membrán modell esetén a konvergencia gyors (40 iteráció már elegendő), addig vékonylemez modellnél több ezer iteráció szükséges.

Az iteráció gyorsítására *multigrid technikát* alkalmaznak, amelynek lényege a következő: Először egy kicsinyített (durva felbontású) fedvényt készítünk a szintvonalpontok átlagolásával, és előbb erre iterálunk (ez gyors). A kapott eredményt kezdőértékként felhasználjuk az eredeti fedvényhez, amelyre ezután már csak keveset kell iterálni.

A kicsinyített fedvény készítése: ha  $n$ -szeres kicsinyítést végzünk, akkor az eredeti fedvényt  $n \times n$ -es négyzetekre osztjuk, és minden négyzetet egy pixellel helyettesítünk. Ha egy  $n \times n$ -es négyzet tartalmazott szintvonalpontot, akkor azok átlaga lesz az új pixel értéke, és ezt a pixelt az iteráció során majd fixen hagyjuk. Ha az  $n \times n$ -es négyzet nem tartalmazott

szintvonalpontot, akkor az új pixel értéke definiálatlan lesz, és értékét az iteráció során határozzuk meg.

A továbbiakban az egyszerűség kedvéért tegyük fel, hogy a kezdeti H mátrix  $2^n \times 2^n$  méretű. Ekkor n iterációs menetet végzünk, egyre finomodó rasztereken: az i-edik menetben egy  $H_i$   $2^i \times 2^i$ -es kicsinyített mátrixszal dolgozunk.

– Először a H mátrixból egy  $H_0$   $1 \times 1$ -es (azaz egyelemű) 'mátrixot' képezünk az összes H-beli szintvonalpont átlagolásával.

– Ezután a H mátrixból egy  $2 \times 2$ -es kicsinyített  $H_1$  mátrixot képezünk, amelynek definiálatlan elemei – ha vannak ilyenek –  $H_0$ -ból kapnak értéket. Ezután  $H_1$ -re végzünk iterációt, ennek eredménye a  $H_1'$  mátrix.

– Ezután egy  $4 \times 4$ -es  $H_2$  kicsinyített mátrixot képezünk, amelynek definiálatlan elemei a  $2 \times 2$ -es  $H_1'$  mátrix megfelelő elemeinek értékét kapják kezdőértékként.  $H_2$ -re is iterálunk, így áll elő  $H_2'$ .

– Az eljárást folytatva eljutunk a  $H_n=H$  mátrixhoz, amelyre iterálva a  $H'$  eredménymátrixot nyerjük.

Az eljárás előnye, hogy egy-egy iterációs menetben csak igen keveset (10-40 lépést) kell iterálni, mivel az előző grid fokozatból már majdnem jó kezdőértékekkel indulunk. A számításgény ezzel drasztikusan csökken (kicsinyített mátrixokra végzett iteráció eleve gyors).

A fenti eljárás akkor is használható, ha nem szintvonalas térképből, hanem egyedi magasságpont értékekből indulunk ki (32. ábra).

## 5.5. TIN előállítása DEM-ből

A cél az, hogy minél kevesebb háromszöggel minél pontosabban fedjük le a terepet. Feltételezzük, hogy ehhez egy elegendően sűrű (finom felbontású) DEM áll rendelkezésre. A TIN előállítása két lépésből áll:

- szögpontok kiválasztása
- háromszögek kijelölése (lásd 5.1. fejezet).

A szögpontok kiválasztására két algoritmust tárgyalunk.

### 5.5.1. Fowler-Little algoritmus

Először megjelöljük az alábbi speciális pontokat:

- csúcs: mind a 8 szomszédja kisebb
- gödör: mind a 8 szomszédja nagyobb
- nyeregpont: a kisebb és nagyobb szomszédok legalább négyszer váltanak a körbejárás során, például:

$$\begin{array}{ccc} + & + & - \\ - & & - \\ - & + & + \end{array} \quad \text{vagy} \quad \begin{array}{ccc} + & - & + \\ - & & - \\ + & - & + \end{array}$$

ahol '+' nagyobb, '-' kisebb szomszédot jelöl.

Ezután 2 x 2-es ablakokban vizsgáljuk a fennmaradó pontokat, egy pont 4 ilyen ablakban szerepel. Most az alábbi speciális pontokat jelöljük meg:

- gerincpont: egyik ablakban sem legkisebb.
- völgypont: egyik ablakban sem legnagyobb.

Ezután nyeregpontról indulva szomszédos gerincpontokon keresztül csúcsig, ill. szomszédos völgypontokon keresztül gödörig haladunk. A gerinc- és völgyvonalakat 1 pixel vastagságúra vékonyítjuk, és vektorizáljuk. A kapott csúcs-, gödör-, nyeregpontok ill. gerinc- és völgyvonalak töréspontjai lesznek a TIN szögpontjai.

Megjegyzendő, hogy a fenti algoritmust még jelentősen finomítani kell gyakorlati alkalmazhatóságához.

### 5.5.2. VIP algoritmus

VIP = Very Important Point: ilyen 'fontos pontokat' keresünk. (Az ArcInfo ezt az algoritmust használja.) Minden P ponthoz egy F fontossági számértéket számolunk a pont szomszédjaiból. A szomszédokat számokkal jelöljük:

8	1	2
7	P	3
6	5	4

Vezessünk egyenest a pont két szemben lévő szomszédján (például a 4-es és 8-as szomszédon) keresztül, és számoljuk ki ennek a – térbeli – egyenesnek a távolságát a P ponttól:  $\text{dist}(P, P_4P_8)$ . A négy ilyen távolság átlaga legyen a pont fontossági mértéke:

$$F_P = (\text{dist}(P, P_1P_5) + \text{dist}(P, P_2P_6) + \text{dist}(P, P_3P_7) + \text{dist}(P, P_4P_8)) / 4$$

Ezután a pontokat növekvő fontossági sorrendben töröljük a DEM-ből (először a legkevésbé fontosat), amíg a pontok száma adott mértékre csökken, vagy a fontosság adott mértékre nő.

## 6. ALKALMAZÁSOK

(Ez a fejezet még átdolgozás alatt áll)

### 6.1. Ingatlan nyilvántartás

*LIS* = Land Information System = ingatlan nyilvántartás.

Feladat: közhiteles alaptérkép vezetése, adatszolgáltatás.

Magyar Phare project (1998):

– Alapelv: térképek szkennelésével teljes, országos raszteres adatbázis előállítása, újrafelméréssel fokozatos lecserélése vektoros állományokra.

– Változásvezetés: a változások mindig új felméréssel keletkeznek, és vektoros foltokként kerülnek a rendszerbe.

– Ausztriában sikerrel alkalmazzák ezt a technológiát már hosszabb ideje.

### 6.2. Közmű nyilvántartás

*AM/FM* = Automated Mapping / Facility Management = közmű nyilvántartás.

*Közműtérkép komponensei:*

– alaptérkép (kataszteri térkép)

– közmű alaptérkép: a kataszteri térkép kiegészítése az ún. közterületi tartalommal (pl. fák, járdák, stb.)

– szakági adatok (vezetékek, szerelvények, stb.)

*Szakágak:*

– Vízvezetési alrendszer

– Csatornahálózati alrendszer

– Gázvezetési alrendszer

– Távhővezetési alrendszer

– Távközlési alrendszer

– Kábeltelevízió-hálózati alrendszer

– Elektromos hálózati alrendszer

– stb.

### 6.3. Önkormányzatok

Egy önkormányzati rendszer jellemzően az alábbi *alapadatokat* tartalmazza:

– *Földmérési alaptérkép.* Karbantartása a Földhivatal feladata.

– *Közmű alaptérkép.* Karbantartása az Önkormányzat feladata.

– *Szakági közmű térképek* és a kapcsolódó leíró adatok. Karbantartása az egyes közművállalatok adatszolgáltatása alapján történik, az Önkormányzat és a vállalatok közötti megállapodás szerint.

– *Alrendszerek* térképi és leíró adatai. Karbantartásuk az Önkormányzat feladata.

Az alrendszerek például a következők lehetnek:

- Területfelhasználási alrendszer (építési engedélyek kiadása)
- Területrendezési alrendszer (ÁRT, RRT)
- Szociális és családvédelmi alrendszer (bölcsőde, óvoda, stb.)
- Egészségügyi alrendszer
- Oktatási alrendszer
- Lakás és egyéb célú helyiségfelhasználási alrendszer (önkormányzati tulajdonú ingatlanok)
- Vállalkozói és kereskedelmi alrendszer (üzletek, stb.)
- Integrált közműnyilvántartás

A rendszer *szolgáltatásai*:

- Tervezési munkák támogatása: gyorsan generálhatók különböző variánsok, a tervezést befolyásoló hatások a térképen ábrázolva vizsgálhatók, elemezhetők.
- Komplex lekérdezési lehetőségek.
- Döntéstámogatás (például útburkolat bontás engedélyezése, iskolák bezárása, létesítése).

#### **6.4. Közlekedés, navigáció**

- Navigáció: gépkocsiban (vagy hajón, repülőgépen) elhelyezett GPS vagy giroszkóp, és számítógép a város digitális térképével. A képernyőn a vezető mindig látja a gépkocsi aktuális helyzetét a térképen.
- Járműpark irányítás: pl. szállítási vállalat járműveinek aktuális helyzetéről nyilvántartás, útvonal optimalizálás, stb.
- Bevetés irányítás. Például rendőrség, tűzoltóság, honvédség.
- Nagyvárosi forgalom irányítás. Utakon elhelyezett érzékelők ill. kamerák segítségével a forgalmi adatok a számítóközpontba kerülnek, amely digitális térképen megjeleníti azokat, előre jelzi a várható dugókat, és lehetővé teszi a beavatkozást (pl. közlekedési lámpák átállításával).

#### **6.5. Honvédelem**

Csapatmozgások koordinálása, légifelvételek kiértékelése, rakétarendszerek vezérlése, stb.

#### **6.6. Környezetvédelem**

Alapadatok lehetnek: talajtérkép, növénykultúra térkép, állatok élőhelyei, uralkodó szélirányok stb.

Feladatok: szennyeződések terjedésének modellezése, ipartelepek létesítéséhez hatástanulmányok, mezőgazdasági terméseredmény becslés stb.

#### **6.7. Meteorológia**

Műholdfelvételek elemzése alapján időjárás előrejelzés. Jellemzően raszteres feldolgozással történik.

## 6.8. Geomarketing

Alkalmazók: kereskedelmi vállalatok, bankok, biztosító társaságok.

Jellemző feladatok: hely kiválasztás üzletek, ATM-ek részére. Területfelosztás (pl. biztosítási tanácsadók részére).

Alapadatok: lakosság eloszlása, vásárlóerő, életmód (pl. gépkocsi használat), saját meglévő hálózat, versenytársak hálózata.

Példa: kockázatbecslés biztosító társaságnál: biztosított épületek környékén meghatározni az erdőtüzek valószínűségét. Adatok: a tűz terjedését befolyásoló tényezők (növényzet, terepviszonyok), tűzoltók általi megközelíthetőség. Ilyen az Insurance Service Office által fejlesztett FireLine rendszer.

## 6.9. Intézmény üzemeltetés

Sok dolgozóval és egy vagy több épületben nagyszámú helyiséggel rendelkező intézmény üzemeltetése (pl. egyetem).

Jellemző feladatok:

- helyiség foglaltság nyilvántartása és kihasználtság elemzése.
- közművek (vezetékrendszerek) nyilvántartása.
- karbantartási munkák tervezése.
- telefonszámok és helyiségek kapcsolata.

Alapadatok: épületek szintrajzai, szervezeti egységek és dolgozók adatai, helyiség foglaltsági adatok.



## 7. FEJLESZTŐ CÉGEK, SZOFTVEREK

(Ez a fejezet még átdolgozás alatt áll)

### 7.1. Általános jellemzők

A térinformatikai szoftverek általában *nyitott rendszerek*. Ez azt jelenti, hogy az alaprendszer a felhasználó által írt modulokkal bővíthető, sőt egyedi, testreszabott felhasználói alkalmazások fejleszthetők. A térinformatikai rendszerek általában kétféle fejlesztő környezetet biztosítanak:

- *Makrónyelv*. Felhasználható eszközök: a rendszer alap parancsai.
- *C nyelvi környezet*. Felhasználható eszközök: a rendszer belső adatstruktúrájának kezelését, menükezelését és grafikáját megvalósító C függvények. (Általában az alaprendszert is C-ben fejlesztik.)

A nyitott rendszer átmenet a kulcsrakész rendszer és a fejlesztő eszköz (pl. C fordító) között. Ha egy célalkalmazást teljesen saját fejlesztő eszközzel oldunk meg, a munka túlnyomó részét a menük, grafika és adatkezelés kialakítása adja, nyitott rendszereknél ez a munka jórészt megtakarítható.

Szerver-kliens architektúra: a legtöbb GIS szoftver két változatban kapható:

- teljes értékű (szerver) változat, amely az adatok lekérdezését és módosítását egyaránt lehetővé teszi,
- csak lekérdező alkalmas (kliens) változat, amely jóval olcsóbb.

A felhasználók rendszerint egy szerver és több kliens változatot vásárolnak.

Fejlődési tendenciák:

- Internetes alkalmazások támogatása, GIS adatok Web-en való publikálása.
- Grafikus és leíró adatok közös adatbázisban, objektum-relációs megközelítés.

### 7.2. Vektoros rendszerek

#### **AutoCAD**

*Fejlesztő cég:* Autodesk, USA.

Magyarországon széles körben használt mérnöki tervező rendszer. Régebbi verziói a térinformatikát nem támogatták, a 12-es verzió már lehetővé teszi a rajz-adatbázis kapcsolatot. Vektoros rajzfájl-formátuma a DWG (a *drawing* szóból), amelyet számos más rendszer is támogat.

*Fejlesztő környezet:*

- AutoLisp makrónyelv,
- C nyelv.

Az Autodesk az 1990-es években erőteljes térinformatikai fejlesztésbe fogott, 1996-ban megjelent GIS termékei:

- *AutoCAD Map*: Az AutoCAD bővítése térinformatikai eszközökkel.

- *AutoDesk World*: kevés rajzszerkesztő funkcióval rendelkező, inkább lekérdező térinformatikai rendszer, Microsoft-Office felhasználói felülettel.
- *AutoDesk MapGuide*: térinformatikai Internet támogatás.

Honlap: [www.autodesk.com](http://www.autodesk.com)

## **Mapinfo**

*Fejlesztő cég*: Mapping Information Systems Corporation, USA.

Asztali térképező rendszer PC-re, kisebb térinformatikai alkalmazásokra javasolt. Például kistéleplési önkormányzatok sikerrel használják.

Spagetti-adatmodellt használ.

Fejlesztő környezet: MapBasic nyelv (SQL-t tartalmaz).

Honlap: [www.mapinfo.com](http://www.mapinfo.com)

## **MicroStation**

*Fejlesztő cég*: Bentley Systems. 1984-ben alapítva. 1995-től saját terjesztői hálózat (előtte az Intergraph termékek részeként jelent meg a MicroStation).

*Változatok (1990 óta)*: 4. verzió, 5. verzió, MicroStation 95, MicroStation SE, MicroStation J, (J a Java nyelvre utal), MicroStation V8i.

Általános célú CAD rendszer, erős térinformatikai támogatással. Magyarországon széles körben használják. Eredetileg VAX gépekre készült (ennek a változatnak a neve: IGDS = Interactive Graphics Design Software), ennek készítették el a PC változatát. Jellemzők:

- teljesen kompatibilis PC és VAX (UNIX) változat
- saját grafikus adatformátum: DGN (bináris felépítése dokumentált).
- külső adatbázis kapcsolat: többek között Xbase, Oracle (komolyabb alkalmazásokhoz az utóbbi).

*Fejlesztő környezet*:

- makrónyelv: régebben UCM, később MicroStation Basic (Visual Basic-hez hasonló).
- fejlesztő nyelv: MDL (C nyelvi környezet): nem gépi kódra, hanem közbülső kódra fordít, így teljes kompatibilitást biztosít a különféle hardver platformok között.

*Kapcsolódó szoftverek*:

1. GeoGraphics: térinformatikai modul. A DGN adatstruktúrán topológiát hoz létre. Ez lehetővé teszi a következőket:

- Digitalizálási hibák automatikus megkeresése és javítása.
- Térbeli elemzések végzése.

2. Descartes: raszteres modul. Raszterképek szerkesztését, transzformációját, (fél)automatikus vektorizálását teszi lehetővé.

3. GeoOutlook: csak megjelenítő szoftver (rajz nem módosítható, kliens szoftver), MicroStation-nél sokkal olcsóbb.

Honlap: [www.bentley.com](http://www.bentley.com)

### **ArcInfo**

*Fejlesztő cég:* ESRI = Environmental Systems Research Institute, USA.

Az ESRI 1969-ben alakult, a 70-es években lassan fejlődött. Az 1980-as évek elején jelent meg az ArcInfo.

Az ArcInfo volt az első széles körben elterjedt, kimondottan térinformatikai célú (nem CAD) szoftver. Magyarországon kevésbé használatos, de a világpiacon az első helyen áll.

Az ún. *georelációs adatmodellt* használja, amelynek lényege, hogy a grafikus adatokat topológikus struktúrában tárolja, és a leíró adatokkal rajzelem-attribútum-táblák segítségével kapcsolja össze.

Saját relációs adatbáziskezelővel rendelkezik, de külső adatbázis (pl. Oracle) kapcsolását is támogatja.

Fejlesztő eszköz: AML = Arc Macro Language.

*ArcView:* asztali térképező szoftver, az ArcInfo-hoz képest csökkentett funkcionalitással. A hangsúly a megjelenítésen és az adatintegráción van. Fejlesztő nyelve az *Avenue*.

*ArcGIS:* az ArcInfo újabb változata. Adatmodellje a *geoadatbázis*, amely a térbeli és leíró adatokat egyaránt objektum-relációs adatbázisban tárolja, és lazít az ArcInfo szigorú topológikus szemléletén.

Honlap: [www.esri.com](http://www.esri.com)

## **7.3. Raszteres rendszerek**

### **ERDAS Imagine**

*Fejlesztő cég:* ERDAS Inc. (USA)

A legelterjedtebb raszteres rendszer. Kezdetben műholdképek feldolgozására készült, később bővítették térinformatikai funkciókkal. Az ArcInfo-val közvetlen adatkommunikációra képes. Korlátozottan vektoros adatokat is kezel.

Fejlesztő nyelv: EML = ERDAS Macro Language.

Honlap: [www.erdas.com](http://www.erdas.com)

### **Idrisi**

*Fejlesztő:* Clark Labs, USA

Elsősorban oktatási célú rendszer.

Honlap: [www.clarklabs.org](http://www.clarklabs.org)

## 7.4. Hazai fejlesztések

**ITR** (= Interaktív Térképszerkesztő Rendszer)

*Fejlesztő cég:* FÖMI, majd DIGICART Kft., Magyarország.

Térképszerkesztő CAD rendszer (nem térinformatika). 1988 óta sok helyen használják Magyarországon, a hazai digitális térképek jelentős része – első változatban – ezzel készült.

Honlap: [www.digicart.hu](http://www.digicart.hu)

**TopoLogic**

*Fejlesztő cég:* Geometria Térinformatikai Rendszerház Kft. (Magyarország).

Komplex térinformatikai rendszer, első változata 1989-ben készült el, ma már nem használatos. Vektor, raszter, DTM modulokat tartalmaz. Grafikus adatbázisa négyesfa alapú (gyors keresést biztosít).

Fejlesztő környezet: makrónyelv, függvénykönyvtár

Adatbázis kapcsolat: Oracle, Ingres

Honlap: [www.geometria.hu](http://www.geometria.hu)

**GreenLine, Kolibri**

*Fejlesztő cég:* Geoview Systems Kft és InterMap Kft (Magyarország.)

A GreenLine Unix ill. Windows alapú, teljesen objektum-orientált térinformatikai rendszer, az 1990-es években használták. Saját objektum-orientált adatbázis-kezelővel rendelkezik. Fejlesztő környezete az L nyelvre épül, amely a rendszer saját objektum-orientált nyelve, algoritmikus eszközöket is tartalmaz.

A Kolibri kezdetben a GreenLine kliens moduljaként indult, majd önálló asztali térinformatikai rendszerként terjedt el, ma is használatos.

Honlap: [www.geoview.hu](http://www.geoview.hu) (GreenLine), [www.intermap.hu](http://www.intermap.hu) (Kolibri)

## A. FÜGGELÉK: MicroStation

A MicroStation rendszer (röviden MS) fontosabb verziói: 4, 5, 95, SE, J, V8. Itt az SE verziót tárgyaljuk.

### A.1. Alapok

*Elnevezések, jelölések:*

- *Menü/Almenü:* menüválasztás jelölése
- *Paletta:* funkciógombokból álló eszköztár. Adott paletta (i,j) gombján az i-edik sor j-edik gombját értjük (bal felső sarok: (1,1)). Dokkolható paletták, *Tools* menüből nyithatók ki.
- *Alpaletta:* paletta-funkciógomb tartós lenyomására nyílik ki, egérrel vonszolva a képernyőn külön is elhelyezhető. Jelölés: (i,j)/k az (i,j) funkciógombhoz tartozó alpaletta k-adik gombja.
- *Egér gombok:* data button (bal), tentative button (középső, vagy két szélső együtt), reset button (jobb).
- *View:* a rajz egy nézete, amely külön ablakban jelenik meg. Egyszerre több nézet használható.
- *View control:* view keretén lévő gombokkal végezhető. Funkciók sorrendben: frissítés, nagyítás, kicsinyítés, nagyítás ablakkal, teljes rajz megjelenítés, stb. Scrollozás: Shift+egér.
- *Adatbeviteli mezők:* vigyázat, sokszor csak mezőváltás (Tab vagy Enter) után érvényes a begépeltek érték.
- *Fence:* kijelölő poligon.

*Settings/Design file:* összes beállítás itt megnézhető és módosítható, például: koordináta rendszer beállítások, grid beállítás (lásd alább).

*File/Save settings:* beállítások mentése DGN fájlba.

*Workspace/Configuration:* konfigurációs változók beállítása:

- Kategória választás (pl. database) után a megfelelő változócsoporthoz állítható.
- Mentés \*.ucf fájlra. Alapértelmezés: config/user/default.ucf, de az ablak *File/Save as* menüpontja segítségével más névre is menthető.
- Rendszerüzenetek* az alsó sorban.

### Adattárolás

Egy rajz egy \*.DGN fájlba kerül tárolásra, amely rajzelemek sorozatát tartalmazza. A MicroStation minden változást azonnal fájlba ír, mentés nem szükséges. Minden változást Edit/Undo-val vissza lehet állítani – mindaddig, amíg a *File/Compress design* parancsot ki nem adjuk.

*Seed file:* rajzelemeket nem tartalmaz, csak a szükséges alapbeállításokat. Pl. új 2D rajz készítésekor a SEED2D.DGN fájl másolatából indulunk ki.

## Koordinátarendszer, grid

A MicroStation 32-bites egész számokkal dolgozik. Egy egység = UOR = Unit Of Resolution (= PU = Positional Unit). Teljes rajzterület:  $2^{32} \times 2^{32}$  négyzet, vagyis kb. 4 milliárd x 4 milliárd.

*Origó:* alapértelmezés szerint a rajzterület közepén, de máshová is tehető.

*Koordináta tengelyek:* a matematikai X,Y konvenció szerint.

*Working units:* a rajzoláskor használt mértékegységek, beállításuk a *Settings/Design file/Working units* menüpontban. Kétféle van, a Master-unit (MU) és a Sub-unit (SU):

$$\text{MU} = n_1 * \text{SU},$$

$$\text{SU} = n_2 * \text{PU},$$

ahol  $n_2$  értéke a Working resolution. Az  $n_1$  és  $n_2$  váltószámok tetszőlegesen adhatók meg. Koordináták megadására a rendszer az MU:SU formát használja (SU tizedesjegyekkel).

*Példa:* térkép esetén MU = km, SU = m. Ha Working-resolution = 1000, akkor mm pontossággal számolhatunk, és a max. ábrázolható terület 4000 x 4000 km.

**Grid:** A rajzon való tájékozódást segítő pontrács. Beállítása a *Settings/Design file/Grid* menüpontban:

- Grid Master: két gridpont távolsága adandó meg MU egységben.
- Grid Reference: egy n egész szám adható meg, minden n-edik grid pontot kiemelten jelenít meg.
- Grid lock: bekapcsolása esetén az egér kurzor mindig grid pontra ugrik, tehát csak 'grid koordinátájú' pont vihető be.

*Settings/View attributes:* grid megjelenítés ki-bekapcsolása.

Az érvényes nagyítástól függően

- minden grid pont megjelenik,
- csak a kiemelt pontok jelennek meg, mivel az alappontok túl közel lennének egymáshoz.
- a kiemelt pontok sem jelennek meg, mivel túl közel lennének egymáshoz.

**AccuDraw:** Primary eszköztáron kalapács szimbólum. Kétféle üzemmódja van, közöttük Space leütésével lehet váltani:

- derékszögű koordináták (X,Y).
- polárkoordináták (distance, angle).

*Funkciói:*

- tájoló: adatpont bevitelkor kis "iránytű" jelenik meg.
- koordináta lekérdezés: az egér kurzor aktuális pozícióját mutatja (első adatpont beviteléig abszolút, utána relatív koordinátákat).
- koordináta begépelés

*Settings/Design file/Coordinate readout:* számábrázolás beállítása

*Példa beállításra:*

- Working units: MU=m, SU=cm, 1 cm = 1000 PU
- Grid: Master=0.1, Ref=10. Ekkor gridtávolság 10 cm, ref. gridtávolság = 1 m.
- Az eredmény kipróbálható AccuDraw bekapcsolásával.

## Parancsvezérlés

*Utilities/Key-in*: parancs begépelése. Az ablak alsó szélét feltolva célszerű egysorosra zsugorítani és felülre dökkolni. Fel-le nyíl billentyűkkel visszahozhatók a korábban begépelte parancsok. Minden rajzi műveletnek megfelel egy parancs, például

PLACE LINE : vonalrajzoló funkcióba belépés

XY=x,y : abszolút koordináta megadás

DL=x,y : az utolsó bevitt pontra relatív koordináta megadás

RESET : funkcióból kilépés

## Rétegek

*Settings/Level/Display*: rétegek kezelése. Kattintás: megjelenítés ki-bekapcsolása. Kettős kattintás: aktív réteg váltás.

*Settings/Level/Manager/Names*: hierarchikus réteg struktúra kezelése. A megfelelő view-t ki kell választani!

*Settings/Level/Symbology*: a beállítás ideiglenesen felülírja a rajzelemek saját attribútumait, feltéve, hogy *Settings/View attributes/Level symb.* be van kapcsolva.

## DGN fájl felépítése

*Rajzelem típusok* (C nyelvű struktúrákként dokumentáltak):

Type 1: cell library header

Type 2: cell header

Type 3: line (egyenesszakasz)

Type 4: line string (töröttvonal)

Type 5: group data

Type 6: shape (poligon)

Type 7: text node

Type 11: curve

Type 12: complex chain

Type 14: complex shape

Type 15: ellipse

Type 16: arc

Type 17: text (felirat)

Type 18: 3D surface header, type 19: 3D solid header

Type 21, 24...28: B-spline

Type 22: point string

Type 23: cone element

Type 66: MicroStation application element

Type 87: raster header, type 88: raster data

## A.2. Rajzszerkesztés

*Primary toolbar:* aktuális szín, réteg, vonaltípus, vonalvastagság beállítása.

*Element/Information (Ctrl+i):* A kijelölt rajzelem összes attribútuma lekérdezhető ill. módosítható. (Elem típus megnevezése a fejlécben)

*Element/Text:* felirat attribútumok

### A Main eszköztár

A kiválasztott rajzoló funkcióhoz automatikusan kinyílik a megfelelő paraméter beállító ablak.

*(1,1) gomb* ('nyíl'): rajzelem kijelölés. Lenyomott egér gombbal téglalap rajzolható, így a bele eső valamennyi elem kijelölhető. Kijelölt elem mozgatása: a kijelölés után folyamatosan lenyomva tartott egér gombbal. Több elem kijelölése: Ctrl+egér.

*(1,2) gomb:* fence műveletek. */1 gomb:* fence rajzolása. A paraméter beállító ablakban lehet megadni, hogy pl. téglalap (block) vagy poligon (shape) módon lehet fence-t rajzolni, illetve a fence módot. */2 gomb:* fence módosítása, */3 gomb:* fence tartalmának kezelése (a paraméter ablak szerint másolás, mozgatás, stb. Fence mód beállítása: Inside: teljesen belül lévő, Overlap: részben belül lévő, Clip: metszés, Void\*: komplementerre ugyanez), */4 gomb:* tartalom törlése. Fence törlése a */1 gomb* újbóli lenyomásával.

*(2,2) gomb:* */1:* töröttvonal (smart line), */2:* egyenesszakasz, */4:* folytonos vonal rajzolás

*(3,2) gomb:* poligon rajzolás. Alpaletta 1. gomb: téglalap, 2. gomb: tetsz. poligon.

*(5,2) gomb* ('A'): szöveg bevitele. (Paraméterek beállítása *Element/Text* segítségével is.)  
Ékezetes fontok: 105-től kezdődően.

*(5,2)/3 gomb:* szöveg módosítása (2 kattintás a rajzelemre).

*(5,2)/6 gomb:* szöveg attribútumainak módosítása (2 kattintás a rajzelemre).

*(5,2)/7 gomb:* text node bevitele (csak egér bal gomb).

*(8,2) gomb:* kijelölt rajzelemek másolása (*/1 gomb*) és mozgatása (*/2 gomb*).

*(9,2) gomb:* rajzelem módosítása. Például kattintsunk a módosítani kívánt csomópontra, és mozgassuk az egérrel.

*Bal alsó gomb* ('áthúzott téglalap'): kijelölt rajzelem(ek) törlése. Ha nincs kijelölve rajzelem, akkor első kattintás kijelölés, második törlés.

### Snapping

Snappelés a tentatív-gombbal (bal+jobb) bevitt próbapont segítségével történik. Ha a próbapont 'megfogja' a kiszemelt rajzelemet, akkor az kiszürkül (egyébként újra kell próbálkozni).

*Settings/Snaps/Button bar:*

1. gomb: a rajzelemnek a kattintás helyéhez legközelebb eső pontjához,

2. gomb: keypoint-hoz (pl. egyenesszakasz végpontjához ill. felezőpontjához).

Az alapértelmezés kettős kattintással változtatható meg.

*Settings/Locks/Full/Snap, vagy Settings/Design file/Snap:*

– Snap lock ki/bekapcsolandó.



– Divisor: értéke adja meg, hogy keypoint snap esetén egy egyenesszakaszt hány részre osztunk. 1 esetén csak a két végpont az osztópont, 2 esetén a felezőpont is, stb.

## Cell-ek

*Element/Cell/File/New*: új cell-könyvtár létrehozása

*Element/Cell/File/Attach*: cell-könyvtár csatolása a rajzhoz

*Cell behívása*:

- cell kiválasztása, utána *Placement* gombot megnyomni.
- *Main (6,2)/(1,1)*: cell elhelyezése. Méret és elforgatási szög a paraméter beállító ablakban megadható.

*Új cell felvétele a cell-könyvtárba*:

- cell megrajzolása, majd kijelölése.
- *Main(6,2)/(1,4)*: illesztőpont megadása.
- *Element/Cell/Create*: cell nevének megadása után a cell felkerül a listára.

## Patterning

Kitöltő minták. Vigyázat, a kitöltő minta sok kis rajzelemet vihet fel a rajzra!

*Main(3,1)*: patterning paletta kinyitása

- 1. gomb: sraffozás. Vonaltávolság és szög után egérrel kijelöljük a megf. zárt alakzatot (kiszürkül), majd újabb kattintásra besraffozza.
- 2. gomb: kereszt-sraffozás, hasonlóan.
- 3. gomb: 2D pattern cell alapján. Például mozaiklap minta rajzolása: *geompa.cel* megnyitása (*ustation/wsmod/default/cell* könyvtárban), *geom12* kiválasztása. *Main(3,2)* gombbal zárt alakzat rajzolása. *Main(3,1)/3*: *Pattern cell=geom12* (a Cell ablak *Pattern* gombjára kattintva beírja), *Scale=...*, 2x bal gomb a zárt alakzatra.
- 4. gomb: 1D pattern cell alapján. Például vasútisín rajzolása: *linepa.cel* megnyitása, *railrd* kiválasztása. *Main(2,2)* gombbal *linestring* rajzolása (*rounding radius* beállítás). *Main(3,1)/4*: *Pattern cell = railrd*, *Scale = ...*, 2x bal gomb a *linestring*-re.

### A.3. Adatbázis kapcsolat

A MicroStation rendszer – többek között – Xbase, ODBC és Oracle adatbázis interface-t támogat. Az adatbázis lekérdezése MicroStation-ból minden esetben SQL nyelvi alapon történik. A különböző DBMS-ekkel való kapcsolat alapvetően különböző:

*Oracle:* MS az Oracle kernel jelenlétét feltételezi, SQL interface-en keresztül kommunikál vele. Az SQL parancsokat az Oracle értékeli ki és adja vissza az eredményt MS-nek. Minden SQL parancs megengedett, amelyet az Oracle értelmezni tud.

```
MS <---> Oracle kernel <---> Adatbázis
      SQL
```

*Xbase:* MS nem igényli az Xbase szoftver jelenlétét, hanem a DBF fájlok és indexfájlokat saját SQL interpreterével kezeli. Csak olyan SQL parancsok megengedettek, amelyeket MS értelmezni tud.

```
MS <---> MS SQL interpreter <---> Adatbázis
      SQL
```

A különböző Xbase rendszerek eltérő indexformátumot használnak, ezért az adatbázis-interface is részben rendszerspecifikus.

*ODBC interface-en keresztül Access adatbázis elérése:*

```
MS <-> MS ODBC-driver <-> Access ODBC-driver <-> MDB fájl
      SQL                ODBC
```

#### Xbase kapcsolat inicializálása

*Adatbázis oldalról:*

1. Az adatbázis fájlok felvitele egy könyvtárba (a továbbiakban DB könyvtár).
2. Minden kapcsolandó táblát egy NUM(10) típusú MSLINK mezővel kell bővíteni, és eszerint indexelni.

3. Az xstation/database könyvtárból a megfelelő alkönyvtárat választjuk: xbase (dBase4), xbasendx (dBase3), xbasefox (FoxPro), a továbbiakban DBUTIL könyvtár. (Az egyes Xbase rendszerek eltérő indexfájl-formátumai miatt szükséges a megkülönböztetés: dBase4 \*.MDX, dBase3 \*.NDX, FoxPro \*.CDX.)

4. A DBUTIL könyvtárban található három fájlt átmásoljuk a DB könyvtárba. Ezen fájlok egy segédprogramot tartalmaznak, a továbbiakban DATADICT segédprogram.

5. A DBUTIL/examples/gis/mscatlog.dbf katalógus-mintafájlról másolatot készítünk a DB könyvtárba, és a tartalmát aktualizáljuk a saját adatbázisunknak megfelelően (a továbbiakban MSCATALOG tábla). Az MSCATALOG tábla annyi sort tartalmazzon, ahány adattáblát akarunk kapcsolni a rajzhoz. Első közelítésben csak a Tablename (kapcsolandó tábla neve) és Entitynum (a tábla MS által használt azonosítószáma) oszlopokat kell kitölteni. Entitynum tetszőleges pozitív szám lehet, de minden táblára más legyen.

6. A DB-könyvtárban egy Data dictionary-t kell létrehozni (ez az sqlindex.dbf, sqltable.dbf, sqlcolumn.dbf táblákból áll) a DATADICT program futtatásával:

```
datadict -init                kezdeti Data dictionary létrehozás
datadict -ta mscatlog mscatalog katalógus tábla felvétele
```

datadict -ta dbffájlnev táblanév      tábla felvétele (minden táblára végrehajtandó)  
 datadict -ia indexfájlnev indexnev      tábla felvétele (minden táblára végrehajtandó)

|| *Egyszerűsítés:* Ha most nem hozzuk létre az MSCATALOG táblát, vagyis az 5. lépést elhagyjuk, akkor a „datadict -ta mscatlog mscatalog” sem kell, és később egy hibaüzenet kíséretében a rendszer automatikusan létrehozza az MSCATALOG-ot, lásd alább.

#### *MicroStation oldalról:*

1. Konfigurációs változók beállítása és mentése (Workspace/Configuration, Database kategória):

– Database Files: MS\_DBASE = DB-alapkönyvtár útvonala, végén "\" (például: c:\ustation\database\proba\). Ez célszerűen megegyezhet a DB könyvtárral. (Több adatbázis esetén a DB-alapkönyvtárból nyílnak az egyes konkrét DB könyvtárak, de a \*.FMT formátumfájlok ebben az esetben is a DB-alapkönyvtárban kell, hogy legyenek!)

– Server Loader: MS\_SERVER = dbload (dBase4), db3lddm.ma (dBase3), foxlddm.ma (FoxPro).

– Database Linkages: MS\_LINKTYPE = XBASE.

– A Database kategórián kívül: MS\_DGNAPPS = server

|| *Egyszerűsítés:* a konfigurációs változók beállítása helyett Settings/Database/Connect: Database server = Foxpro, Connect stringet üresen hagyjuk. Ekkor csak MS\_DBASE beállítása szükséges, a többi a rendszer automatikusan állítja.

2. MS újraindítása (adatbázis-szerver most már aktív).

3. Kapcsolódás az adatbázishoz (connect). Key-in: DB=adatbázis könyvtára (a DB-alapkönyvtár (MS\_DBASE változó) értékére relatív útvonallal, de \ főkönyvtárra ugrás helyett ..\..\... alkalmazandó). Ha a DB-alapkönyvtár és a DB-könyvtár megegyezik, akkor DB=. írandó.

|| Ha még nem hoztuk létre az MSCATALOG táblát, akkor hibajelzést kapunk, és Settings/Database/Setup segítségével létrehozhatjuk. Tartalmát a: Table/Add... menüpont segítségével adhatjuk meg.

## **ODBC kapcsolat inicializálása**

A beállításokat Access esetén mutatjuk be.

#### *Adatbázis oldalról:*

1. Létrehozunk egy Access adatbázist.

2. Minden kapcsolandó táblát egy „hosszú egész” típusú MSLINK mezővel kell bővíteni.

3 Létrehozunk egy MSCATALOG (tablename char(32), entitynum hosszú\_égesz, screenform char(64), reporttable char(64), sqlreview char(240), fencefilter char(240), dastable char(32), formtable char(64)) táblát. Az MSCATALOG tábla annyi sort tartalmazzon, ahány adattáblát akarunk kapcsolni a rajzhoz. Első közelítésben csak a Tablename (kapcsolandó tábla neve) és Entitynum (a tábla MS által használt azonosítószáma) oszlopokat kell kitölteni. Entitynum tetszőleges pozitív szám lehet, de minden táblára más legyen.

|| *Egyszerűsítés:* Ha most nem hozzuk létre az MSCATALOG táblát, később egy hibaüzenet kíséretében a rendszer automatikusan létrehozza, lásd alább.

4. A MicroStation nem szabályos telepítése esetén lehet szükség a következőkre: Az `ustation/database/odbc` könyvtárban lévő fájlokat az `ustation` könyvtárba másoljuk, és szükség esetén a `qelibnt.dll` fájlt átnevezzük `qelib.dll`-re.

*Windows oldalról:* ODBC adatforrás beállítása:

1. Vezérlőpult, Felügyeleti eszközök, ODBC adatforrások
2. Hozzáadás, "Microsoft Access Driver (\*.mdb)" kiválasztása, Befejezés gomb
3. Az "Adatforrás neve" mezőbe azt a nevet kell írni, amilyen néven el akarjuk majd érni az adatbázist a későbbiek során, Kiválasztás gomb, A csatolni kívánt adatbázis kiválasztása, OK gomb (ahányszor csak lehet).

*MicroStation oldalról:*

1. Konfigurációs változók beállítása és mentése (Workspace/Configuration, Database kategória):

- Server Loader: MS\_SERVER = `odblddlm.ma`.
- Database Linkages: MS\_LINKTYPE = ODBC.
- A Database kategórián kívül: MS\_DGNAPPS = server
- 2. MS újraindítása (adatbázis-szerver most már aktív).

3. Kapcsolódás az adatbázishoz (connect). Key-in: DB=az ODBC adatforráshoz beállított név.

*Egyszerűsítés:* a konfigurációs változók beállítása helyett Settings/Database/Connect: Database server = ODBC, Connect string = az ODBC adatforráshoz beállított név. Ha még nem hoztuk létre az MSCATALOG táblát, akkor hibajelzést kapunk, és Settings/Database/Setup segítségével létrehozhatjuk. Tartalmát a: Table/Add... menüpont segítségével adhatjuk meg.

## Adatbázis kapcsolat kezelése

*SQL lekérdezés.* Key-in: SQL. Az SQL-ablak megjelenik, az MSCATALOG-ba felvett táblák lekérdezése lehetséges. Lekérdezés végrehajtása a *Submit* gombbal, rekordokon léptetés *Next* gombbal.

Megjegyzés: SQL parancsot a rendszer csak akkor ismeri, ha az adatbázis szerver aktív. (Ehhez a Database kategóriában lévő konfigurációs változókat be kell állítani, és MS-t újraindítani.)

*Aktív entitás (AE):* egysoros segédtábla neve, amely SQL Select utasítással lekérdezhető. Tartalma az adatbázis aktuális rekordja is lehet. Beállításának egy lehetséges módja: Key-in: FIND SELECT \* FROM tábla WHERE feltétel. A feltételnek megfelelő rekord lesz az AE.

*Settings/Database/Dialog:* adatbázis dialógusablak megnyitása.

– Linkage mode = New: az AE prototípus rekordot jelent, amely új rekordként kerül felvételre a táblába.

– Linkage mode = Duplicate: az AE a tábla egy rekordját jelenti. Ha a rajzot már létező adatbázissal akarjuk összekapcsolni, akkor ez a beállítás kell.

– Forms = None: SQL ablakban jelenik meg a lekérdezés eredménye.

– Forms = Text screen: Xbase FMT-fájl (lásd alább) szerinti ablakban jelenik meg a lekérdezés eredménye, a rekord módosítható.

Ha a beállításokat meg kívánjuk őrizni, *File/Save settings* adandó ki.

*Tools/Database*: adatbázis eszköztár kinyitása.

(1) *gomb*: Attach active entity: rajzelemhez AE hozzárendelése, a rajzelemre történő kattintással (első kattintás: rajzelem kiválasztás, második kattintás: kapcsolat létrehozása.). Hatására egy (Entitynum, Mslink) bejegyzés kerül a rajzelemhez, ahol Entitynum a kapcsolt tábla MSCATALOG-beli azonosítószáma, Mslink pedig a kapcsolt rekord MSLINK értéke.

(2) *gomb*: Show linkage mode: linkage mode kiírása az alsó sorban.

(3) *gomb*: Show active entity: AE megjelenítése az SQL ablakban.

(4) *gomb*: Define active entity graphically: rajzelemhez kapcsolt rekord lesz az AE (a rajzelemre történő két kattintással). Ha Settings/Database/Confirm rows = ON, akkor AE megjelenik az SQL ablakban.

(5) *gomb*: Review database attributes of element: rajzelemhez kapcsolt rekord megjelenítése SQL ablakban, vagy a formátumfájlban megadott formátum szerint (két kattintás).

(6) *gomb*: Detach database linkage: rajzelemről rekord leválasztása (két kattintás).

(7) *gomb*: Attach Displayable Attributes: DAS-link text node-hoz (lásd alább).

(8) *gomb*: Load displayable attributes: DAS-frissítés (lásd alább).

(9) *gomb*: Generate report table: egy fence által tartalmazott rajzelemekhez kapcsolt rekordokból képez riport táblát. Akkor működik, ha a rajzhoz kapcsolt táblához az MSCATALOG tábla Reporttable oszlopában egy tetszőleges táblanevet adunk meg, ez lesz a riport tábla. Ezt a táblát automatikusan létrehozza a rajzhoz kapcsolt tábla struktúrájával, ha a fence területére kattintunk.

Megjegyzés: az MSCATALOG tábla tartalma MS-ből is módosítható a SET DATABASE parancs segítségével.

## **Űrlap (Form) használata**

Xbase kapcsolat esetén a kapcsolandó táblákhoz formátumfájl (\*.FMT) lehet készíteni, amely az MS-ből való lekérdezéskor megjelenítendő attribútumok formátumát adja meg.

*Key-in*: SET DATABASE: Screen form = formátumfájl neve.

*Settings/Database/Dialog* ablakban: Forms = Text screen: beállítás.

Ezután a Database(5) gombbal történő lekérdezés eredménye a formátumfájl szerinti űrlapban jelenik meg, és az egyes mezők módosíthatók is.

Xbase és ODBC esetén egyaránt használható: *key-in*: VSQL REVIEW, ezután a rajzelemre kattintva szabvány MS-űrlapban jelenik meg (és módosítható) a kapcsolt rekord. Az űrlap Settings/Database/Query builder/Settings menüben testreszabható.

## **Displayable attributes (DAS)**

*Text node*: olyan text rajzelem, amelynek felirata az adatbázisból kerül frissítésre, a rajzra helyezésekor érvényes text beállításoknak megfelelően (displayable attribute).

Egy speciális táblát kell létrehozni (a továbbiakban DASTABLE), oszlopai:

– dastype num(3): DAS-típuszám. Megegyezik a Settings/Database/Dialog/DAS type aktuális értékével. Akkor van jelentősége, ha egy tábla különböző oszlopait kívánjuk megjeleníteni az egyes text node-okban.

– formname char(240): Xbase formátumfájl neve, amely a megjelenítendő attribútumok formátumát adja meg. Csak akkor kell megadni, ha sqldas értékét nem adjuk meg.

– sqldas char(240): SQL-select utasítás, amely oszlopokat választ ki (elhagyható, ha formname-et megadjuk).

Ha formname és sqldas egyaránt NULL vagy hibás, akkor ez az összes attribútum kiválasztását jelenti, vagyis "select \* from tábla" megadásával egyenértékű.

DASTABLE nevét fel kell venni MSCATALOG Dastable oszlopába (SET DATABASE) és Xbase esetén a Data Dictionary-be (DATADICT).

*DAS-kapcsolat létrehozása:*

– Felveszünk egy text node-ot: Main(5,2)/7.

– A megfelelő rajzelemet kapcsoljuk egy rekordhoz Database(1)-gyel (find select...). A rajzelem speciális esetként lehet a text node is.

– Database(7) (DAS-típusszámot itt lehet megadni): kapcsolt rajzelem kijelölése 2 x bal gomb (ezzel kiválasztódik a kapcsolt rekord), utána text node kijelölése 2 x bal gomb (ezzel a text node-hoz hozzárendelődik a rekord, de text még nem jelenik meg).

– Database(8): text node-hoz felirat frissítése adatbázisból (2 x bal gomb). Ha előzőleg fence-et hoztunk létre, akkor a paraméter ablakban „Use fence” bekapcsolása után egyszerre több felirat frissíthető.

*Settings/View attributes/Text node:* megjelenítés kikapcsolható.

## **Adatfeltöltés NEW linkage módban**

Ha legalább egy kapcsolt rekord már van, és űrlap is van:

- Database(4): kapcsolt rajzelem kiválasztása, kapcsolt rekord lesz az AE.
- Database(1): AE-t sorban hozzákapcsoljuk minden további kapcsolandó rajzelemhez.
- Database(5): sorra behívjuk az újonnan kapcsolt rekordokat, és űrlappal megfelelően módosítjuk.

## **Példa adatbázis kapcsolatra (Foxpro)**

1. Adattáblák létrehozása (Foxpro) az ustation\database\xbasefox\telek könyvtárban

– telek.dbf létrehozása: tulaj c(20), hrsz n(5), mslink n(10) mezők, 3...5 rekord feltöltése.

– index on mslink tag mslink (telek.cdx létrejön)

– telek.fmt létrehozása szövegszerkesztővel:

```
@ 0, 0 say "Helyrajzi szám: "
```

```
@ 0, 20 get telek->hrsz
```

```
@ 2, 0 say "Tulajdonos: "
```

```
@ 2, 20 get telek->tulaj
```

– MSCATALOG létrehozása

2. Rajz készítés, összekapcsolás (MicroStation)

– új rajz: telek.dgn. Kicsinyítés úgy, hogy grid eltűnjön.

– utca és telkek rajzolása snappinggel.

– telkekre feliratok: alma, körte, dió, barack (ékezetes betűkhöz Font=105).

- adatbázis megnyitása: DB=.
- próba: SQL ablak kinyitása, select \* from telek.
- Settings/Database/Dialog/Linkage mode = Duplicate.
- Key-in: find select \* from telek where hrsz=...
- adatbázis eszköztár: kapcsolat létrehozás (1) gomb, kapcsolat lekérdezés (5) gomb.  
(Formátumozott lekérdezéshez Settings/Database/Dialog/Forms = Text screen beállítás kell.)

### 3. DAS-kapcsolat létrehozása

- Foxpro-val DAS-tábla létrehozás: telekdas.dbf: dastype n(3), formname c(240), sqldas c(240). Egy rekordot veszünk fel: (1, NULL, "select hrsz from telek")
- Felvétel a Data Dictionary-be: datadfox -ta telekdas telekdas
- MS indítás
- SET DATABASE: DAS Table=telekdas
- Text node létrehozás az alma, korte, stb. feliratok mellé ( Main(5,2)/7 ).
- Text node-hoz rekord kapcsolás: Database(1), „find select” módon.
- Text node-hoz DAS kapcsolás: Database(7): kapcsolt rajzelem kiválasztása, majd text node kiválasztása (4 x bal gomb).
- Text node tartalmának frissítése: Database(8)
- Settings/View attributes/Text node: megjelenítés kikapcs.
- Telek táblában hrsz változtatása (FoxPro vagy MS űrlap segítségével), majd rajz frissítése Database(8) segítségével.

## A.4. Referenciafájl, digitalizálás

A rajzhoz háttérképként raszteres vagy vektoros fájlt tölthetünk be, amely nem módosítható. Ezt nevezik referenciafájlnak.

*File/Reference/Display:* típus (raszter vagy vektor) kiválasztása.

*File/Reference/Tools/Attach/Interactive:* referenciafájl csatolása. Kiválasztás után a beillesztés sarokpontjait kell megadni a rajzterületen (figyeljük az utasításokat az alsó sorban).

*Digitalizálási gyakorlat:*

- Új rajz nyitása, Working unit: méter, cm. Master grid = méter. Grid lock bekapcs.
- Kicsinyítés, amíg a grid megjelenik. AccuDraw, (0,0) és (100,0) pont egyszerre látsszon a képernyőn.
- File/Reference/Display: raster
- File/Reference/Tools/Attach/Interactive: fájl típus TIFF, /pub/terinformatika/demo1.tif Preview, OK, alsó sort figyelni: bal alsó sarok = (0,0), jobb alsó sarok = (100,0)
- Állítsuk inverzbe és átlátszóra, ekkor a gridpontok látszanak.
- Vonalarajzolóval digitalizáljunk be néhány épületet piros vonallal! (Nagyítás, grid lock ki, rajzolás Main(2,2) és Main(3,2) funkciókkal, javítás: Main(jobb alsó).

## B. FÜGGELÉK: ArcInfo

Az alábbiakban az ArcInfo rendszer PC változatának adatstruktúráját ismertetjük.

Minden rajzelemhez két azonosító tartozik:

– belső azonosító, amelynek az ArcInfo ad értéket és módosíthatja, a felhasználó nem fér hozzá.

– felhasználói azonosító, amelynek az ArcInfo ad kezdőértéket, de értékét a felhasználó módosíthatja (az ArcInfo ezt nem módosítja).

Az adatstruktúra alapvetően tartományterképek kezelését támogatja. Két szomszédos tartomány határát töröttvonalal írja le, amelyet ívnek (arc) nevezünk. Az ívek találkozási pontjai a csomópontok. Magukat a tartományokat a határoló ívek sorozataként definiálja. Az adatstruktúra elemei:

*a). Pont típusú objektum:* csomópont, vagy pontszerű objektum (pl. kút). Leírásukat a LAB fájl tartalmazza, ennek egy rekordja:

```
label :      belső azonosító (sorszám)
id      :      felhasználói azonosító
x       :      pont x koordináta
y       :      pont y koordináta
```

*b). Ív (vonal) típusú objektum:* töröttvonal (polyline). Leírásukat az ARC fájl tartalmazza, ennek egy rekordja:

```
arc      :      belső azonosító (sorszám)
id      :      felhasználói azonosító
fnode   :      kiinduló csomópont sorszáma
tnode   :      végcsomópont sorszáma
lpoly   :      baloldali poligon azonosítója
rpoly   :      jobboldali poligon azonosítója
num_of_coord : az ív koordinátapárjainak darabszáma
coord [num_of_coord] : koordináták felsorolása
```

*c). Poligon (tartomány) típusú objektum:* a határoló ívek sorozatával definiált. Leírásukat a PAL fájl tartalmazza, ennek egy rekordja:

```
poly;      :      belső azonosító
id;        :      felhasználói azonosító
num_of_arcs; : az ívek darabszáma
arc_[num_of_arcs] : ívek belső azonosítói
```



## C. FÜGGELÉK: Grafikus formátumok

### TIFF formátum

TIFF = Tagged Image File Format, az 1980-as évek végén kidolgozott raszteres képformátum. A TIFF fájl felépítése:

- header,
- IFD-k és tag-ek, amelyek a képet leíró paramétereket tartalmazzák,
- maga a kép (pixelek sorozata).

Header: 8 byte

- 0-1. byte: bytesorrend, két lehetséges értéke:

a) Ascii 'II' = hexa 4949: SHORT, LONG adatokban a kisebb című byte a kisebb helyértékű. Ezt a konvenciót „little endian”-nak nevezik, a PC processzorok így működnek.

b) Ascii 'MM' = hexa 4D4D: SHORT, LONG adatokban a kisebb című byte a nagyobb helyértékű. Ezt a konvenciót „big endian”-nak nevezik, így működnek a Sun és Motorola processzorok.

- 2-3. byte: verziószám, mindig 42 (dec.)
- 4-7. byte: első IFD kezdőpontere (byte sorszám a fájl kezdetétől számítva)

Image File Directory (IFD):

- 0-1. byte: tag-ek száma
- tag-ek felsorolása

Egy tag felépítése (12 byte):

- 0-1. byte: tag azonosítószám
- 2-3. byte: tag típusa (1=BYTE, 2=ASCII, 3=SHORT, 4=LONG, 5=RATIONAL)
- 4-7. byte: hossz (hány db fenti típusú érték van)
- 8-11. byte: a tag értéke (ha elfér 4 byte-on), vagy az érték(ek) kezdőpontere (byte sorszám a fájl kezdetétől számítva).

Fontosabb tag típusok:

<i>megnevezés</i>	<i>azonosító</i>	<i>típus</i>	<i>hossz</i>
kép szélesség	256	SHORT vagy LONG	1
kép magasság	257	SHORT vagy LONG	1
bit-per-érték	258	SHORT	érték-per-pixel
tömörítésmód	259	SHORT	1
(1: tömörítetlen, 2...6: különféle tömörítő eljárások)			
foto-interpr.	262	SHORT	1
(0: fehéret jelent a 0 érték, 1: feketét jelent a 0 érték)			
kép kezdőpont.	273	SHORT vagy LONG	sávok száma
érték-per-pixel	277	SHORT	1
sáv sorok száma	278	SHORT vagy LONG	1

Példa:

Header: 'II', 42, 8

IFD: 6, 256, 4, 1, 2000

257, 4, 1, 3000

258, 3, 1, 1

259, 3, 1, 1

262, 3, 1, 1

273, 4, 1, 110

képpontok sorfolytonosan (8 pixel byte-onként)

## GeoTIFF formátum

Az 1990-es évek közepén definiált, speciális tag-ekkel bővített, térinformatikai célú TIFF formátum. A GeoTIFF kép olyan programokkal is megjeleníthető, amelyek csak az alap TIFF formátumot ismerik, de ezek természetesen nem tudják értelmezni a speciális címkéket.

Sok új privát címke bevezetése helyett egy speciális címkét, a 34735 azonosítójú *GeoKeyDirectoryTag*-et definiáltak. Ezen keresztül lehet elérni az ún. kulcsokat (geoKeys), amelyek a kép térinformatikai leírását adják.

A geoTIFF formátumot elsősorban *georeferencia* leírására használják: ez lényegében az alkalmazott vetületi rendszer leírását jelenti, amely segítségével az egyes pixeleknek megfelelő vetületi koordináták meghatározhatók.

## DXF formátum

Az AutoCAD rendszer fejlesztője, az Autodesk cég által specifikált vektoros adatsere formátum (DXF = Drawing eXchange File format).

Csoport: 2 sorból áll:

- csoportkód (FORTRAN i# formátum)
- érték

Fontosabb csoportkódok:

- 0: fájl elválasztó (vagy elem, tábla bejegyzés kezdete)
- 1: egy elem szöveg értéke
- 2: szekciónév, blokknév, attribútum címke, stb.
- 6: vonaltípus név (pl. CONTINUOUS)
- 9: változónév azonosító (header-ben használatos)
- 10, 11, ..., 18: X koordináta
- 20, 21, ..., 28: Y koordináta
- 30, 31, ..., 38: Z koordináta
- 40, 41, ..., 48: lebegőpontos érték

Szövegfájl, az alábbi szekciókból áll:

- HEADER: változók beállítása (koord. rendszer, stb.)
- TABLES: vonaltípus, réteg, stb.
- BLOCKS: blokk definíciók
- ENTITIES: rajzelemek

*Header változó, típus:*

\$ACADVER 1: AutoCAD verziószám, például az AC1006 érték 10-es verziónak felel meg.

\$EXTMIN 10,20,30: rajzterjedelem bal alsó sarokpontja

\$EXTMAX 10,20,30: rajzterjedelem jobb felső sarokpontja

\$LIMMIN 10,20: rajzhatár bal alsó sarokpontja

\$LIMMAX 10,20: rajzhatár jobb felső sarokpontja

\$TEXTSIZE 40: alapértelmezett szövegmagasság

*Entities:*

LINE 10, 20, 30 (kezdőpont), 11, 21, 31 (végpont)

CIRCLE 10, 20, 30 (középpont), 40 (sugár)

TEXT 10, 20, 30 (beillesztési pont), 40 (magasság), 1 (szöveg)

POLYLINE: töröttvonal kezdete

VERTEX 10, 20, 30: töréspont

SEQEND: polyline vége

*DXF fájl felépítése:*

0	Header szekció kezdete
SECTION	
2	
HEADER	
...	
0	Header szekció vége
ENDSEC	
0	Tables szekció kezdete
SECTION	
2	
TABLES	
...	
0	Tables szekció vége
ENDSEC	
0	Blocks szekció kezdete
SECTION	
2	
BLOCKS	
...	
0	Blocks szekció vége
ENDSEC	
0	Entities szekció kezdete
SECTION	
2	
ENTITIES	
...	
0	Entities szekció kezdete
ENDSEC	
0	
EOF	Fájl vége

*Minta DXF fájl (demo.dxf):*

0	0	0
SECTION	SECTION	VERTEX
2	2	8
<b>HEADER</b>	<b>ENTITIES</b>	0
9	0	6
\$ACADVER	LINE	CONTINUOUS
1	8	10
AC1006	0	112.387200
9	62	20
\$EXTMIN	1	22.591800
10	6	0
111.134400	CONTINUOUS	VERTEX
20	10	8
22.550900	111.134400	0
9	20	6
\$EXTMAX	23.491500	CONTINUOUS
10	11	10
112.414500	112.414500	111.570200
20	21	20
23.709600	23.709600	22.550900
9	0	0
\$LIMMIN	POLYLINE	SEQEND
10	8	8
110.134400	0	0
20	6	0
21.550900	CONTINUOUS	ENDSEC
9	66	0
\$LIMMAX	1	EOF
10	70	
113.414500	0	
20	0	
22.709600	VERTEX	
9	8	
\$TEXTSIZE	0	
40	6	
0.2	CONTINUOUS	
0	10	
ENDSEC	111.556500	
0	20	
SECTION	22.796300	
2	0	
<b>TABLES</b>	VERTEX	
0	8	
ENDSEC	0	
0	6	
SECTION	CONTINUOUS	
2	10	
<b>BLOCKS</b>	112.332700	
0	20	
ENDSEC	23.368800	

## D. FÜGGELÉK: Polinomiális interpoláció

Legyen adott az  $f(x, y)$  függvény értéke az  $(x_1, y_1), \dots, (x_m, y_m)$  pontokban, legyenek ezek az értékek  $d_1, \dots, d_m$ . Az  $f$  függvényt a legkisebb négyzetek módszerével szeretnénk közelíteni, vagyis olyan  $p(x, y)$   $r$ -edfokú polinomot találni, amelyre az

$$E = \sum (p(x_i, y_i) - d_i)^2$$

összeg minimális. A könnyebb írásmód kedvéért legyen  $r = 3$ , ekkor

$$p(x, y) = a_{00} + a_{10}x + a_{01}y + a_{20}x^2 + a_{11}xy + a_{02}y^2 + a_{30}x^3 + a_{21}x^2y + a_{12}xy^2 + a_{03}y^3$$

Az együtthatók számát jelöljük  $m_r$ -rel,  $m_r = 1 + 2 + \dots + (r + 1) = (r + 1)(r + 2)/2$  (például  $r = 1, 2, 3, 4, 5$  esetén rendre  $m_r = 3, 6, 10, 15, 21$ ). Vezessük be a következő jelöléseket:

$$\underline{u} = (1, x, y, x^2, xy, y^2, x^3, x^2y, xy^2, y^3)$$

$$\underline{a} = (a_{00}, a_{10}, a_{01}, a_{20}, a_{11}, a_{02}, a_{30}, a_{21}, a_{12}, a_{03})^T$$

Ekkor

$$p(x, y) = \underline{u}^* \underline{a}.$$

Legyen  $\underline{u}_i$  az  $\underline{u}$  értéke az  $i$ -edik pontra, vagyis

$$\underline{u}_i = (1, x_i, y_i, x_i^2, x_i y_i, y_i^2, x_i^3, x_i^2 y_i, x_i y_i^2, y_i^3)$$

és képezzük az

$$U = \begin{bmatrix} \underline{u}_1 \\ \vdots \\ \underline{u}_m \end{bmatrix}$$

$m \times m_r$ -es mátrixot. Legyen továbbá az adott pontok vektora  $\underline{d} = (d_1, \dots, d_m)^T$ . Ekkor a legkisebb négyzetek feltétele a következőképp írható:

$$E = (\underline{u}_1 \underline{a} - d_1)^2 + \dots + (\underline{u}_m \underline{a} - d_m)^2 = (U^* \underline{a} - \underline{d})^T (U^* \underline{a} - \underline{d})$$

Mivel a polinomot keressük, így ebben a formulában most  $\underline{a}$  elemei az ismeretlenek, és olyan értéküket keressük, amelyre  $E$  minimális. Ez ott fog teljesülni, ahol az  $E(\underline{a})$  függvény parciális deriváltjai nullák, vagyis

$$\partial E / \partial a_i = 2u_{1,i}(\underline{u}_1 \underline{a} - d_1) + \dots + 2u_{m,i}(\underline{u}_m \underline{a} - d_m) = 2(u_{1,i}, \dots, u_{m,i})^* (U^* \underline{a} - \underline{d}) = 0$$

Az összes parciális deriváltra együtt a következő adódik:

$$U^T (U^* \underline{a} - \underline{d}) = 0$$

Innen

$$U^T U^* \underline{a} = U^T \underline{d}$$

A megoldások száma  $m$  és  $m_r$  viszonyától függ:

– Ha  $m < m_r$ , akkor több megoldás is lehetséges, amely az adott pontokra illeszkedik.

- Ha  $m = m_r$ , akkor a megoldás egyértelmű, és pontosan illeszkedik az adott pontokra ( $E = 0$ ). Ezt az  $U^* \underline{a} = \underline{d}$  egyenletrendszer megoldásával nyerjük.
- Ha  $m > m_r$ , akkor pontos illeszkedés általában nem teljesül, a legkisebb négyzetek szerinti optimális megoldást  $\underline{a} = (U^T * U)^{-1} * U^T \underline{d}$  adja.

## E. FÜGGELÉK: Variációs spline interpoláció

A variációs feladatot végelelemes módszerrel oldjuk meg. Négyzetrács szerint felosztjuk a síkot: az  $f(x, y)$  függvény helyett egy  $Z[i, j]$  mátrixot veszünk, ahol  $z_{i,j}$  az  $f$  függvény átlagértéke az  $(i, j)$  négyzeten. A továbbiakban az egyes  $z_{i,j}$  értékeket tekintjük változóknak, és ezek függvényeként írjuk fel az  $E$  energiaértéket.

*Membrán modell* esetén az

$$E_f^1 = \iint (f_x^2 + f_y^2) dx dy$$

integrál értékének a következő felel meg:

$$E_Z^1 = \sum_i \sum_j [(z_{i+1,j} - z_{i,j})^2 + (z_{i,j+1} - z_{i,j})^2]$$

Keressük azon  $Z$  mátrixot, amelyre az  $E_Z^1$  függvény értéke minimális. Ez ott teljesül, ahol a függvény valamennyi  $z_{i,j}$  szerinti parciális deriváltja nulla:

$$\partial E_Z^1 / \partial z_{i,j} = 8z_{i,j} - 2z_{i+1,j} - 2z_{i-1,j} - 2z_{i,j+1} - 2z_{i,j-1} = 0$$

Tehát olyan  $Z$  mátrixot keresünk, amelynek minden  $z_{i,j}$  elemére

$$4z_{i,j} - z_{i+1,j} - z_{i-1,j} - z_{i,j+1} - z_{i,j-1} = 0$$

teljesül. így egy lineáris egyenletrendszerrel kapunk, amely a  $Z$  mátrix minden eleméhez egy egyenletet tartalmaz, és a megoldás adja a keresett  $Z$  mátrixot. Szemléletesen ez azt jelenti, hogy minden egyes mátrixelemre a

$$\begin{array}{ccc} & -1 & \\ -1 & 4 & -1 \\ & -1 & \end{array}$$

maszkot illesztve a szomszédok súlyozott átlaga nullát kell hogy adjon.

Végezzük el *vékonylemez modell* esetén is a fenti levezetést! Itt az

$$E_f^2 = \iint (f_{xx}^2 + 2f_{xy}^2 + f_{yy}^2) dx dy$$

integrál értékének a következő végelelemes összeg felel meg:

$$E_Z^2 = \sum_i \sum_j [(z_{i+1,j} - 2z_{i,j} + z_{i-1,j})^2 + 2(z_{i+1,j+1} - z_{i,j+1} - z_{i+1,j} + z_{i,j})^2 + (z_{i,j+1} - 2z_{i,j} + z_{i,j-1})^2]$$

A  $z_{i,j}$  szerinti parciális derivált értékének meghatározásához a fenti összegből csak a  $z_{i,j}$ -t tartalmazó tagokat kell figyelembe venni. A részletszámítások mellőzésével az alábbi eredményhez jutunk:

$$\begin{aligned} \partial E_Z^2 / \partial z_{i,j} = & + 2z_{i+2,j} + 4z_{i+1,j+1} - 16z_{i+1,j} + 4z_{i+1,j-1} + \\ & + 2z_{i,j+2} - 16z_{i,j+1} + 40z_{i,j} - 16z_{i,j-1} + 2z_{i,j-2} + \\ & + 4z_{i-1,j-1} - 16z_{i-1,j} + 4z_{i-1,j+1} + 2z_{i-2,j} \end{aligned}$$

2-vel leosztva az



$$\begin{array}{cccccc}
 & & & & & 1 \\
 & & & & & 2 & -8 & 2 \\
 1 & -8 & 20 & -8 & 1 & & & \\
 & & & & & 2 & -8 & 2 \\
 & & & & & & & 1
 \end{array} \quad (3)$$

maszk adódik. Vagyis olyan  $Z$  mátrixot keresünk, amelynél minden egyes elemre a fenti maszkot illesztve a környező elemek súlyozott átlaga nullát ad.

Az eddigiek során mind a membrán, mind a vékonylemez modell esetén egy-egy lineáris egyenletrendszer megoldására vezettük vissza a feladatot. A ritka mátrixú egyenletrendszerek szerkezetét szemléltető maszkok sugallják, hogy megoldásuk *iterációs módszerrel* célszerű. Jacobi-iterációt alkalmazva membrán esetén a

$$z_{i,j}' = (z_{i+1,j} + z_{i-1,j} + z_{i,j+1} + z_{i,j-1})/4 \quad (4)$$

iterációs formula használható, amely annyit jelent, hogy a  $Z$  mátrixra ismételten *konvolúciót* alkalmazunk az

$$\begin{array}{ccc}
 & 1/4 & \\
 1/4 & 0 & 1/4 \\
 & 1/4 & 
 \end{array} \quad (5)$$

maszkkal mindaddig, amíg a mátrix be nem konvergál. Az iterációs maszk konvergenciatulajdonságai Fourier-analízissel vizsgálhatók. Kimutatható, hogy vékonylemez modell esetén a (3) feltételből a legkézenfekvőbb módon képezett

$$\frac{1}{20} \begin{bmatrix} & & -1 & & & \\ & & -2 & 8 & -2 & \\ -1 & 8 & 0 & 8 & -1 & \\ & & -2 & 8 & -2 & \\ & & & & & -1 \end{bmatrix}$$

iterációs maszk divergál, ugyanakkor az

$$\frac{1}{32} \begin{bmatrix} & & -1 & & & \\ & & -2 & 8 & -2 & \\ -1 & 8 & 12 & 8 & -1 & \\ & & -2 & 8 & -2 & \\ & & & & & -1 \end{bmatrix} \quad (6)$$

átalakítással már konvergens maszkot kapunk. Megjegyzendő, hogy a gépi számítás gyorsítása érdekében választottuk a  $1/32$  tényezőt, ekkor ugyanis osztás helyett csak 5 bináris helyértéket kell jobbra léptetni.

## IRODALOM

- Bana István: *Az SSADM rendszerszerkezési módszertan*. LSI Oktatóközpont, 1995.
- Belényesi Márta – Kristóf Dániel – Magyar Julianna: *Térinformatika elméleti jegyzet*. Szent István Egyetem, Mezőgazdaság- és Környezettudományi Kar, Gödöllő, 2008. [http://www.kti.szie.hu/TTTT/letoltes/terinformatika/elmeleti\\_jegyzet.pdf](http://www.kti.szie.hu/TTTT/letoltes/terinformatika/elmeleti_jegyzet.pdf)
- Detrekői Ákos – Szabó György: *Térinformatika*. Nemzeti Tankönyvkiadó, Budapest, 2003.
- Elek István: *Bevezetés a geoinformatikába*. ELTE Eötvös Kiadó, 2006.
- Green, D., Bossomaier, T.: *Online GIS and Spatial Metadata*. Taylor and Francis, 2001,
- Katona Endre: *Térképi adatbázisok*. Egyetemi tananyag. Tipotex Kiadó, 2011 (107 oldal) [http://tananyagfejlesztes.mik.uni-pannon.hu/images/stories/vegleges\\_tananyagok/KATONA\\_TERKEPI\\_ADATB/terkepi\\_adatbazis.pdf](http://tananyagfejlesztes.mik.uni-pannon.hu/images/stories/vegleges_tananyagok/KATONA_TERKEPI_ADATB/terkepi_adatbazis.pdf)
- Kertész Ádám: *A térinformatika és alkalmazásai*. Holnap kiadó, 1997.
- Kraus K.: *Fotogrammetria*. Tertia Kiadó, Budapest, 1998.
- Lerner János: *Térképészeti alapismeretek*. ELTE jegyzet, 1989.
- Longley P. A., Goodchild M. F., Maguire D. J., Rhind D. W. (eds): *Geographical Information Systems. Principles, Techniques, Applications and Management*. Second Edition, John Wiley and Sons, 1999.
- Magos Gábor: *MicroStation/J*. MindiGIS Kft, 2000.
- Mezősi Gábor (szerk.): *A mikroszámítógépes módszerek használata a természetföldrajzban*. JATE jegyzet, Szeged, 1991.
- Mezősi G., Bárány-Kevei I., Mucsi L., Balogh I.: *First results of GIS based geocological mapping*. Acta Geogr. Szeg. Tom. XXXI. pp. 71-83, 1993.
- Mucsi László: *Műholdas távérzékelés és digitális képfeldolgozás I. kötet*. Egyetemi jegyzet, JatePress, Szeged, pp. 170, 1995.
- NCGLA Core Curriculum magyar fordítása, hazai alkalmazásokkal kiegészítve. Kiadja: EFE FFFK, Székesfehérvár, 1994. Az anyag négy kötetből áll:
- *Bevezetés a térinformatikába*. (Szerk.: Márkus Béla)
  - *Térinformatikai alapismeretek*. (Szerk.: Márkus Béla, Márton Mátyás, Paksi Judit)
  - *Térinformatikai alkalmazások*. (Szerk.: Mezősi Gábor, Balogh Imre)
  - *Térinformatika Magyarországon '94*. (Szerk.: Márkus Béla)
- Rigaux Ph., Scholl M., Voisard A.: *Spatial databases, with application to GIS*. Morgan Kaufmann Publishers, San Francisco, 2002.
- Ramakrishnan R., Gehrke J.: *Database Management Systems*. McGraw-Hill, 2000. (Chapter 26. Spatial data management)
- Samet H.: *Design and Analysis of Spatial Data Structures*. Addison Wesley, 1989.
- Sárközi Ferenc: *Térinformatika elméleti oktató anyag*. [http://bme-geod.agt.bme.hu/tutor\\_h/](http://bme-geod.agt.bme.hu/tutor_h/).
- Stegena Lajos: *Vetülettan*. Tankönyvkiadó, Budapest, 1988.

*Térinformatika*. A HUNGIS alapítvány kiadásában megjelenő folyóirat. HU ISSN 0864-8549, [www.terinformatika.geocentrum.hu](http://www.terinformatika.geocentrum.hu)

*Térinformatika lépésről lépésre (MicroStation/J, MS Geographics, MS Descartes)*. L-Tér Stúdió Kft, 2001

Ullman J. D., Widom J.: *Adatbázis rendszerek – Alapvetés*. Második, átdolgozott kiadás, Panem, 2008.

Unger J.: *Bevezetés a térképészetbe*. JatePress, Szeged, pp.141, 1992.

Verbyla D. L.: *Practical GIS Analysis*. Taylor and Francis, 2002.

Wilson J. P., Gallant J. C.: *Terrain Analysis*. John Wiley and Sons, 2000.

Zentai L.: *Számítógépes térképészet*. ELTE Eötvös Kiadó, 2000.

Zeiler, M.: *Modeling Our World. The ESRI's Guide to Geodatabase Design*. ESRI Press, 1999.

### **Néhány webhely:**

[fish.fomi.hu](http://fish.fomi.hu) – a FÖMI (Földmérési és Távérzékelési Intézet) által kezelt adatbázis.

[www.gnssnet.hu](http://www.gnssnet.hu) – GNSS Szolgáltató Központ (GPS támogatás)

[maps.google.com](http://maps.google.com) – A Google világtérkép-rendszere

[www.agt.bme.hu](http://www.agt.bme.hu) – A Budapesti Műszaki és Gazdaságtudományi Egyetem térinformatikai oktatási anyagai.

[www.geox.hu](http://www.geox.hu): internetes térképi keresők

[www.gisinfo.hu](http://www.gisinfo.hu) – Magyarország összes településének térképe letölthető (raszteres)

[www.intermap.hu](http://www.intermap.hu) – hazai térképes keresési lehetőségek

[www.microimages.com](http://www.microimages.com) – TNTlite térinformatikai szoftver szabadon letölthető

[www.mapquest.com](http://www.mapquest.com) – USA térképes keresőrendszer

[www.opengeospatial.org](http://www.opengeospatial.org) – az Open Geospatial Consortium honlapja. (Térinformatikai rendszerek együttműködését (interoperability) specifikációk kidolgozásával támogató nemzetközi szervezet.)

[www.usgs.gov](http://www.usgs.gov) – az USGS (United States Geological Survey) honlapja

[www.flag.wr.usgs.gov/USGSFlag/Data/shadedRel.html](http://www.flag.wr.usgs.gov/USGSFlag/Data/shadedRel.html) – digitális terepmodellek az USGS-től

Céges honlapokat lásd a 7. fejezetben.