

Adatbázisok, térképek, információs rendszerek

Elek István

2010

Előszó

A térképészet évszázadokon át tartó fejlődése a térképszerű ábrázolásoktól a pontos vetületi számításokon alapuló transzformációkig hosszú utat járt be. Mint ahogy a matematika az 1600-as évekre szinte minden természettudomány megkerülhetetlen részévé vált, úgy mára az informatika is hasonló szerepkörbe került. Nincs olyan természettudomány, amely ne alkalmazna magas színvonalú, professzionális szoftvereket, adatbázistechnológiát, grafikát. A térképészet eredeti célját megtartva, az informatika, a számítástudomány legfrissebb eredményeit alkalmazza a geoinformatikai szoftverekben. A feladat ma is ugyanaz, mint az elmúlt századokban, vagyis válaszolni a hol mi van, hol mennyi van típusú kérdésekre, de a kornak megfelelő eszközökkel, és annak megfelelő minőségben.

A következőkben bemutatjuk azokat a fogalmakat, elméleti ismereteket, amelyeket a geoinformatika használ. Kicsit mélyebben foglalkozunk a szoftverek háttérében meghúzódó matematikai módszerekkel, a szoftverek háttérében meglévő törvényszerűségekkel, algoritmusokkal, adatmodellekkel, a számítástudomány azon részeivel, amelyek a geoinformatika létét lehetővé tették. Fontosnak tartjuk, hogy a napjainkban egyre nagyobb hangsúlyt kapó gyakorlat-orientált képzésben résztvevő hallgatók ne csak szoftverhasználókká, gyakorlati probléma megoldókká váljanak, hanem igenis legyenek tisztában a térinformatikai szoftverek háttérében meghúzódó elvekkel, elméleti ismeretekkel. Ha nem értjük egy eszköz, gép, szoftver háttérében működő törvényszerűségeket, akkor előbb utóbb jó eszközökkel is helytelen eredményre fogunk jutni.

Áttekintjük a relációs adatbázisok alapfogalmait, a vektoros rendszerek legfontosabb fogalmait, az adatmodelleket, úgy mint a spagetti, a relációs és a topológiai modellt. A raszteres adatmodell alapelvein kívül néhány fontosabb részletkérdésre is kitérünk, úgy mint Gauss-piramis, a georeferencia és a rektifikálás problémáira. Bemutatjuk a képfeldolgozás térinformatikában használt főbb eljárásait, továbbá a távérzékelésben kiemelkedően fontos szerepet játszó osztályozás.

A 3D-s grafika legfontosabb módszereit is áttekintjük, nemcsak a szabályos és szabálytalan mintavételezésű magassági modelleket, hanem a 3D-s megjelenítés módszereit általánosságban, amelyek részben vagy egészben megjelennek a GIS szoftverek 3D-s moduljaiban.

A függelékben összefoglaljuk a fentiek matematikai háttérét, úgy mint a gráfelméleti alapjait, némi mátrixalgebrát, vektorterek elméletét, a Fourier-transzformációt, stb. Ez a függelék természetesen nem pótol alapos matematikai tanulmányokat, célja kizárólag annyi, hogy orientálja az elméleti ismeretekben elmerülni kívánó olvasót további tanulmányok folytatásában.

Budapest, 2010. augusztus

Dr. Elek István
habilitált egyetemi docens

Tartalomjegyzék

1. Az adatbáziskezelés alapvonalai	9
1.1. Relációs adatbázisok	10
1.1.1. Rekord, kulcs	12
1.1.2. Rendezés, indexelés	13
1.1.3. Táblák összekapcsolása	14
1.1.4. Kliens-szerver kapcsolatok	15
1.1.5. Normalizálás	17
1.2. Az SQL-ről röviden	18
2. A térbeliség	23
2.1. Síkbeli transzformációk	25
2.1.1. Egybevágósági transzformáció	26
2.1.2. Hasonlósági transzformáció	26
2.1.3. Affin transzformáció	26
2.1.4. Projektív transzformáció	30
2.1.5. Topológikus transzformáció	31
2.2. A tesszelláció	31
2.3. Térképek és térbeli funkciók	34
2.3.1. Téma, tematika	35
2.3.2. Térbeli objektumok	35
2.3.3. Térbeli funkciók	37
3. A vektoros adatmodell	45
3.1. Spagetti modell	49
3.2. Hálózat modell	51
3.2.1. Lineáris referencia	51
3.2.2. Dinamikus szegmentálás	54
3.3. Relációs modell	54
3.4. Topológikus modell	55
3.4.1. Síkgráfok	56
3.4.2. Befoglalási fa	57
3.4.3. Síkgráf és duális gráf előállítás	59
3.4.4. Síkgráf megváltoztatása	61

3.5.	Térbeli indexelés	61
3.5.1.	Grid index	63
3.5.2.	Négy-fa (Quadtree)	64
3.5.3.	R-fa (R-tree)	67
4.	A raszteres adatmodell	69
4.1.	Raszteres alapfogalmak	69
4.1.1.	Az elektromágneses spektrum	69
4.1.2.	Színmélység és átlátszóság	70
4.1.3.	Színmodellek	71
4.1.4.	Raszteres adatszerkezetek	72
4.1.5.	Intenzitás transzformációk	74
4.2.	Digitális szűrési eljárások	81
4.2.1.	Konvolúciós szűrők	83
4.2.2.	Nemlineáris szűrők	91
4.3.	Szegmentálás, küszöbölés	94
4.4.	LOD algoritmusok	97
4.4.1.	A Gauss-piramis	98
5.	3D grafika, felület modellezés	101
5.1.	A 3D-s domborzat modellezés áttekintése	101
5.1.1.	Vektoros domborzat leírás, a TIN modell	102
5.1.2.	Raszteres domborzat leírás, a DEM modell	103
5.2.	A 3D-s megjelenítés alapelvei	108
5.2.1.	A virtuális színtér	108
5.2.2.	A geometriai modellezés	109
5.2.3.	Modellek elhelyezése a színtérben	109
5.2.4.	A grafikus szerelőszalag	111
5.2.5.	A megvilágítás modellje	113
5.2.6.	Egy tériformatikai példa	115
6.	Adatbányászati módszerek	119
6.1.	Osztályozás, szegmentálás	119
6.1.1.	Távolság mátrix	119
6.1.2.	Particionáló klaszterezés	120
6.1.3.	Hierarchikus eljárások	121
6.1.4.	Képek klaszterezése	121
6.2.	Dimenzió csökkentés PCA-val	123
6.2.1.	Tematikus térkép és osztályozás	125
A.	Matematikai összefoglaló	129
A.1.	Gráfelmélet	129
A.2.	Mátrixszámítás	144
A.3.	Vektorterek	151

TARTALOMJEGYZÉK

7

A.4. Komplex számok	157
A.5. A digitális szűrések matematikai alapjai	165

1. fejezet

Az adatbáziskezelés alapvonalai

A GIS-ről nem lehet beszélni alapos adatbáziskezelési ismeretek nélkül. Az adatbázis adatok rendezett formában tárolt gyűjteménye. Az adatokat táblákba szervezve tároljuk. A táblák rekordokból (sorokból) állnak. Mindegyik rekord ugyanannyi számú mezőt tartalmaz, mindegyik mező meghatározott típusú (character, string, integer, float, boolean, date & time, stb.). Valószínűleg a legismertebb példa a címlista. Az adatbázis mindegyik bejegyzése egy rekord: egy ismerősünk neve és címe. Mindegyik rekord mezőkből áll: családnév, keresztnév, lakáscím, telefonszám és így tovább.

Az adatbázist legtöbbször táblázatos formában ábrázolják, sorokra és oszlopokra bontva. Minden sor egy rekordot jelent, és minden oszlop egy mezőt. Emiatt a rekordokat gyakran csak soroknak, a mezőket pedig oszlopoknak hívjuk (1.1. ábra).

A legtöbb adatbázis valójában több táblát is képes tárolni, egyelőre azonban vizsgáljuk csak a címlistát. Adatbáziskezelő programokkal szembeni általános követelmények első közelítésben a következők:

- új rekordok hozzáadása,
- létezők módosítása,
- törlése,

Név	Cím	Telefon
Fónagy Félix	Üllői út 32.	343-565
Kádár Mihályné	Igló utca 8.	272-129
Lengyel Sarolta	Jáhn Ferenc utca 24.	472-142
Tatár Imre	Madách tér 65.	221-062

1.1. ábra. A legegyszerűbb, egytáblás adatbázis: a címlista. Ebben az esetben teljes az adatleírás.

- adatok keresésének lehetősége.

Egy címlistát nyilvántartó programtól elvárjuk, hogy például rendezhessük az adatokat név szerint. Ha adatbázis fejlesztésbe fogunk, majdnem minden esetben relációs adatbázist fogunk használni.

1.1. Relációs adatbázisok

Minden modern adatbázis-kezelő rendszer relációs adatbázisokat használ (alig akad néhány kivétel). Az egytáblás adatbázis egyetlen táblát tartalmaz. Összesen két dimenziója van, a sorok és az oszlopok (rekordok és a mezők). Az egytáblás adatbázis-kezelés tökéletes megoldás például a címlista esetén, de ez inkább speciális, mint tipikus eset.

Az egytáblás rendszer és a relációs adatbázis közötti különbséget legjobban egy példával tudjuk bemutatni. A példában egy ingatlan nyilvántartó rendszert fogunk vizsgálni, amely az ingatlanok adatain kívül azok tulajdonosait is kezelni kívánja. Lássuk először az ingatlanok adatait leíró táblát:

- Helyrajzi szám
- Irányító szám
- Település
- Utca
- Házsám
- Terület
- Művelési ág
- Építési övezeti besorolás
- Jelzalog (ha van)

Ez az adatartalom az ingatlan teljes fizikai leírását adja, de a tulajdonosokról nem árul el semmit. Ha most azt is tárolni szeretnénk, hogy az ingatlanoknak kik a tulajdonosai, akkor az egytáblás módszer használhatatlan. Képzeld el, hogy számos ingatlannak több tulajdonosa is van. Ha a tulajdonos adatait is egytáblás struktúrába erőltetjük, akkor az ingatlan fizikai adatait annyiszor kellene tárolnunk, ahány tulajdonos van, ami képtelenség. Ezzel nemcsak sok felesleges adatot tárolnánk, hanem az adathibák létrejöttének valószínűségét is megnövelnénk, több más egyéb hátrányos következmény mellett (pl. ha egy tulajdonos címe megváltozik, akkor a változásvezetést annyi rekordon kell elvégezni, ahány telke van az illetőnek).

A megoldás egy relációs adatbázis létrehozása. Ahelyett, hogy egyetlen táblában tárolnánk minden adatot, a relációs adatbázisban több táblát hozunk létre, amelyekbe szétsztyjuk az információt. Ebben az esetben a relációs adatbázis két táblából fog állni:

A *Ingtatlanok* tábla

- Helyrajzi szám
- Irányító szám
- Település
- Utca
- Házszám
- Terület
- Művelési ág
- Építési övezeti besorolás
- Jelzalog (ha van)

A *Tulajdonosok* tábla

- Tulajdonos családneve
- Tulajdonos keresztnéve
- Irányító szám
- Település
- Utca
- Házszám
- E-mail címe
- Telefonszáma
- Helyrajzi szám

Ebben a struktúrában nincs többszörösen tárolt adat. Minden tulajdonoshoz egy rekord tartozik a *Tulajdonosok* táblában, és minden ingatlanhoz egy rekord tartozik a *Ingtatlanok* táblában. A két tábla egy olyan mezővel van összekapcsolva, ami mindkettőben megtalálható, vagyis a *Helyrajzi szám* nevű mezővel. A táblákat úgy kell tervezni, hogy az összes szükséges információt tartalmazzák.

Az adatbázisok létrehozása nem öncélú, valamilyen funkciókat kívánunk megvalósítani velük. Vegyük például a következő egyszerű funkcionalitást:

- Automatikus felvételi lehetőség a tulajdonosokhoz (ha még nem szerepel a *Tulajdonosok* táblában), amikor egy új elemet visznek fel a *Ingatlanok* táblába.
- Meg lehessen jeleníteni, illetve ki lehessen nyomtatni a tulajdonos adatait egy ingatlannal kapcsolatban.
- Ki lehessen listázni az összes ingatlant egy adott tulajdonosra vonatkozólag.
- Meg lehessen jeleníteni, hogy egy megadott ingatlannak kik a tulajdonosai

1.1.1. Rekord, kulcs

Egy önmagában álló tábla valamely relációs adatbázisban semmiben sem különbözik egy olyan táblától, amit egy egytáblás adatbázisban találhatunk. Minden oszlop, illetve mező az információ egy darabkáját tartalmazza, és minden rekord, illetve sor valamely elem összes adatát tartalmazza. Az esetek túlnyomó többségében minden rekord egy valóságos dolgot reprezentál, mint például telkeket, embereket, stb.

Az adatbázis-kezelésben használatosak még egyéb kifejezések is. A táblára néha entitás vagy entitás osztály néven hivatkoznak. A térinformatikában ugyanezt a fogalmat gyakran réteg, layer vagy feature class néven említik. A sorokat (rekordokat) entitás-előfordulásnak nevezik. Az oszlopokat (mezőket) attribútum osztálynak is nevezhetjük. Egy attribútum osztály vagy mező képviseli az adat legelemibb formáját, az információ legkisebb egységét. Egy sor és oszlop kereszteződése, metszete – egy adott rekord adott mezője – egy attribútum. Egy attribútum az adott valóságbeli objektumról szolgáltat egyfajta információt, mint például egy adott ingatlan címét.

A relációs adatbázisok táblái esetében minden táblában kell legyen egy elsődleges kulcs, egy olyan mező, amely egyértelműen azonosít minden sort az adott táblában, vagyis az elsődleges kulcsban tárolt adatnak minden rekord esetén különbözőnek kell lennie a táblában. A példa adatbázisban a *Helyrajzi szám* a *Ingatlanok* tábla elsődleges kulcsa, a *Tulajdonos családneve* pedig a *Tulajdonosok* tábláé. Érdekes problémát fel a névazonosság, mert ebben az esetben a *Tulajdonos családneve* nem képes eljátszani az elsődleges kulcs szerepét. Emiatt az adatbáziskezelők gyakran maguk állítanak elő elsődleges kulcsmezőt (pl. egy inkrementálisan növelt értékű mezőt), aminek nincs más szerepe, mint egyedinek lenni.

A legtöbb adatbáziskezelő alkalmazás támogatja összetett elsődleges kulcsok használatát, ami azt jelenti, hogy az elsődleges kulcsban tárolt adat több mező értékéből tevődik össze. Az összetett elsődleges kulcsokat akkor használják, ha a két vagy több mezőből összevont adatra mondható, hogy egyértelműen különbözik a tábla minden sorában, ellentétben bármelyik önállóan vett mezővel.

Idegen kulccsal állunk szemben, ha egy mező olyan adatot tartalmaz, amely egy másik tábla elsődleges kulcsára mutat. Az idegen kulcsnak nem kell különbözőnek lennie minden rekordra, és a legtöbbször nem is az. A *Tulajdonosok* táblában a *Helyrajzi szám* nevű mező egy idegen kulcs. A *Ingatlanok* tábla nem tartalmaz idegen kulcsot.

A táblákra szokás szülő-, illetve gyermektáblaként is hivatkozni. A gyermektábla adatai függenek a szülő tábla adataitól. A szülő tábla adataira nincs semmilyen függőség, ezek teljesek, ahogy vannak. A példában a *Tulajdonosok* tábla kapcsolódik a *Ingatlanok* táblához csak a két tábla együttes használatával tudjuk az árucikkek adatait kezelni. Általában a szülő táblákat azért hozzák létre, hogy elkerüljék a redundanciát, az adatismétlődést.

1.1.2. Rendezés, indexelés

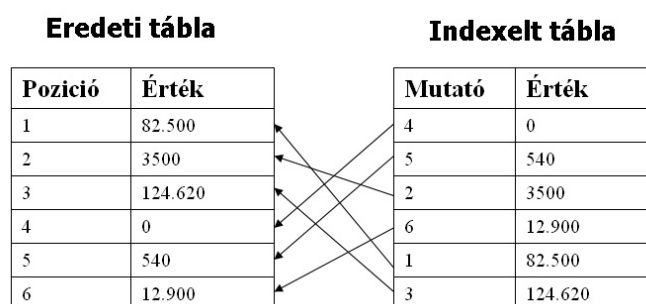
A táblák rekordjai nem szükségszerűen rendezettek. A rekordok fizikai sorrendje általában megegyezik azzal a sorrenddel, ahogyan a rekordokat felvitték, aminek ritka kivétellektől eltekintve nincs különösebb jelentése. Az adatbázis alkalmazásnak el kell rejtenie a felhasználó szeme elől a fájlokat, fájlstruktúrákat és mindent, ami ezekkel kapcsolatos. Éppen ez a relációs adatbázisok egyik nagy előnye.

A fájl fizikai struktúrája, és adatbázis táblák logikai struktúrája két teljesen különböző dolog. Éppen ezért logikailag rendezhetjük a táblák adatait anélkül, hogy törödnünk kellene a fájlokban lévő adatsorrenddel. A rendezés nagyon fontos az adatbázisok használatakor. Ahelyett, hogy az alkalmazás fizikailag rendezné a rekordokat, logikai rendezést alkalmaz indexet használva.

Ha rendezni akarjuk a rekordokat egy adott mező szerint, akkor egyszerűen meg kell mondanunk az adatbázis-kezelő programnak, hogy használja az ezen a mezőn alapuló indexet (1.2. ábra). A megjelenítés, nyomtatás során a tábla rekordjai a megadott index szerinti rendezésben jelennek meg. Könnyen átállíthatjuk a rendezési szempontot – rendezzünk például név szerint, az irányítószám helyett – egy másik index használatával.

Az index nagyban meggyorsíthatja az indexelt mező adataira alapozott kereséseket és lekérdezéseket. Ha adatot keresünk egy nem indexelt mezőben, akkor az adatbázis alkalmazásnak végig kell néznie az összes rekordot minden egyes keresés alkalmával. Ha viszont van index a mezőre, akkor a rendszer sokkal gyorsabban tud keresni, például bináris kereséssel (az ehhez szükséges adatstruktúrát, a bináris fát láthatjuk az **A Függelékben**, a A.17. ábrán, a 143. oldalon). A bináris keresés a következőképpen működik:

1. Megkeresi a rendezés szerinti középső rekordot.
2. Egyezik a rekord a keresettel? Ha igen, akkor megtaláltuk a keresett adatot. Ha nem, akkor jön a következő lépés.



1.2. ábra. Minden index bejegyzés tartalmaz egy mutatót a megfelelő rekord helyére

3. Ha a keresett rekord adott mezőjének értéke „kisebb, mint” az aktuális rekordé, akkor a keresett rekordnak (ha van ilyen) az index első felében kell lennie. Ha nem, akkor a másodikban.
4. Tekintsünk el az indexnek attól a felétől, ahol a keresett rekord nem lehet, majd térjünk vissza az első lépésre mindaddig, amíg el nem érünk egy előre megadott felosztást, vagy meg nem találjuk a rekordot, vagy el nem fogy a még végig nem nézett rekordok listája.

A bináris kereséshez, természetét tekintve, igencsak hasonlatos egy több dimenziós keresési algoritmus, a quad-tree (négyfa) algoritmus, amely sok GIS rendszer adatstruktúrájában játszik fontos szerepet, de erről majd később.

Amíg az adatbázist csak tervezzük, rendszerint felmerül a következő kérdés: melyik mezőre, illetve mezőkre rendezzük a táblát? Bár növeljük a lehetőségeket azzal, hogy több indexet használunk, a teljesítmény rosszabb is lehet ennél, mivel minden indexet frissíteni kell, ha új rekord kerül a rendszerbe vagy ha egy meglévőt módosítottunk.

A táblákat feltétlenül rendezni kell az elsődleges kulcsra, ezt számos adatbázis-kezelő automatikusan el is végzi. Valamint indexelnünk kell a táblát minden idegen kulcsra is, ami előfordul benne. Minden további index az alkalmazás részleteitől függ, és általában azokra a mezőkre korlátozódik, amely a lekérdezések alapjául szolgál.

1.1.3. Táblák összekapcsolása

Az egyik legfontosabb feladat, hogy információt nyerjünk az adatbázis tábláiból. Ha ez a művelet kettő vagy több táblát igényel, akkor ezt többtáblás lekérdezésnek nevezzük, vagy általánosabb nevén összekapcsolásnak (join). Három különböző módon kapcsolhatunk össze két táblát:

1 : 1 Akkor történik, ha a gyermektáblának legfeljebb csak egy rekordja tartozik a szülő tábla egy rekordjához. Ez a fajta kapcsolat ritka a relációs adatbázisokban. Ha két tábla 1:1-hez kapcsolatban áll egymással, akkor mezőiket összerakhatjuk egy táblába, ez általában jobb megoldás. Ilyen kapcsolatot csak akkor szoktunk használni, ha bizonyos tulajdonságokat külön akarunk tárolni, vagy egymástól távoli helyeken lévő adattáblákat szeretnénk összekapcsolni, esetleg nincsen jogunk a két tábla összevonásához, mert csak az egyiknek vagyunk tulajdonosai.

1 : n Ebben az esetben a gyermektáblából több rekord tartozhat az szülő tábla egy rekordjához. Ez a legáltalánosabb típusa az összekapcsolásnak, ezt használják leggyakrabban az adatbáziskezelő programokban. Az ingatlan nyilvántartási példánál maradva, egy a többhöz kapcsolat van a *Ingatlanok* és a *Tulajdonosok* tábla között, hiszen több telek is tartozhat egy adott tulajdonoshoz.

n : m A gyermektábla több rekordjához a szülő táblának több rekordja is tartozhat. Például, ha egy teleknek több tulajdonosa van, de ezek között vannak olyanok, akik több telket is birtokolnak, akkor több a többhöz kapcsolattal van dolgunk. Természetesen ez a fajta összekapcsolás nem alapulhat egyik tábla elsődleges kulcsán sem. Az n : m kapcsolat nem vehető igazi összekapcsolásnak, mert szükség van hozzá egy kapcsolótáblára (kereszt tábla), amely az idegen kulcsok értékeit tartalmazza, és a két tábla közötti összekapcsolást definiálja. A kapcsolótábla 1 : n és 1 : m kapcsolatban áll mind a két táblával.

Amikor egy adatbázis létrehozásán dolgozunk, tulajdonképpen egy felhasználói alkalmazást használunk (front-end application), vagy készítünk, amellyel majd az adatokat nézegethetjük és manipulálhatjuk. A szerveroldal (backend) a relációs adattáblák halmaza, és számos esetben az adatbázismotor az, amely kapcsolatot biztosít a táblák és a felhasználói alkalmazás között. Néha a kliens és a szerver kifejezéseket is használjuk a felhasználói alkalmazás és a szerveroldal szavak helyett.

1.1.4. Kliens-szerver kapcsolatok

Felmerülhet bennünk a kérdés, hogy miért nem egyesítünk minden adatbázis-feladatot egy alkalmazásba? A válasz erre a kérdésre világos, ha figyelembe vesszük, hogy milyen szervezetek használják az adatbázisokat. Egy szervezet különböző adatbázisai széles skálán mozognak. A szervezet különböző felhasználóinak az adatok részhalmozaira van szüksége, valamilyen számukra meghatározott formában. Ha a kliens és a szerver egy alkalmazásba lenne integrálva, akkor ennek az alkalmazásnak le kellene fednie az összes lehetséges felhasználó igényét. Ez elkerülhetetlenül naggyá, bonyolulttá és nehezen kezelhetővé tenné a rendszert.

A kliens-szerver megközelítés megoldja a problémák nagy részét. A felhasználók minden csoportjának kialakítható a saját igényei szerinti szoftver, melyet úgy tervezhetünk meg, hogy a csoport igényeit kielégítse, de nem nyújt semmi többet. Egy egyszerű programmal – a szerver belső motorjával – koordinálhatunk

minden hozzáférést az aktuális adatbázisfájlokhoz, az előbb említett problémák sokkal könnyebben kezelhetők, az adatbázis különböző részeihez való hozzáférési jogosultságok ellenőrzése is könnyebb lesz. A kliensek két kategóriába sorolhatók:

Döntéstámogató alkalmazás: lehetővé teszi a felhasználók számára, hogy az adatbázis adatait nézgezzék, lekérdezéseket készítsenek, de nincs joguk módosítani vagy felvinni adatokat.

Adatrögzítő alkalmazás: lehetővé teszi adatok felvitelét, törlését és szerkesztését az adatbázisban.

A döntéstámogató alkalmazásokat egészen szélsőségesen lehet specializálni és korlátozni. Ezek a programok nagyon rugalmasak is lehetnek, mint például egy bonyolult menedzsment információs rendszer (MIS), amely összegző adatokat biztosít a cég adatbázisának teljes tartalmáról. A térinformatikai rendszerek gyakran ilyenek. A döntéstámogató alkalmazásoknak általában csak olvasni van joguk az adatbázisban, azaz nem változtathatják meg az adatokat.

Két megközelítés létezik ebben az alkalmazástípusban, akár egymással kombináltan is, attól függően, hogy mennyire intelligens a program. A felhasználó több, előre definiált vizsgálódást hajthat végre az adatokon, vagy lehetősége nyílna arra, hogy maga tervezze meg a lekérdezéseket (query) és jelentéseket (report).

Az első módszer előnye, hogy nem igényel túl nagy erőfeszítést a végfelhasználóktól, de előfordulhat, hogy a program előre definiált lehetőségei nem egyeznek meg a felhasználó igényeivel.

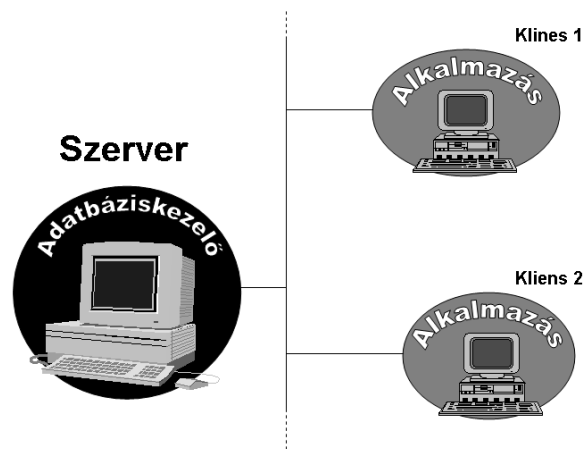
Az adatrögzítő alkalmazások azt teszik lehetővé, hogy új adatokat vehessünk fel az adatbázisba, illetve, hogy a már meglévőket szerkeszthessük. Mivel egy adatrögzítő alkalmazás módosíthatja az aktuális tábla adatait, néhány további programozási megfontolást is számításba kell vennünk. Az alkalmazásnak figyelnie kell a táblázat adatainak egységességére és helyességére (integritás).

A legtöbb adatbázisrendszer úgynevezett kliens-szerver felépítésű (1.3. ábra). A kliens és szerver szavak az adatok tárolásának feladatát végző program (szerver) és az adatokkal való munkához használható program (kliens) elkülönítésére szolgálnak.

Az adatbázis alkalmazások legtöbbször hálózati környezetben működnek. Az adatbázisok egy központi gépen (szerveren) vannak, amit a hálózat bizonyos felhasználói, munkaállomásai el akarnak érni, vagyis az adatok, de gyakran a programok is a szerveren vannak, ugyanakkor a tényleges adatfeldolgozás a munkaállomáson folyik.

Hogyan működik a kliens-szerver architektúrájú rendszer?

- A kliens megad egy lekérdezést, és megszólítja a szerveret, vagyis elküldi a kérést a szervernek
- A szerveren futó szerver alkalmazás megnyitja az adatbázist, és elvégzi a lekérdezést, megkapja a választ
- A választ elküldi a hálózaton keresztül a kliensnek



1.3. ábra. Kliens-szerver architektura.

- A konkurens hozzáférésből eredő problémák az éppen editálás alatt lévő rekord lezárásával (lock) megoldhatók, miközben a többi rekord – valamennyi táblát beleértve – hozzáférhető marad más felhasználók számára.

A kliens-szerver architektúrának nemcsak a helyi hálózatok esetében van jelentősége, hanem különösen hasznos ez a konstrukció a WEB-re fejlesztett információs rendszerek esetében. Itt ugyanis a sávszélesség – vagyis a hálózat adatátbocsátó képessége – lényegesen rosszabb, mint a helyi hálózatok esetében, tehát különösen fontos, hogy csak olyan adatsomagok utazzanak a hálózaton, amire feltétlenül szükség van.

1.1.5. Normalizálás

A normálformák azt a célt szolgálják, hogy ne törölhessünk ki igazi adatokat az adatbázisból, valamint, hogy hamisak se kerülhessenek bele. Az egyik lehetséges e hibák elkerülésére, hogy egy tényt csak egyszer jelölünk az adatbázisban. Ha ugyanis többször előfordulna, akkor gondoskodnunk kellene az állandó összhang megteremtéséről, és ez kaotikus helyzetet eredményezne.

Első normál forma. Az első normálforma azt jelenti, hogy a tábla nem tartalmaz ismétlődő csoportokat. Minden oszlopban csak elemi adat van és nem tömb vagy lista, amelynek saját stuktúrája lehet. Ha a tábla tartalmazna ismétlődő csoportokat, akkor számos nehézséggel kerülnénk szembe. Ebben az esetben előfordulhat, hogy több adatot törölünk az adatbázisból, pedig csak egyet akartunk (törlési probléma), egy adat módosításával hamis adatok is bekerültek az adatbázisba, pedig csak egyet akartunk megváltoztatni (módosítási probléma), vagy nem tudunk felvenni egy

egyszerű adatot önmagában (beszűrési probléma). A megoldás több tábla alkalmazása, amelyek mindegyike egy kapcsolatot illetve egy egyszerű ténytet jelöl.

Második normál forma. Egy tábla második normál formában van, ha első normál formában van, és a kulcsa meghatározza az összes nem kulcs tulajdonságot a táblában.

A harmadik normál forma. Egy tábla harmadik normál formában van, ha minden olyan oszlop, amely nem kulcs, meghatározható a kulccsal, az egész kulccsal, és csak a kulccsal.

1.2. Az SQL-ről röviden

Az SQL-t (Structured Query Language) a legáltalánosabban elfogadott, standardizált lekérdező eszközkészlet. A neve ellenére az SQL nem egy hagyományos értelemben vett programnyelv, hanem egy szabvány, amit a legkülönbözőbb szoftvergyártók előszeretettel építenek rendszereikbe, hiszen aki egyszer megtanulta az SQL használatát, az bármely adatbázis-kezelő környezetben könnyedén boldogul, mert nagyon gyorsan képes lesz akár a legbonyolultabb lekérdezések összeállítására is. Ráadásul a mértékadó adatbáziskezelő gyártók (ORACLE, SQL Szerver, MySql, PosgreSQL) jelentős térbeli lekérdező (Spatial Extension, Spatial Modul) funkcionalitást is építettek a rendszereikbe.

Az SQL, jellegéből adódóan, parancsértelmező üzemmódban működik, hiszen az SQL utasítások parancsok formájában öltönek testet. A kívánt parancsot a felhasználó begépezi, a parancsértelmező lefordítja, a gép végrehajtja. Ez egy lehetséges megoldás, de csak bizonyos feladatokra, és csak bizonyos szakemberek számára (pl. adatbázis rendszergazdák). Egyszerű felhasználók számára az SQL felület nem kínál igazi megoldást, mivel az SQL-t főként informatikai szakemberek ismerik.

Programokba is beépíthetők SQL utasítások, de ezekben az esetekben valamely magasszintű programnyelv nyújt számunkra lehetőséget az SQL utasítások beágyazására. Egyébként a legtöbb szoftverfejlesztő eszköz kínál SQL felületet, illetve utasításkészletet, amellyel szabványos SQL parancsok generálhatók és végrehajthatók.

Az SQL alapú rendszerek a tranzakciós elv alapján működnek, vagyis addig nem történik meg az adatok tényleges megváltozása az adatbázisban, amíg az éppen végrehajtás alatt lévő művelet sikeresen be nem fejeződik. Ez tehát azt jelenti, hogy a változások először csak ideiglenes hajtódnak végre. Külön erre a célra szolgáló utasítás véglegesíti a változásokat.

Adatleíró nyelv (Data definition language)

Lássuk az SQL néhány elemét. Mint tudjuk az adatbázisok alkotó elemei a táblák. Ezek létrehozására való a

CREATE TABLE utasítás. Segítségével definiálhatók a táblákban szereplő oszlopok nevei típusai, stb.

CREATE VIEW utasítással a már feltöltött táblákból jeleníthetünk adatokat tetszőleges csoportosításban.

CREATE INDEX utasítással létrehozhatunk egy adott tábla valamely oszlopára vonatkozó indexet.

ALTER TABLE utasítással lehetőségünk van egy már létező tábla struktúrájának a megváltoztatására.

DROP TABLE segítségével eltávolíthatunk egy táblát az adatbázisból.

Adatkezelő nyelv (Data manipulation language)

A **SELECT** utasítás az SQL leggyakrabban használt utasítása. Segítségével adatokat válogathatunk le az adatbázisból, mégpedig olyanokat, amelyek az általunk megadott feltételeknek megfelelnek. Az **INSERT** utasítással új sorokat fűzhetünk egy adott táblához. Ennek ellenkezője a **DELETE** utasítás, amivel sorokat törölhetünk egy táblából. Ezzel lényegében át is tekintettük a legfontosabb utasításokat. (Ez a kijelentés persze erős túlzás, hiszen vastos kötetek szólnak az SQL részleteiről.)

Klauzulák

A klauzulák segítségével finoman szabályozható feltételeket adhatunk meg utasításaink számára. A **FROM** klauzulával megadhatjuk, hogy melyik táblából kívánunk szelektálni. A **WHERE** klauzulával a lekérdezés feltételét adhatjuk meg, vagyis azt, hogy mely feltételeknek eleget tevő rekordokat kívánunk legyűjteni. A **GROUP BY** klauzulával csoportokra bontva kapjuk meg a lekérdezés eredményét. Az **ORDER BY** klauzulával valamely oszlop szerint rendezve kapjuk a lekérdezés eredményét.

Operátorok

Logikai és összehasonlító operátorok állnak rendelkezésünkre a **WHERE** klauzulán belül. Ezek a következők, úgymint logikai operátorok: **AND** (és), **OR** (vagy) **NOT** (negáció), valamint az összehasonlító operátorok, úgymint **<** (kisebb mint), **>** (nagyobb mint), **<=** (kisebb vagy egyenlő), **>=** (nagyobb vagy egyenlő), **=** (egyenlő), **≠** (nem egyenlő), **BETWEEN** (két érték között), **LIKE** (hasonló, vagy minta illesztés).

Aggregáló függvények

Az aggregáló függvények segítségével a **SELECT** utasításon belül bizonyos mezők összevont értékét adhatjuk egy adott mezőnek. Az aggregáló függvények a következők: AVG (átlag), COUNT (beütésszám, hány rekord felelt meg a lekérdezés feltételeinek), SUM (összeg), MAX (legnagyobb érték), MIN (legkisebb érték).

Példák

Összeállítottunk néhány példát egyszerűbb lekérdezésekre. Noha kevés eleműnek, és egyszerűnek látszik az SQL, mégis nagyon bonyolult lekérdezések állíthatók össze, ha több táblából, bonyolult feltétel rendszeren keresztül kívánunk lekérdezni. Trivialitás, de mégis fontos megjegyezni, hogy ahhoz, hogy kézben tudjuk tartani adatbázisunkat az SQL-lel, mindenek előtt jól kell ismernünk az adatbázisunkat, milyen táblákat, milyen mezőket tartalmaz.

```
SELECT fieldname FROM tablename
```

vagyis válogassuk le a *tablename* nevű táblából a *fieldname* nevű mező adatait.

```
SELECT fieldname1, fieldname2 FROM tablename
```

vagyis válogassuk le a *tablename* nevű táblából a *fieldname1* és a *fieldname2* nevű mezők adatait.

```
SELECT * FROM tablename
```

vagyis válogassunk le mindent a *tablename* nevű táblából.

```
SELECT * FROM tablename WHERE Country='Hungary'
```

vagyis válogassuk le a *tablename* nevű táblából az összes olyan rekordot, amelyben a *Country* nevű mező értéke 'Hungary'.

```
SELECT * FROM tablename1 WHERE Country='Hungary' OR  
Country='England'
```

vagyis válogassuk le a *tablename1* nevű táblából az összes olyan rekordot, amelyben a *Country* nevű mező értéke 'Hungary' vagy 'England'.

```
SELECT * FROM tablename WHERE Area < 90000 ORDER BY Population
```

vagyis válogassuk le a *tablename* nevű táblából az összes olyan rekordot, amelyben az *Area* nevű mező értéke nagyobb 90000, és ezeket rakjuk sorba a *Population* mező nagysága szerint.

```
SELECT * FROM tablename2 WHERE ContactName LIKE 'A%'
```

vagyis válogassuk le a *tablename* nevű táblából az összes olyan rekordot, amelyben a *ContactName* nevű mező első betűje „A”.

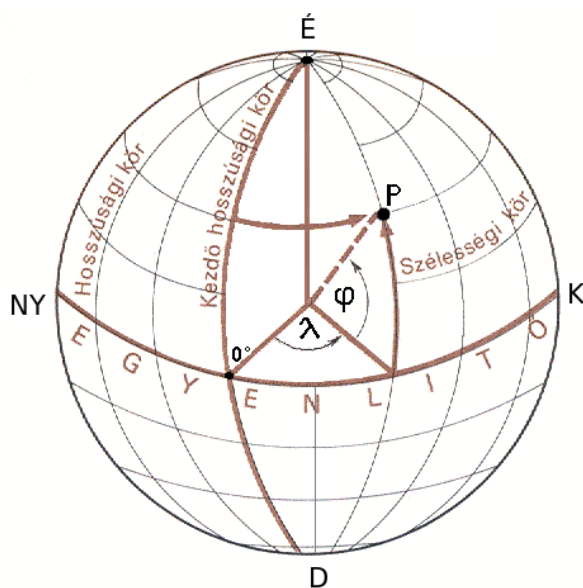
```
SELECT tablename1.Country, tablename2.ContactName FROM tablename1  
INNER JOIN tablename2 ON tablename1.ID=tablename2.ID
```

vagyis válogassuk le a *tablename1* táblából a *Country* nevű mezők, valamint a *tablename2* táblából a *ContactName* nevű mezők adatait, de csakis azokat, amelyeknek ID nevű mezőik azonosak. Ezzel az utasítással két különböző táblát kapcsolunk össze, és válogattunk le belőlük összetartozó adatokat.

2. fejezet

A térbeliség

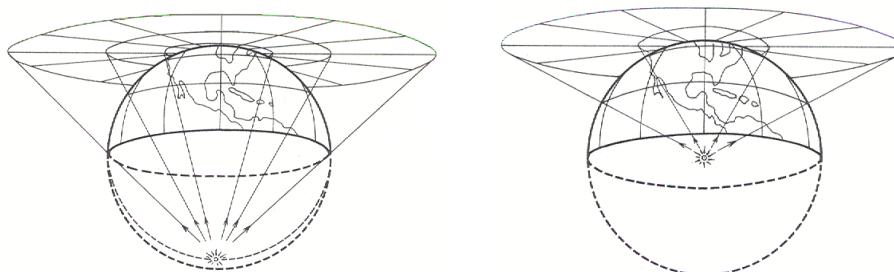
A föld első közelítésben gömb alakú, melynek leírására kézenfekvő vonatkoztatási rendszer a polárkoordináta rendszer, amely λ és φ szögekkel és az r sugárral adja meg egy tetszőleges P pont koordinátáit (2.1. ábra). (Fogadjuk el itt ezt a nagyvonalú megközelítést, mert a vetületani fejezet úgyszólván részletesen megmutatja, hogy a Föld nem gömb alakú, a referencia felület nem gömb, stb.).



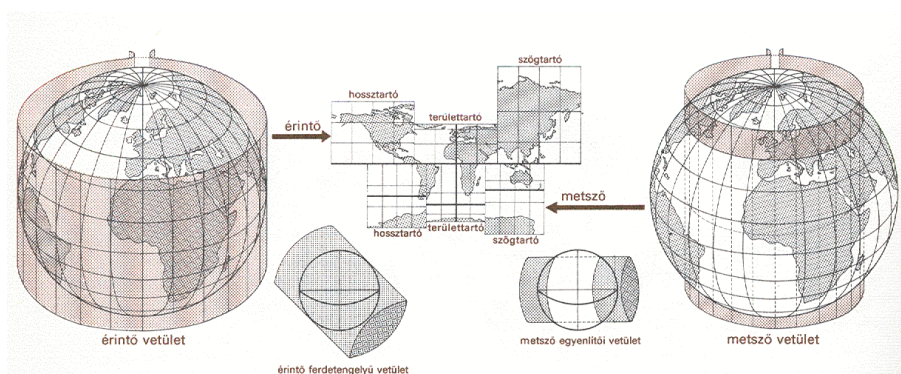
2.1. ábra. A gömfelszín leírása polárkoordinátákkal (Klinghammer [19])

A térkép egyik legfontosabb feladata a térbeliség kifejezése, vagyis a gömb alakú Föld felszínén való elhelyezkedés megmutatása. Akár a nyomtatott papírt, akár a monitoron megjelenő digitális térképet nézzük, a gömbfelület síkra történő vetítése megkerülhetetlen, aminek viszont egyenes következménye a keletkezett kép torzítása. Meg kell barátkoznunk azzal a gondolattal, hogy gömbfelület nem

képezhető le a síkba torzításmentesen. Nézzünk erre néhány szép példát Klinghammer István [19] klaszikussá vált könyve alapján.



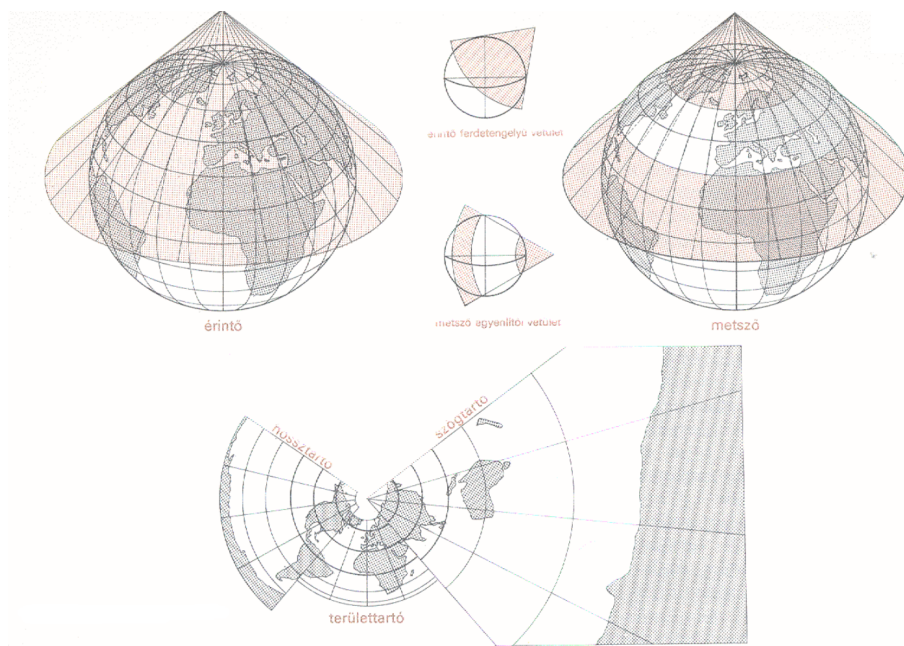
2.2. ábra. A sztereografikus (bal) és a gnomónikus (jobb) vetület (Klinghammer [19])



2.3. ábra. A hengervetületek (Klinghammer [19])

A 2.2., 2.3. és a 2.4. ábrákból látható, hogy nem lehetséges torzításmentesen vetíteni gömbfelületet síkra. Ezzel a problémakörrel a vetülettan foglalkozik, amelyet egy másik fejezet tárgyal. Feltételezzük, hogy aki térinformatikai rendszert készít, annak rendelkezésére áll valamilyen térkép, amely már síkra vetítve ábrázolja a földgömb valamely részletét. Ez alól csak a GPS mérési eredményei kivételek, amelyek földrajzi koordinátákban állnak elő (bár a síkba transzformált adatok is megkaphatók).

Amikor valamilyen térképről, vektorizálással készül digitális térkép, akkor egy már síkra vetített analog alapanyagból indulunk ki. Ha más koordináta rendszerben kívánjuk kalibrálni (georeferálni) a térképünket, mint amiben a papírtérkép készült, akkor is síkból síkba történő transzformációval van dolgunk. Abban az esetben pedig, amikor valamilyen digitális térképet importálunk, és állítunk elő belőle egy másik koordináta rendszerbeli adathalmazt, akkor is síkból síkba transzformálunk.



2.4. ábra. A kúpvetületek (Klinghammer [19])

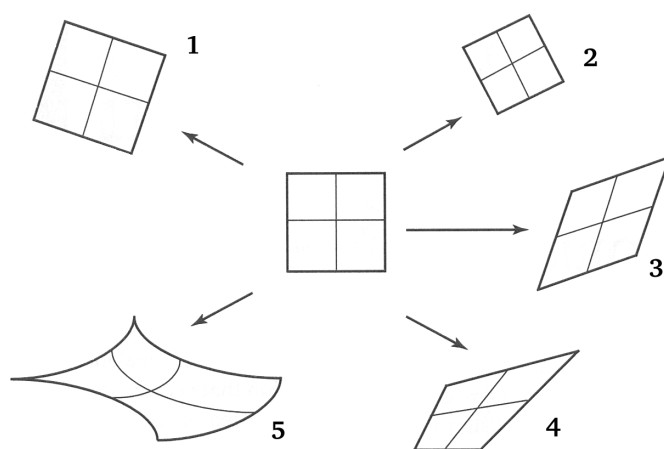
Úrfelvételek és légifotók esetében a sarokpont koordináták adják meg a képeket tartalmazó relatív koordinátájú rendszerből a valódi koordináta rendszer történő konverzió kalibrációs pontjait. A következő részben áttekintjük azokat a síkbeli transzformációkat, amelyek a leggyakrabban fordulnak elő a térinformatikában.

2.1. Síkbeli transzformációk

Miután sikerült rávetítenünk a síkra a gömbfelület valamely darabját, további koordináta transzformációk válhatnak szükségessé. Tegyük fel, hogy rendelkezünk egy papírtérképpel, amely valamilyen vetületi rendszerben készült. Feladatunk, hogy vektorizáljuk ezt a térképet. Vektorizáláskor választhatjuk ugyanazt a koordináta rendszert is, amelyben készült a térkép, de bármely másik rendszert is, ha a feladat további teendői ezt megkívánják.

Általában a szkenneléssel létrejött raszteres adatállomány relatív koordinátákban van (a bal felső sarok a koordináta rendszer origója, x és y irányú méretei a kép pixelben megadott mérete), aminek valódi koordináta rendszerbe transzformálása (georeferálás) szükséges.

Nézzük a 2.5. ábrát, amelyen a lehetséges síktranszformációkat vázoltuk, úgy, mint az egybevágósági, a hasonlósági, affin, projektív és topológikus transzformáció. Minden transzformációhoz tartoznak úgynevezett invariánsok, amelyek



2.5. ábra. A lehetséges síkbeli transzformációk: 1 – egybevágósági, 2 – hasonlósági, 3 – affin, 4 – projektív, 5 – topológikus transzformáció (Mortenson [27])

a transzformáció során nem változnak.

2.1.1. Egybevágósági transzformáció

Az egybevágósági transzformáció során az alakzatoknak sem a mérete, sem az alakja nem változik.

2.1.2. Hasonlósági transzformáció

A hasonlósági transzformáció során az alakzatoknak a mérete nem, de az alakja megőrződik. Konform transzformációnak is nevezik, mivel a megfelelő szögek nem változnak meg a transzformáció hatására (2.6. ábra).

A síkbeli hasonlósági transzformációt a következő egyenlettel írhatjuk le:

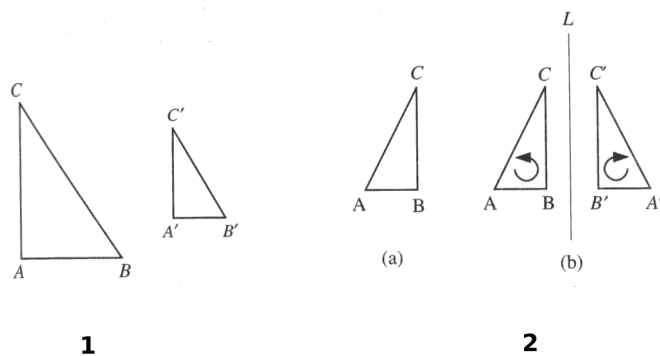
$$\begin{aligned} x' &= ax - by \\ y' &= \pm(bx + ay) \end{aligned} \quad (2.1)$$

ahol k a hasonlóság mértéke, és $\sqrt{a^2 + b^2} = k$, valamint

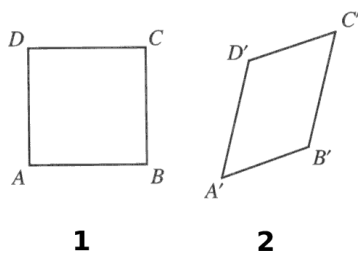
$$k \neq \begin{cases} k > 0 & \text{orientációt megőrző esetben} \\ k < 0 & \text{orientációt megfordító esetben} \end{cases}$$

2.1.3. Affin transzformáció

Az affin transzformáció során sem a méret, sem az alak nem őrződik meg, vagyis a párhuzamosság igen, de a bezárt szögek nem invariánsok (2.7. ábra).



2.6. ábra. Példák konform transzformációkra: 1 – az orientációt megőrző konform transzformáció, 2 – az orientációt megfordító konform transzformáció (a): eredeti alakzat, (b): transzformált alakzat)



2.7. ábra. Affin transzformáció: 1 – az eredeti alakzat, 2 – a transzformált alakzat. Az $A'B' \neq AB$, és az $\angle A' \neq \angle A$, viszont $A'D' \parallel B'C'$, továbbá AD minden transzformált pontja $A'D'$ -re esik

Az affin transzformációt leíró egyenletrendszer:

$$\begin{aligned}x' &= ax + by \\y' &= cx + dy\end{aligned}\tag{2.2}$$

ahol az együtthatókra igaz a következő feltétel:

$$\begin{vmatrix} a & b \\ c & d \end{vmatrix} \neq 0$$

Az affin transzformáció tulajdonságait összegzi a következő táblázat.

Transzformáció	Feltételek és együtthatók
Általános	$ad - bc \neq 0$
Azonosság	$a = 1, b = c = 0, d = 1$
Hasonlóság	$a^2 + c^2 = b^2 + d^2$ $ab + cd = 0$
Nyújtás	$a = k, b = c = 0, d = k$
Tükrözés	$a = -1, b = c = 0, d = -1$

Az affin transzformáció a leggyakoribb a térinformatikában. A georeferenciának nevezett művelet, amelyet például akkor végzünk, amikor egy szkenneléssel nyert, relatív koordinátákkal rendelkező raszteres adatbázist elhelyezünk egy földi koordináta rendszerben, voltaképpen egy affin transzformáció. A térinformatikai rendszerekben megtalálható koordináta rendszer átszámítási módszerek jelentős részben az affin transzformáción alapulnak.

Egyenes affin transzformációja

Amint fentebb láttuk, az (x, y) pont affin transzformációját a következő egyenlet adja meg:

$$\begin{aligned}x &= a'_{11}x' + a'_{12}y' \\y &= a'_{21}x' + a'_{22}y'\end{aligned}\tag{2.3}$$

Legyen az egyenes egyenlete:

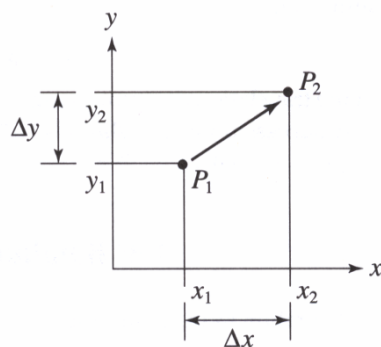
$$Ax + By + C = 0\tag{2.4}$$

Helyettesítsük be a 2.3 egyenletet a 2.4 egyenletbe. Némi átrendezés után a következőt kapjuk:

$$(Aa'_{11} + Ba'_{12})x' + (Aa'_{21} + Ba'_{22})y' + C = 0\tag{2.5}$$

azaz

$$A' = (Aa'_{11} + Ba'_{12})B' = (Aa'_{21} + Ba'_{22})C' = C\tag{2.6}$$



2.8. ábra. A P_1 pont eltolása P_2 -be, ahol az eltolás mértéke Δx és Δy

Az átalakítások után a 2.5 egyenlet a következő lesz:

$$A'x' + B'y' + C' = 0 \quad (2.7)$$

ami egy egyenes egyenlete, vagyis az egyenest egyenesbe viszi át az affin transzformáció. Hasonló megállapítás tehető magasabb fokú görbék esetére is.

A fenti transzformáció mátrix formában is megfogalmazható, amely sokkal áttekinthetőbb, tömörebb formalizmus. A 2.2 egyenlet mátrix formában a következő:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \quad (2.8)$$

amit egyszerűbben a következőként írható:

$$\mathbf{P}' = \mathbf{A}\mathbf{P},$$

tovább \mathbf{P} -re megoldva az egyenletrendszer:

$$\mathbf{P} = \mathbf{A}^{-1}\mathbf{P}',$$

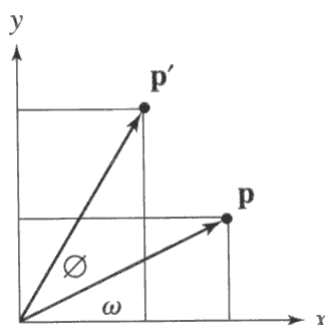
ahol \mathbf{A}^{-1} az \mathbf{A} mátrix inverze.

Eltolás

Legyen adva két pontunk a síkon, $P_1 = (x_1, y_1)$ és $P_2 = (x_2, y_2)$. Az a transzformáció, amely a P_1 pontot P_2 pontba tolja a következő:

$$\begin{aligned} x' &= x + \Delta x \\ y' &= y + \Delta y \end{aligned} \quad (2.9)$$

amely vektorosan felírva: $\mathbf{p}' = \mathbf{p} + \mathbf{t}$, ahol \mathbf{t} az eltolás vektora.

2.9. ábra. A p pont elforgatása Φ szöggel

Elforgatás

Forgassuk el a P pontot P' pontba Φ szöggel. A P' pontot adjuk meg a következő egyenletekkel:

$$\begin{aligned} x' &= |\mathbf{p}'| \cos(\omega + \Phi) \\ y' &= |\mathbf{p}'| \sin(\omega + \Phi) \end{aligned} \quad (2.10)$$

ahol $\cos\omega = x/|\mathbf{p}|$ és $\sin\omega = y/|\mathbf{p}|$.

Az ismert trigonometrikus azonosságok felhasználásával, amelyek

$$\begin{aligned} \cos(\omega + \Phi) &= \cos\omega \cos\Phi - \sin\omega \sin\Phi \\ \sin(\omega + \Phi) &= \sin\omega \cos\Phi + \cos\omega \sin\Phi \end{aligned} \quad (2.11)$$

felírható a következő egyenlet:

$$\begin{aligned} x' &= x \cos\Phi - y \sin\Phi \\ y' &= x \sin\Phi + y \cos\Phi \end{aligned} \quad (2.12)$$

ami mátrix formában a következő:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\Phi & -\sin\Phi \\ \sin\Phi & \cos\Phi \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \quad (2.13)$$

ami vektoros formában a következő: $\mathbf{P}' = \mathbf{R}_\Phi \mathbf{P}$, ahol \mathbf{R}_Φ a forgatási mátrix.

2.1.4. Projektív transzformáció

A projektív transzformáció során sem a méret, sem az alak, sem a párhuzamosság nem őrződik meg, de az egyenes vonalak egyenesek maradnak. Ez a transzformáció fajta ritkán fordul elő a térinformatikai gyakorlatban, mivel ritkán fordul elő olyan hiba, amely annyira szisztematikus, hogy helyreállítása a pro-

jektív transzformációt kívánna meg. Előfordul viszont a 3D-s problémák megjelenítési fázisában, ahol a perspektívikus ábrázolás során projektív transzformációra kerülhet sor.

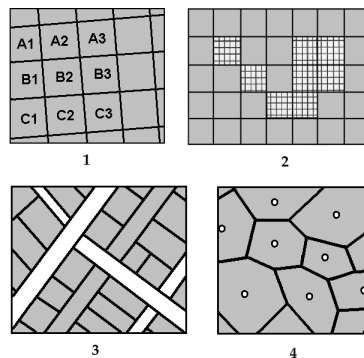
2.1.5. Topológikus transzformáció

A topológikus transzformáció során sem a szögek, sem a távolságok, sem a párhuzamosság nem maradnak meg, ellenben a folytonosság, a sorrendiség és a szomszédság megmarad. Ennek a transzformációnak fontos szerepe van a térinformatikában, azokban az esetekben, amikor az affin transzformáció nem hozott megfelelő eredményt, azaz a georeferencia a raszteres adatállomány szabálytalan deformációi miatt nem adott megfelelő eredményt.

Mikor fordulhat elő ilyen típusú deformáció? Ha például szkenneléssel állítottunk elő egy raszteres térképet, és a szkener papírtovábbító görgőin időnként elcsúszott a papír. Ilyenkor a pontos georeferálás sem ad megfelelő eredményt. Hiába jók a sarokpont koordináták, azok az ismert koordinátájú pontok, amelyek már rendelkezésünkre állnak, és amelyek a szkennelt térképen is felismerhetők, nem esnek megfelelő helyre. Ilyen esetekben a térképen helyről helyre változó paraméterű affin transzformáció válik szükségessé, amihez természetesen sok illesztőpont felvétele válik szükségessé.

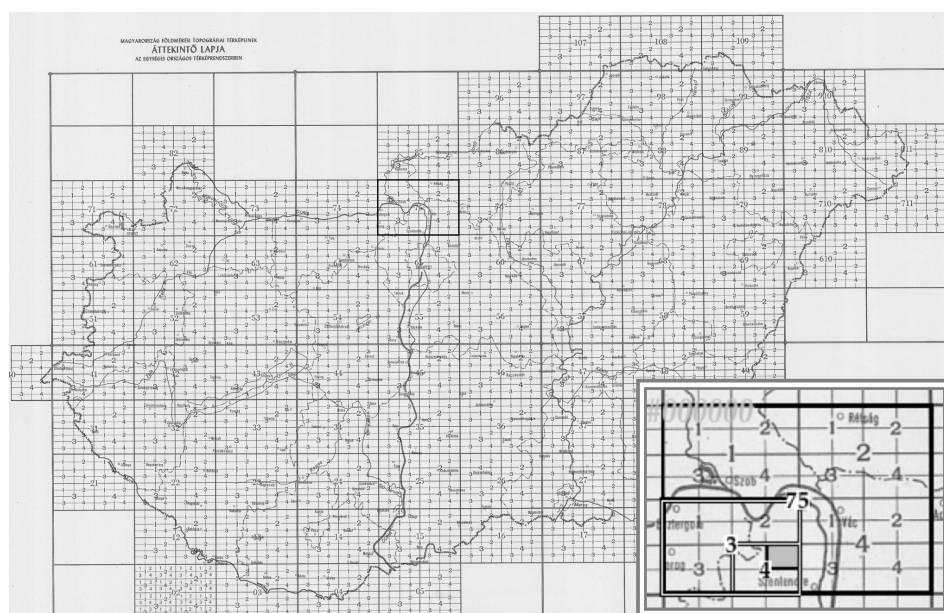
2.2. A tesszelláció

A sík lefedésének alapvetően kétféle módja létezik, a szabályos és szabálytalan tesszelláció. A 2.10. ábrán ezek alapesetei láthatók.



2.10. ábra. Szabályos (1,2) és szabálytalan (3,4) tesszelláció

A 2.10. ábra első esete a papíralapú térképek szelvényezési rendszere. A klasszikus, papír alapú térképészetben alapvető kérdés volt a célterület papíretérképekkel való lefedésének módja, vagyis a szelvényezés. Ennek során

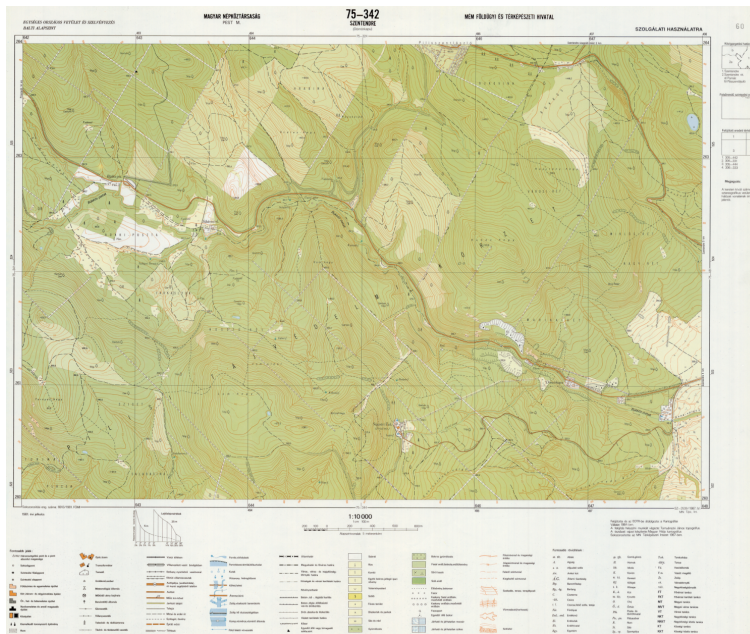


2.11. ábra. Az EOVS szelvényezés rendszere, jobb sarokban a 75-342 szelvéyszámú területtel

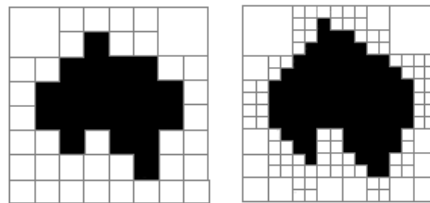
megtervezték, hogy mekkora térképlapokkal fedik le az ország területét, azokon milyen tematikus tartalom lesz, mennyi információ jeleníthető meg az adott papírméretben, mekkora lesz a térképek méretaránya, stb. Erre alapozva kialakultak a szabványos méretarányok, és tartalmak. A 2.11. ábrán látható egy szelvénybeosztás, és egy kinagyított részlete az ábra jobb alsó sarkában, valamint az ehhez tartozó térkép (2.12. ábra). Az ábrákon megfigyelhető, hogy az ország felszínét hézag és átfedésmentesen lefedik a hierarchiába szervezett, különböző méretű (méretarányú) szelvények. A térképszelvények számozásának rendje összefügg a hierarchiában elfoglalt helyzettel.

A 2.10. ábra második esete szintén a szabályos tesszellációt mutatja. Amikor szabályos rácsban tároljuk a felszín bizonyos adatait, mint például alapszínenkénti intenzitás értékek az űrfelvételek és légifotók esetében, vagy magasság értékek digitális domborzati modellek esetén, akkor a szabályosság a mintavételi rácsból származik. Ilyenkor egy mintavételi helyhez (pl. pixelhez) valamekkora területet rendelünk hozzá, és ezen a területen belül tekintjük érvényesnek a tárolt adatot (2.13. ábra). Ez tehát képek esetében pixeleket jelent, azonos intenzitás értékkel, vagy egy magassági lépcsőt állandó magasság értékkel.

A 2.10. ábra harmadik és negyedik esete a szabálytalan tesszellációra mutat példát. A harmadik olyan felszín lefedést mutat, ahol jogi szabályozás révén előállt hézag és átfedésmentes poligonok rendszere, (vagy valamilyen természettudományos ténymegállapítást mutat, mint például egy geológiai térkép,



2.12. ábra. A 75-342 számú térképszelvény



2.13. ábra. Ausztrália „térképe” különböző rácsállandójú tesszellációval ábrázolva

2.15. ábra), végül pedig a negyedik esetben egy matematikai, pontosabban geometriai algoritmus által előállított poligonrendszer fedi le a felszínt, szintén hézag és átfedésmentesen. Mindkét módszerről lesz szó. A harmadik esetről a „A vektoros adatmodell” című részben, míg a negyedikről a „3D grafika, felület modellezés” című fejezetben.

A 2.14. ábrán a kataszteri térkép tartalmának egy része látható (csak a földrészlet határok), amely nyilvánvalóan szabálytalan lefedést eredményez, hiszen a telekhatárok kialakításában, vagyis a jogi vonalak létrejöttét az emberi élet szabályozta.

A természetben előforduló képződmények, különösen a geológiában (2.15. ábra), gyakran eredményeznek hézag és átfedésmentes lefedést, amelyben sem a mintavétel szabályossága, sem pedig valamilyen tesszellációs algoritmus léte nem



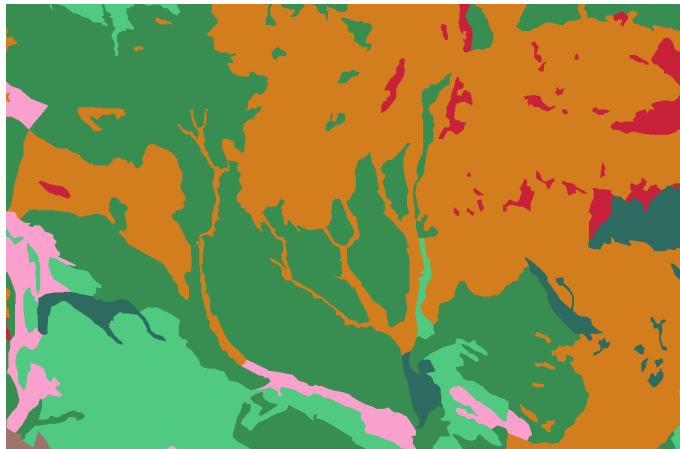
2.14. ábra. Példa szabálytalan tesszellációra, ahol jogi vonalak alkotta poligonok fedik le a felszínt

figyelhető meg.

2.3. Térképek és térbeli funkciók

A térképpel kapcsolatos elvárásaink összefoglalhatók néhány mondatban. A térkép a valóság egy lehetséges modellje, amely meghatározott szabályok szerint képezi le a föld felszínét, és annak tulajdonságait. A modelltől elvárjuk, hogy segítsen eligazodni, navigálni, azonosítani saját pozícionkat a térben, és helyzetünket viszonyítani más objektumokhoz. Egyrészt segítsen nagyobb területek áttekintésében, másrészt tartalmazzon minden olyan részletet, amire kíváncsiak vagyunk egy konkrét területen, egy bizonyos kontextusban. Legyen a segítségünkre abban, hogy valamely dolog, fizikai mennyiség, térbeli eloszlásának, minőségi és mennyiségi paramétereinek térbeli elterjedését ábrázoljuk, segítségével egy problémát megértsünk, és ha kell, döntést hozzunk. Sokféle objektum összehasonlítása, térbeli elterjedésének egybevetése által elemezni legyünk képesek ezen objektumok kapcsolatrendszerét, adott esetben kölcsönhatásukat. Máskor egyszerűen csak a dolgok számbavételét várjuk a modelltől, amely ebben az esetben egy térbeliséget is tartalmazó nyilvántartás.

Ezeknek a funkcióknak a klasszikus papírtérképek éppen úgy megfelelnek, mint a modern digitális térképek. Alapvető azonban a különbség abban, hogy a papírtérképek esetében a fentebb említett funkciók okán létrejövő eredmény minden esetben önálló térképműként testesül meg egy papírlapon, vagyis egyszer s mindenkorra változatlan formában jelenik meg. A digitális térképek esetében a folyamat eredménye dinamikus, vagyis gyorsan változtatható az eredmény, gyorsan igazítható a változó kérdésfeltevéshez. Permanens eredmény csak a pillanatnyi



2.15. ábra. Példa szabálytalan tesszellációra, ahol a felszín fizikai objektumai fedik le a felszínt (jelen esetben eltérő minőségű kőzetek)

állapotok nyomtatásával keletkezik.

A következőkben áttekintünk néhány alapvető térbeli műveletet, amelyek minden geoinformatikai rendszerben előfordulnak. Igaz, megjelenésük igen sokféle lehet, működésüket áthatják egy-egy szoftver alapkonceptiójából adódó eltérések, de alapvetően valamennyi szoftver ugyanazokat funkciócsoportokat valósítja meg.

2.3.1. Téma, tematika

A téma, vagy tematika változtathatósága minden GIS szoftver sajátja. A téma voltaképpen azonos struktúrájú objektumok halmaza. Megjelenési tulajdonságai csoportosan vagy egyénileg változtathatók meg. Témát alkotnak például a folyók, városok, megyék, amelyek, ha a grafikus attribútumaikat láthatóvá tesszük a számítógép monitorán vagy nyomtatón, akkor térképet alkotnak (2.16. ábra). Ilyenkor méretarányt és jelmagyarázatot is rendelhetünk hozzá.

Több tematika is elhelyezhető egy-egy térképen, mint pl. a 2.16. ábrán, ahol a megyéken kívül a nagy folyóink is láthatók.

2.3.2. Térbeli objektumok

Mint említettük, egy-egy téma azonos sémájú elemek halmaza. Ezek az elemek a térbeli objektumok¹. Egy objektum térbeliségét sokféleképpen fejezhetjük ki. Ábrázolhatjuk pontként, vonalként, felületként (2.17. ábra). Bizonyos esetekben nem tudjuk az objektumok pontos helyét, méretét (mocsaras terület, bokros, cserjés vidék, stb), sőt esetenként csak feliratok utalnak egyes területi egységek

¹Angolul geographic objects, csak hogy földrajzi objektumokként fordítva hasonlóan ügyetlen fordítást kapunk, mint amikor a GIS-t földrajzi információs rendszereknek fordítjuk



2.16. ábra. Példa a tematika fogalmára. A megyék és nagy folyóink, mint objektumok, egy-egy különálló tematikán helyezkednek el. Amint látható, a téma geometriájának megjelenítése feliratok, nevek nélkül nem eléggé informatív, noha mindannyian tudjuk, akik ennek az országnak a lakói vagyunk, hogy a folyók tematikán két objektum van, a Duna és a Tisza

létére (Jászság, Hortobágy, stb) anélkül, hogy pontosan, poligonokkal megadva lehatárolhatók lennének.

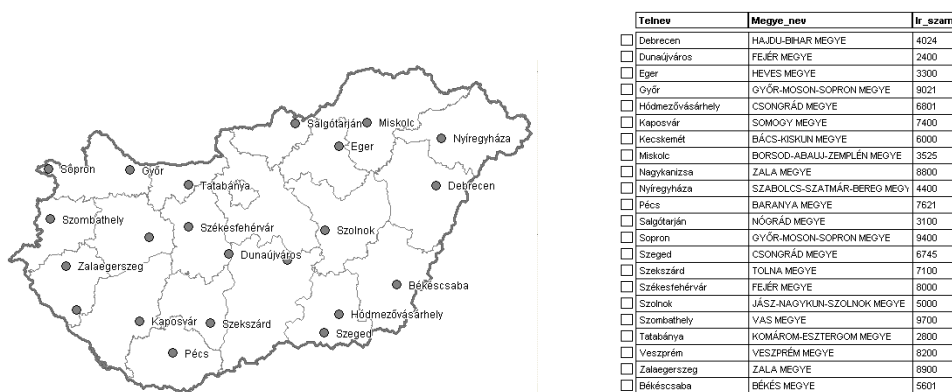
A térbeli objektumoknak két alapvető komponense van. Az egyik az attribútum adatok köre ((2.17. ábra, jobb oldali táblázat), amely leíró jellegű alfanumerikus adatot közöl az objektumról (pl. név, terület, súly, lakosság, szám, valamely kategória változó, stb.). A másik, a térbeliségre jellemző adategyüttes, amely megadja az objektum térbeli pozícióját (ha pont, akkor x, y, z koordinátát, ha vonal, akkor a vonalakat alkotó nodeok koordinátáit, stb.), valamint az objektumok egymáshoz való viszonyát (topológia). Ezek a viszonyok lehetnek adatok a szomszédságról, az objektumok kapcsolatrendszeréről, de akár hiányozhatnak is. Egy város belterülethatára ritkán szomszédos egy másikkal, általában a városok belterületét ábrázoló poligonok elszigetelt objektumok. Máskor viszont a poligonok minden határa közös valamely szomszédos poligonnal (pl. megyék, települések külterület határa, országhatárok).

Tekintettel a világ komplexitására, a modellül használt térbeli objektumoknak is hasonló összetettségűnek kell lenni. Célszerű megkülönböztetni elemi térbeli objektumokat (geometric primitives) és komplex térbeli objektumokat. Lássuk a következő absztrakt definíciót a tematikára:

$$\text{Tematika} = \{\text{Térbeli_objektumok_halmaza}\}$$

$$\{\text{Térbeli_objektumok}\} = (\text{leiras, elemi_terbeli_obj}) | (\text{leiras, komplex_terbeli_obj})$$

A fentiekből kilovasható, hogy a tematika, mint azonos strukturájú és típusú elemekből felépített halmaz leírására igen alkalmas a relációs adatbáziskezelők



2.17. ábra. Többféle téma, sokféle geometriai reprezentáció lehetséges. Jelen esetben ponttal jelöltük a nagyobb városokat, míg más esetekben poligonokkal is megmutatható a térbeli elhelyezkedésük (pl. belterület határ)

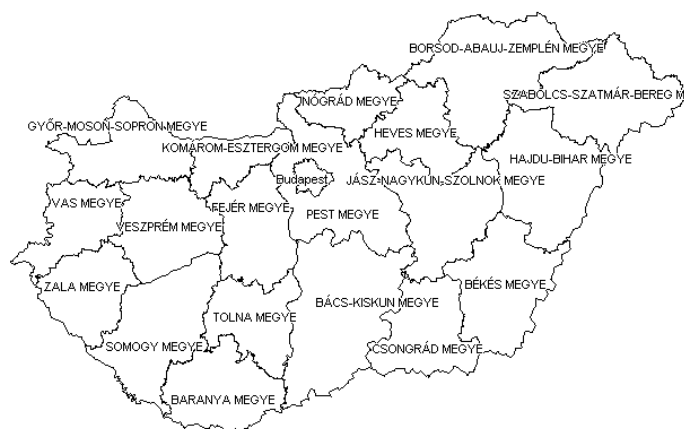
eszközrendszere.

2.3.3. Térbeli funkciók

Egyszerű megjelenítés. Olyankor alkalmazzuk, amikor csak az objektumaink térbeli elhelyezkedésének láttatása a cél. Kérdés persze, hogy mit ér egy olyan térkép, amelyen kizárólag geometriai elemek láthatók (pl. országhatárok országnevek nélkül, telkek helyrajzi szám feliratok nélkül, városok városnevek nélkül, mint pl. a 2.16. ábra, ahol sem megye-nevek, sem a folyó-nevek nem láthatók. Jogos persze az ellenvetés, hogy ekkora méretben vajnos lehet-e olvasható folyó-nevet írni egy térképre.

Adatok révetítése a térképre. A térbeli objektumok, komplex objektumok megjelenítése akkor igazán hatékony, ha valamely leíró adatát rávetítjük a térképre (pl. a megyék neveit, mint ahogy az a 2.18. ábrán látható). Ha rendelkezésre áll elegendő hely, akár többféle leíró adata is megjelenhet egy objektumhoz tartozóan. Itt most nem szelektáltunk a tematika objektumai között, valamennyit megjelenítettük, bár meg kell jegyezni, hogy a megjelenítő szoftver eleve szelektál, hiszen azok az objektumok nem láthatók, amelyek a viewporton (az aktuális ablakon) kívül helyezkednek el.

Tematikus tartalom révetítése a térképre. Másként is rávetíthetjük az adatokat a térképre. Ha az adattartalom és valamely geometriai tulajdonság (pl. szín, vonaltípus, szimbólum) között megfeleltetést hozunk létre, akkor előáll a tematikus térkép. Ha például egy adat kategória változó (pl. települések jogállása: főváros, megyei jogú város, község, stb.) akkor a megjelenítendő objektum olyan



2.18. ábra. Poligonok, amelyekre feliratként rávezettük az attributum adatok egyikét, a megye neveket

stílusban (színben, szimbólummal) jelenik meg, amelyet előzetesen az adott kategória változóhoz hozzárendeltünk. Hasonlóan lehet kategória változó például az utak rangsora: autópálya, autót, főútvonal, mellékút, földút, stb., amikor is a különböző kategóriájú vonalszakaszokhoz előre megadott szint, vonaltípust, vonalvastagságot rendelünk.

Nemcsak kategóriaváltozókat, hanem fizikai mennyiségeket (pl. lakosság szám, légszennyezés kvantitatív mértékei, stb.) is megjeleníthetünk tematikus térképként. Ebben az esetben előzőleg osztályoznunk kell a nyers adatokat. Meg kell adnunk, hogy a fizikai mennyiségeket hány előre megadott osztályba kívánjuk sorolni, mik lesznek az osztályok határai, és azok milyen stílusban (színnel, kiegészítő mintázattal, vonalvastagsággal, stb.) fognak megjelenni. Erre a tematikus térképfajtára mutat példát a 2.15. ábra. Itt a színkitöltés a közetminőséget hivatott jelezni.

Legyűjtés téma szerint. Gyakori funkció, hogy válogassuk le azokat az objektumokat valamely tematikából, amelyek megfelelnek valamely kritériumnak. A kritérium lehet egy egyszerű reláció (<, >, =, stb) valamely paraméter és egy leíró adatelem között, de lehet valamely összetettebb művelet eredményével való összehasonlítás is, amely akár több tematikát is figyelembe vehet a kritériumokkal való összehasonlítás során. Ebben a műveletben a térbeliség nem játszik szerepet.

Egyesítés attributum szerint. Két objektum egyesítésének kritériuma lehet valamely attributum adatok azonossága. Ilyenkor nemcsak az attributum adatokat kell egyesíteni, hanem az elemi térbeli objektumokat is (pl. poligonok egyesítése, ha a nevük azonos). Az egyesítés nem triviális abban az esetben, ha az attributum adatok között van olyan, amely az egyesítendő objektumok

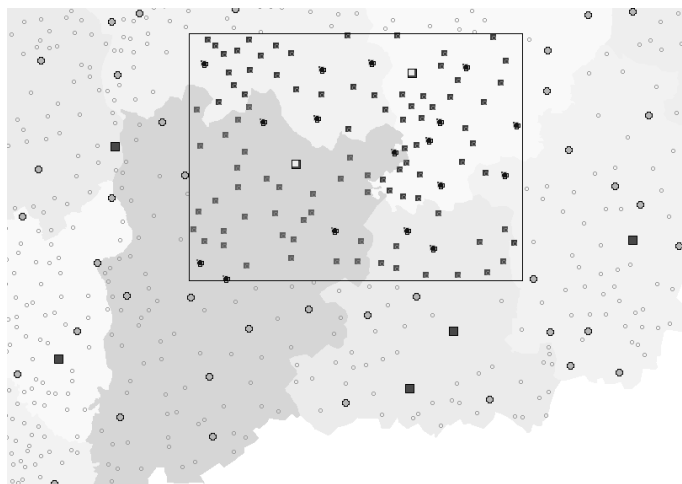


2.19. ábra. Példa poligonok egyesítésére. Az ábrán látható két tematika, a megye poligonok halmaza (vékony szürke vonallal), és a természetvédelmi igazgatóságok határait tartalmazó poligon halmaz (vastag szürke határvonalú poligonok). Ez utóbbi poligon halmaz többnyire a megye tematika egyes szomszédos objektumainak egyesítéséből származik. Mivel a valóságban mindig előfordulnak rendhagyó esetek is, így például Csongrád megye két igazgatósághoz is tartozik, a Tiszántúli a része az egyikhez, a másik része a másik igazgatósághoz

méretével, kiterjedésével arányos, vagy azok diszjunkt mivoltához kötött. Ilyen esetben az egyesítés utáni attribútum, az adat fizikai jelentésétől függően (extenzív vagy intenzív fizikai mennyiség), összeadódik, vagy valamilyen más aritmetikai művelettel számítható ki (pl. két különböző település poligon egyesítése után a hozzájuk tartozó lakosságszám összegződik, de ha az adat légnyomás, akkor kiegyenlítődik). A végzendő művelet az adat fizikai jelentésének ismeretében állapítható meg.

A 2.19. ábrán látható vastag határoló vonalú poligonok a természetvédelmi igazgatóságok határai, amelyek jogalkotási tevékenység által jönnek létre, mégpedig úgy, hogy egy-egy poligont megadott nevű megyék (vagy települések) alkotják meg. Ha tehát legyűjtjük a megadott nevű megyéket, akkor azok egyesítésével hozzuk létre a fent nevezett poligon halmazt. Azt egyelőre nem vizsgáljuk, hogy a geometriai elemekből miként jön létre az új poligon. A síkgráfokkal foglalkozó fejezetben (3.4.1) ezzel részleteiben is foglalkozunk.

Geometriai legyűjtés. Gyakori, és igen hasznos térbeli művelet a geometriai szelekció. Térbeli kritériumok alapján válogatunk le objektumokat valamely tematikából. A legegyszerűbb eset a rámutatás, amikor valamilyen grafikus szelekciós eszközzel pl. az egérrel rámutatva, kijelölünk egy objektumot. Pont esetén a pont hozzávetőleges helyére klikkelünk, vonal esetén, a vonal valamely pontjára, poligon esetében pedig a poligon belsejébe klikkelünk.

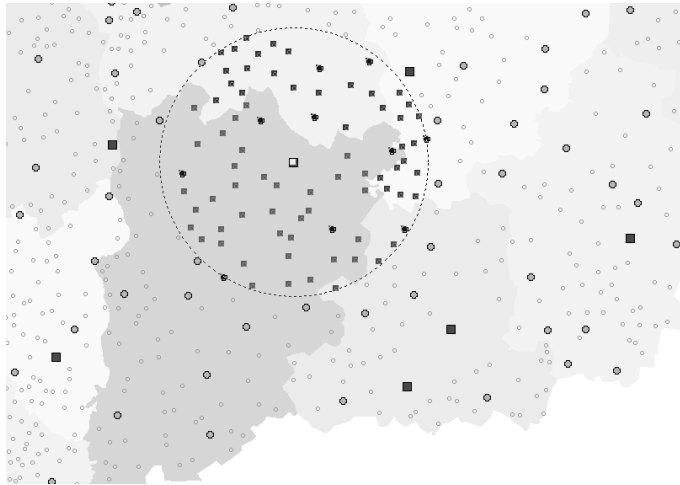


2.20. ábra. Kijelölés ablakkal: azon objektumokat szelektáljuk, amelyek beleesnek egy átmenetileg definiált ablak területébe

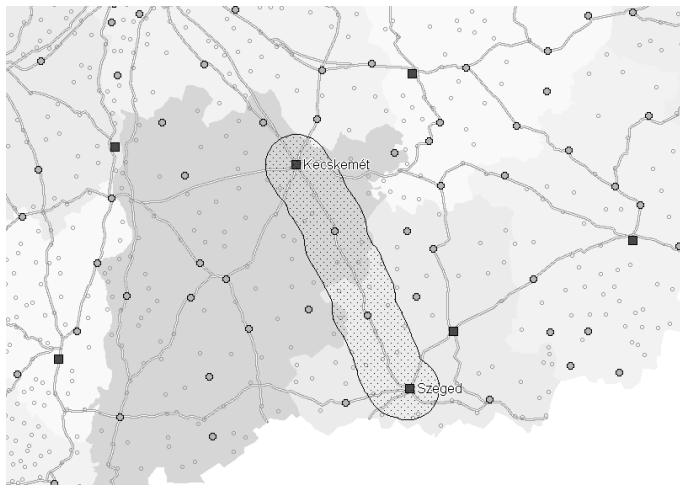
Leválogathatunk objektumokat nemcsak pontszerű szelekciós eszközzel, hanem felületi elemekkel való átfedés révén is. A legyűjtés kritériuma a felületi elem általi tartalmazás. Ilyen felületi elem lehet egy négyszög vagy ablakozás (2.20. ábra), vagy egy objektum meghatározott sugarú környezete (pl. egy pont, esetleg tetszőleges hely adott sugarú környezete. 2.21. ábra), vagy pedig egy objektum valamilyen sugarú környezete (2.22. ábra). További grafikus legyűjtő eszköz lehet valamilyen tematikából kijelölt poligon (2.23. ábra), amely azokat az objektumokat gyűjti le, amelyek részben vagy egészben beleesnek a szelekciós objektumba (intersect, overlap, contain, is within, near to, adjacent, stb.).

Valamilyen alfanumerikus, attributum adat is kombinálható térbeli legyűjtésekkel. Keressük például azokat a megyéket, ahol a népsűrűség nagyobb egy előre megadott küszöbértéknél. Ha településekhez kötött népességadataink vannak, akkor azokat előbb összegeznünk kell, de csak azokat, amelyek azonos megyébe esnek. Azután a megyénkénti összegeket el kell osztani az adott megye poligon területével, hogy megkapjuk a népsűrűséget.

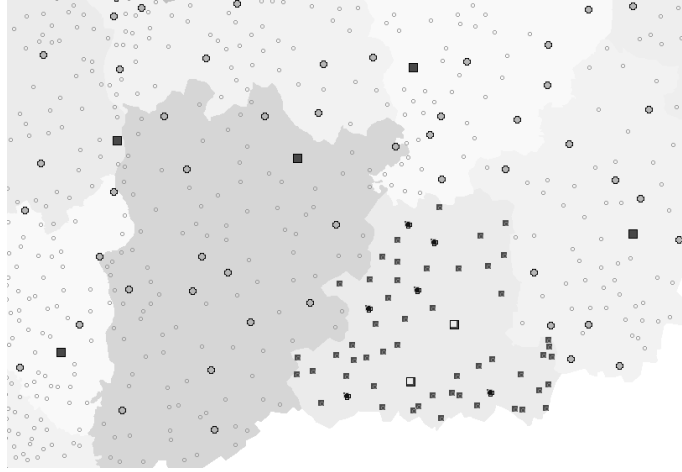
Egyesítés geometria alapján. Attributum adatoktól függetlenül is egyesíthetünk objektumokat. Önkényesen is kijelölhetünk egyesíteni kívánt objektumokat (akár nem szomszédosokat is), vagy valamilyen geometriai reláció alapján is kijelölhetjük az egyesítendő objektumokat. Figyeljük meg a 2.24. ábrát, amelyen látható, hogy hogyan lehet egy geometriai reláció (jelen esetben átfedés) alapján legyűjtött poligonokat egyesíteni. Keressük azokat a település külterület határokat, amelyeknek van közös része a Balaton nevű poligonnal. Ezek fogják alkotni a *part menti települések* nevű poligont, amely a burkolója lesz az előbbi feltételeknek eleget tevő poligonoknak.



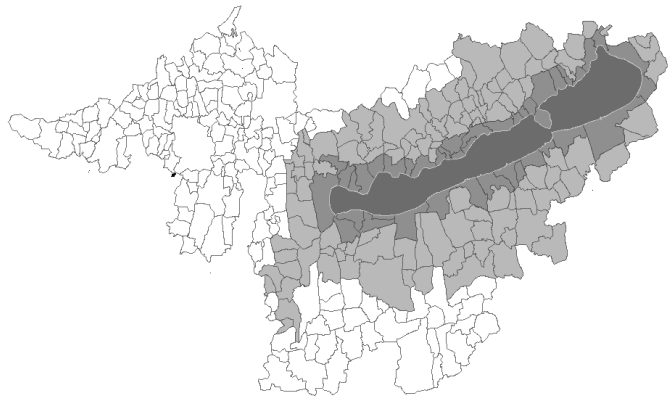
2.21. ábra. Kijelölés adott sugarú kör alakú környezettel: ebben az esetben vagy egy létező pontszerű objektum köré, vagy egy fiktív pont köré rajzolunk adott sugarú kört, és ezen belüli objektumokat szelektálunk



2.22. ábra. Kijelölés valamely tematikán lévő objektum adott sugarú környezetével: ha pont köré rajzolunk egy adott sugarú környezetet, akkor a szelektációs eszközünk egy kör alakú poligon lesz; ha vonalszegmens vagy vonallánc köré rajzolunk adott sugarú környezetet, akkor a szelektációs eszköz egy szabálytalan alakú poligon lesz; poligon adott sugarú környezetét kétféleképpen hozhatjuk létre: a poligon határvonalának adott sugarú környezete egy gyűrű alakú poligon lesz, vagy a gyűrű alakú poligon és az eredeti poligon uniója. Az ábrán az 5-ös út Kecskemét és Szeged közé eső szakaszának 10 km-es sugarú környezetét láthatjuk, mint poligont



2.23. ábra. Kijelölés poligon legyűjtéssel: egy létező tematika valamely poligonját tekintjük szelektációs eszköznek (jelen esetben Csongrád megy poligonsját), és az ebbe a poligonba eső objektumokat gyűjtjük le valamelyik tematikából (itt a ponttal szimbolizált településekből)



2.24. ábra. Egyesítés geometriai reláció alapján: a Balaton poligont átfedő település külterülethatár poligonok legyűjtése, majd a kollekció egyesítése egyetlen poligonná *part menti települések* néven. Jogalkotási folyamat alapján előállt listák alapján alkothatók meg az *üdülő övezet* (világos szürke poligonok) és a *vízgyűjtő* nevű poligonok (fehéren kitöltött poligonok)

Elő szeretnénk állítani az *üdülő övezet* nevű poligont. Erre már biztosan nem elegendő egy geometriai reláció (pl. legyen szomszédos a part menti településekkel), mert ez jogalkotási kérdés. Ilyenkor egy törvényben meghatározott településlista lehet a kiindulás alapja, amelyből szöveges legyűjtéssel állítjuk elő a kérdéses poligonokat, amelyek aztán egyesíthetők.

Számos hézag és átfedésmentes poligonstruktúra egyesítésével épülnek föl egyre komplexebb objektumok. Álljon itt példaként a következő: a települések belterülete megalkotható azoknak a poligonoknak az uniójából, melynek egyik minőségi jellemző kategória változója 'belterület' értékű. Ugyancsak megalkotható a külterületi ingatlanok és belterületi ingatlanok uniójából egy település külterület határa. A külterülethatárok uniójából a megye, azok uniójából az ország, az országok uniójából a kontinens, és így tovább ad infinitum.

Interpoláció, extrapoláció. Érdekes probléma megállapítani két mintavételi pont között valamely paraméter értékét. Számos szaktudományi mérési eredmény diszkrét pontokban keletkezik (pl. vízmérce adatok, szennyezési adatok), miközben ezen értékek folytonos eloszlásának ismerete nagyon hasznos lenne. Ilyenkor alkalmazunk interpolációt. Feltételezzük ugyanis, hogy az egyes mérési pontok között korreláció áll fenn, vagyis a mért értékek nem függetlenek egymástól. Előfordul, hogy olyan helyeken is szeretnénk megbecsülni egy paraméter értékét, amely már kívül esik a diszkrét mérési pontokat burkoló poligon területén. Ekkor alkalmazunk extrapolációt.

Az ideális interpoláció módszerét mutatja a Függelék mintavételezéssel foglalkozó fejezete (A.5. fejezet). Egy nem ideális, de gyakran alkalmazott interpolációs módszert, az inverse distance weighting módszert ismerteti a 3D-s domborzat modellezéssel foglalkozó rész (5.1.2. fejezet).

A problémakör egészével a geostatisztika foglalkozik, amelynek bemutatása már meghaladta volna e könyv terjedelmi korlátait.

3. fejezet

A vektoros adatmodell

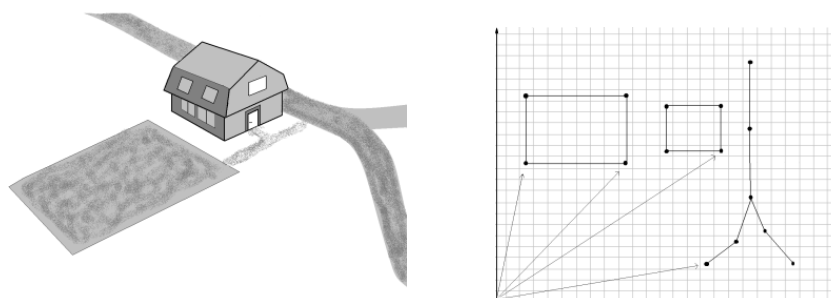
A vektoros adatmodell az ábrázolni kívánt objektumok általunk jellemzőnek, fontosnak tartott pontjainak helyvektorait tárolja, ami két dimenziós esetben x, y koordinátákat, három dimenziós esetben x, y, z koordinátákat jelent, továbbá azt is tárolnunk, tudnunk kell, hogy a megadott koordináták milyen koordináta rendszerben értelmezettek. Ez a tény feltételezi, hogy az a szoftver, amely tárolja a pontok koordinátáit, kezelni tudja a koordináta rendszereket.

Minden, a pontnál bonyolultabb objektum a helyvektorosan tárolt pontok sokaságából épül fel valamely összekötési szabály alapján (amely leegyszerűbb esetben lehet az összetartozó pontok beviteli sorrendje). A függelék gráfelméleti összefoglalójával összhangban, a vektoros adatmodellben az ábrázolni kívánt objektumokat gráfokkal írjuk le. A jellegzetes pontok a gráf pontjai (vertexei, nodejai), amelyek alkotják a vonalakat és a poligonokat.

Kérdés, hogy ki dönti azt el, hogy melyik pont jellegzetes illetve fontos, elvégre éppen ezeket akarjuk tárolni? A kérdés megválaszolásához vessünk egy pillantást a vektoros adatbevitel módszereire. Ha úrfelvételekről vagy légifotókról vektorizálunk, előbb értelmeznünk kell, hogy mit látunk. Ha felismertük a folyókat, utakat, házakat, akkor azok jellegzetes pontjai (ház sarkai, folyók, utak jellemző pontjai, úgy mint torkolat, útkereszteződés, stb.) egyszerűen megállapíthatók. A folyamat kiemelten fontos momentuma a felismerés, az emberi szemlélő, interpretátor tudása, tapasztalata, háttér ismeretei. Enélkül a vektorizálás lehetetlen.

A másik szempont, hogy milyen célra készül a vektortérkép. Az előre definiált célokból következtetni lehet arra, hogy mely objektumot tekintjük fontosnak, és melyet csak háttérnek, vagy éppen elhanyagolhatónak. Példaként vegyük azt az esetet, amikor közigazgatási célból készítünk vektoros térképet szkennelt papírtérkép alapanyagból. Ilyenkor minden közigazgatási vonatkozású adat fontos (település határ, megye határ, stb.), ellenben a folyók pontos ábrázolást nem igényelnek, noha jelenlétük szükséges. A patakok elhanyagolhatóak, nem is kell, hogy rákerüljenek a digitális térképre (még akkor sem, ha rajta vannak a papír alapanyagon).

Amikor papír térkép a vektorizálás nyersanyaga, akkor is igaz, hogy valamikor



3.1. ábra. A valóság egy kis darabja (bal oldali ábra), amelyet a vektoros adatmodellel fogunk leírni, és a vektoros adatmodell szerinti reprezentációja (jobb oldali ábra). A helyvektoros leírás – mint látható – elhelyezte az ábrázolni kívánt objektumok felszínre merőleges vetületeinek jellemző pontjait egy koordináta rendszerben, ennek következtében a vektoros adatok megjelenítésére hivatott grafikus ablakban felvette a konvencionális észak-déli irányítottságnak megfelelő pozíciót [8]

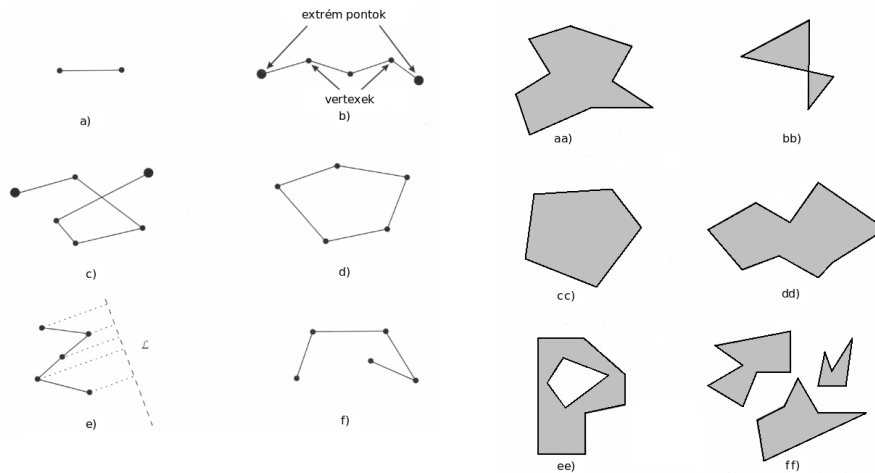
valakinek értelmeznie kellett a képet, valakinek valamilyen előismeretei alapján ki kellett tűzni mérési pontokat. Összefoglalva tehát elmondhatjuk, hogy a vektoros adatmodellt követő digitális térképekben a helyvektorokon és az összekötési szabályon kívül nagy mennyiségű emberi tudás is benne van.

A 3.1. ábrán látható egy idealizált kép a valóság egy szeletéről, amelyet vektorosan fogunk leírni. A valóságról alkotott modellünkhöz geometriai elemeket használunk, úgymint a nulla dimenziós pont, és az egy dimenziós vonal. Ennek speciális esete a vonallánc (polyline), amely egymáshoz kapcsolódó szakaszok sorozata (az összekapcsoltság már ebben az esetben is topológiai, vagyis a szomszédsággal kapcsolatos megszorításokat ró a modellre). A következő geometriai elem a (két dimenziós) poligon, amelyet pontok sorozata épít fel (3.2. ábra). Ennél precízebb megfogalmazás, hogy egy poligon egymáshoz kapcsolódó szakaszok sorozata, amelynek kezdő és végpontja azonos. Az egymáshoz kapcsoltság, valamint a zártság, a topológiára, vagyis a vonalak szomszédsági viszonyaira vonatkozó megszorítás.

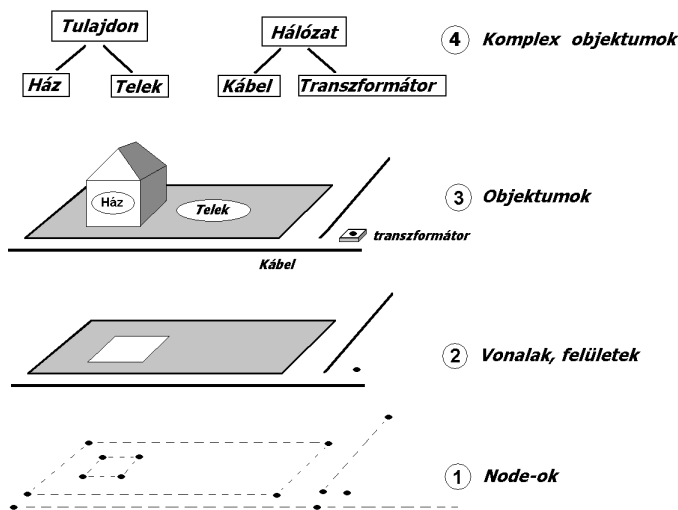
Amint a 3.3. ábrán látható, a legelemibb építőelem, csak a pont tartalmaz koordináta adatot. A többi objektum, a hierarchiában fejlebb álló vonal, vonallánc, és a poligon csak strukturális információt tartalmaz arról, hogy a magasabb hierarchiájú objektum milyen viszonyban van a hierarchiában alatta lévővel.

Ez egyben azt is jelenti, hogy a pont és koordinátáinak tárolása nem elegendő a geometria vektoros leírásához, hanem azt is tárolni kell, hogy a hierarchiában a pontok felett álló objektumok miként jönnek létre a pontokból alkotott szakaszok, és azok esetleges kapcsolódása által.

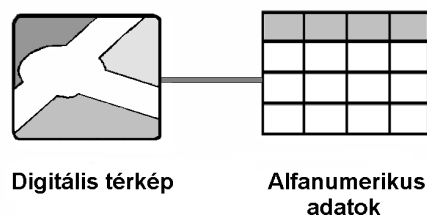
A vektoros objektumleírásnak számos előnye van, mint például:



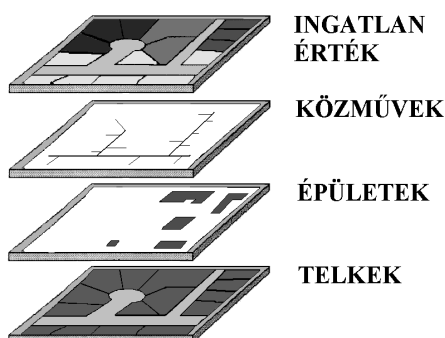
3.2. ábra. Példák egydimenziós objektumokra (bal oldali ábra): a) vonal szegmens (szakasz); b) vonallánc (polyline) végpontokkal és vertexekkel; c) nem kereszteződő vonallánc; d) zárt polyline; e) monoton polyline; f) nem monoton polyline. Példák kétdimenziós objektumokra (jobb oldali ábra): aa) egyszerű poligon; bb) nem egyszerű poligon (ilyen poligon szinte csak hibaként fordul elő a geoinformatikában); cc) konvex poligon; dd) monoton poligon; ee) poligon sziget struktúra (lyukas poligon); diszjunkt poligonokból álló régió [31]



3.3. ábra. A vektoros rendszerek elemeiből (geometric primitives) egyre komplexebb objektumok, objektumcsoportok alkothatók: 1 – a nodeok szintje; 2 – vonalak, poligonok szintje; 3 – objektumok szintje; 4 – komplex objektumok szintje [8]



3.4. ábra. A grafikus és a leíró adatok kapcsolatának sematikus ábrája. A grafikus objektumok és a leíró adataik 1:1 típusú kapcsolatban vannak egymással [8]

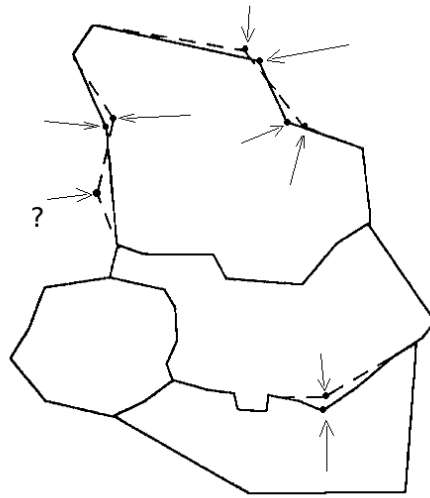


3.5. ábra. Objektumcsoportok létrehozásával rugalmasan használható adatszerkezet hozható létre. Az objektumorientált megközelítésű rendszerek ezeket *feature class*oknak hívja (tulajdonság osztályok), a hagyományos, CAD terminológiát előnyben részesítő szoftver gyártók pedig inkább a *layer* elnevezést használják [8]

- Csak ott van pont (vertex), ahol változás van a geometriában
- Helytakarékos grafika leírás
- Objektum hierarchia felépítésének lehetősége (3.3. ábra)
- Korlátlan adatbázis kapcsolati lehetőségek (3.4. ábra)

Az adatbázis kapcsolatot a 3.4. ábra mutatja sematikusán. A geometriai hierarchia bármely szintjéhez rendelhetünk attributum adatokat leíró táblákat, ha azt a problémakör üzleti logikája (business logic) megkívánja.

A vektorosan tárolt adatokat érdemes logikai csoportokba szedve kezelni, attól függően, hogy milyen csoportosítást részesítünk előnyben (3.5. ábra). Egy lehetőség az önkényes csoportosítás, amely a tervező, rendszerépítő deklarációjától függ. Ő dönti el, hogy mely objektumok kerüljenek egy objektum csoportba (feature class, layer). Ebben az esetben igen tág tere nyílik a tervezőnek, de egyben megnő a felelősége is.



3.6. ábra. A spagetti modellben előforduló hibák egyike. Az ábrán a folytonos és a szaggatott vonalas poligonoknak egymáson kellene futniuk, de amint a nyilak mutatják, ez nem teljesül, sőt a ? jellel jelölt nyíl olyan nodeot mutat a szaggatott poligonban, amelynek nincs is megfelelője a folytonosban [39]

Egy másik lehetőség az objektumok valamilyen tulajdonság azonosság alapján történő beorolásán, osztályozásán alapuló csoportosítás. A csoportosítás lehet fix, vagyis nem változtatható, de lehet rugalmasan változtatható is, ami a pillanatnyi legyűjtési szempontrendszer alapján végzett csoportosítás.

A vektoros adatokat nemcsak koordináták szerint, hanem egymáshoz képest is megadhatjuk, sőt célszerű a szomszédsági, a kapcsolódási viszonyokat is figyelembe venni. Számos hétköznapi eset mutat „topológikus” megfontolásokat: „Hogyan találok oda a pályaudvarra?” kérdezi az eltévedt turista. „Ezen az utcán menjen végig a második kereszteződésig, ott forduljon jobbra, menjen addig, amíg az út nem keresztesz egy vasúti átkelőt, majd a sín mellett haladva elérí az állomást.” Egy másik, topológiát rejtő kérdés: „Mi van a szomszédom telke és az én telkem között?” Semmi, csak a kerítés (a kerítés nem telek, hanem egy egydimenziós objektum), mert ha lenne, akkor nem volnánk szomszédok.

A topológia tehát leírja az objektumok szomszédsági viszonyait, az objektumok kapcsolatrendszerét (átfedések, szakadások).

3.1. Spagetti modell

A spagetti modell egy olyan vektoros adatmodell, amely a nodeokon és az összekötési szabályon kívül mást nem vesz figyelembe. Nem vizsgálja, hogy van-e

Geometriai elemek			
ID	id	xcoord	ycoord
1	1	639933,9375	232427,797
2	1	639957	232447,828
3	1	639973,75	232429
4	1	639949,625	232409,9845
5	1	639933,9375	232427,797
6	2	639957	232447,828
7	2	639933,9375	232427,797
8	2	639919,9375	232443,625
9	2	639943,0625	232463,297
10	2	639951	232454,5625
11	2	639957	232447,828
12	3	639943,0625	232463,297
13	3	639919,9375	232443,625
14	3	639904,3125	232460,953
15	3	639927,9375	232480,703
16	3	639943,0625	232463,297
17	4	639940,125	232490,7185

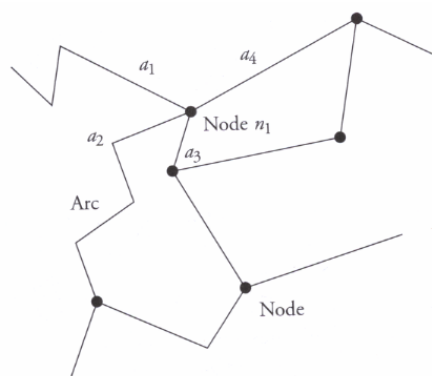
Objektumok		
id	Name	Field1
1	Region 1	5
2	Region 2	6
3	Region 3	5
4	Region 4	6
5	Region 5	6
6	Region 6	9
7	Region 7	6
8	Region 8	21
9	Region 9	5
10	Region 10	5
11	Region 11	7
12	Region 12	6
13	Region 13	6
14	Region 14	6
15	Region 15	6
16	Region 16	5
17	Region 17	6
18	Region 18	6
19	Region 19	6

3.7. ábra. A nodeok és a belőlük alkotott poligonok relációs táblák kapcsolatával leírhatók. Ez az egyszerű szerkezet azonban nem tartalmaz semmilyen információt a topológiáról [8]

közvetlen szomszédja egy poligonnak, és így vannak-e közös nodjai a szomszédos poligonoknak. Nem törődik a folytonossággal, és az egyes objektumok esetleges térbeli sorrendiségével, szomszédságával (3.6. ábra).

A legtöbb általános célú, vektoros rajzoló spagetti modellt követ. Előnye az egyszerűsége, amely egyben a hátránya is, mivel olyan laza minőségi kritériumok mellett, mint amelyet a spagetti modell kíván meg, nagyon könnyű rossz minőségű adatbázist létrehozni. Az egyszerűség persze nem zárja ki, hogy hozzáértők a spagettiből is professzionális minőséget hozzanak létre, de széleskörű használata térinformatikai rendszerekben kerülendő. Az általános rajzoló programok alkalmazzák a spagetti modellt, és saját adatformátumot használnak, amely (többnyire) nem tartalmaz relációs adatbázisok elméletére épülő megfontolásokat.

Meglepő, hogy egyes térinformatikai szoftverek a fentiek ellenére alkalmazzák a spagetti modellt. Amikor poligonok zártságát, vonalláncok szakadás mentességét kívánjuk ellenőrizni, akkor azt a szoftverek külön utasításra megteszik, de az adatmodell nem tartalmaz erre vonatkozó garanciát, azt a node koordináták összehasonlításával lehet megtenni. A 3.7. ábrán a spagetti modell egy lehetséges megvalósítása látható. A *Geometriai elemek* nevű tábla a nodeok azonosítóit (elsődleges kulcsait) és a koordinátáit tartalmazza. Az *Objektumok* tábla az objektumok azonosítóit (elsődleges kulcsait), és az objektumokra vonatkozó leíró adatokat tartalmazza, mint például *Name*, *Field1*. Figyeljük meg, hogy a nodeok sorrendje fontos, mivel csak növekvő *id* szerint kirajzolva eredményez poligont. Szomszédos poligonok topológikusan helyes tárolását lehetővé teszi ugyan ez a struktúra, de nem garatálja. Ebben a tárolási szerkezetben külön gondoskodni kell (szoftveresen, emberi odafigyeléssel) a korrekt topológiáról.



3.8. ábra. Nodeok, polylineok, arcok (ívek) alkotják a hálózatokat [31]

3.2. Hálózat modell

A hálózat modell arra hivatott, hogy vonalak hálózatával írjunk le hálózatos működésű dolgokat, úgy mint közlekedés szervezés, szállítmányozás, logisztikai problémák, közmű hálózatok, kapcsolatrendszerek, stb. Minden esetben arról van szó, hogy gráfokkal modellezzük az üzleti logika által definiált csomópontokból felépülő hálózatokat (3.8. ábra). Megkülönböztetünk arcot (ívet) és vonalláncot (polylinet).

Az ábráról látható, hogy célszerű lehet, különösen útvonal optimalizációs problémák esetében, a sok nodeból álló vonalláncokat arcoknak nevezve másként kezelni, elvégre ezeken a polyline szakaszokon nincs elágazás. Az optimális útvonalak megkeresésre mutat egy példát a gráfelmélettel foglalkozó fejezet az **A. Függelékben** (129. oldal), ezért az útvonal optimalizálás bemutatását itt mellőzzük.

3.2.1. Lineáris referencia

A helymeghatározás többféle módon is lehetséges. A jól ismert x, y, z koordináták megadása mellett a postai cím alapján történő helymeghatározás is régóta alkalmazott módszer. (Ha valamelyik ismerősünk a címe helyett a lakhelye koordinátáit adná meg, meglepődnénk, mert bizonyos problémakörben a postai cím a helymeghatározás elfogadott rendje.) További helymeghatározási mód is lehetséges.

A vonalas létesítmények, mint a közmű- és közlekedési hálózatok (út, vasút) valamely pontjának térbeli leírására használjuk a következő módszert, amely a hálózati modell egyik igen általános felhasználása. Vegyük például az utak mentén lévő kilométerköveket, vagy a folyók mentén a folyamkilométereket. Ezek a helyek pontos koordinátákkal rendelkeznek. Számos esetben azonban a helymeghatározás egy-egy ilyen mérföldkőhöz képest történik (pl. a 123 folyamkilométertől 150 méterre folyásirányban). Utakon történt események (pl. balesé-

tek) helyének megállapítására sem használjuk az abszolút koordinátákat, hanem valamilyen relatív meghatározást adunk, mint pl. baleset történt a 4-es út 52. é 53-as kilométere között, félúton. A folyamkilométerek, kilométerkövek, mint szelvénybeosztások, akár pontoknak is tekinthetők.

A hálózatot reprezentáló vonalláncokat alkotó nodeok azonban egyáltalán nem biztos, hogy egybeesnek a szelvénybeosztás pontjaival (képzeljünk el 10 km hosszú nyílegyenes vasúti vonalszakaszt, amelynek egy kezdő- és egy végnodeja van. Teljesen felesleges lenne oda is nodeot tenni, ahol nincs változás a geometriában. A szelvénybeosztás kilométerenként tartalmaz egy pontot, így akár ez alapján generálhatnánk vonalláncokat. Ez sem lenne jó megoldás, mert a geometria szempontjából mindenhogyan redundáns nodeok tömegét hoznánk létre.

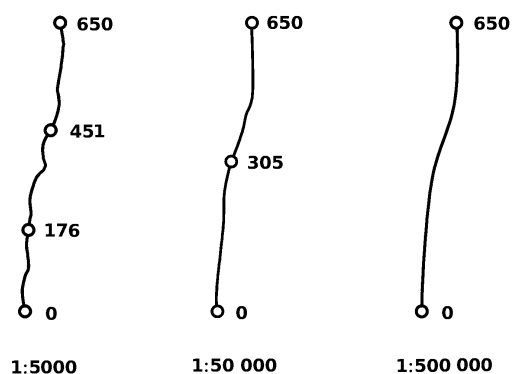
Tovább bonyolítja a helyzetet, hogy előfordulhatnak változások a geometriában (pl. nagyobb, többszintes csomópontot építenek ki több útvonal találkozásában, vagy véglegesen elterelik a folyót egy gát, erőmű megépülése miatt). Ilyenkor akár jelentősebb hosszváltozás is bekövetkezhet, ami ha a vonalas logikát követjük, valamennyi kilométerszelvény áthelyezését igényelné. Ez nyilván képtelenség.

Végül is mindegy, hogy milyen módszerrel határozunk meg azonosítani kívánt helyeket, ha ezek egymással összhangban vannak, és azonos eredményre vezetnek. Szakterületenként eltérhetnek ezek az azonosítási módszerek. Lássunk ezek közül néhányat:

- földrajzi koordináták
- útszelvényszám
- szelvényazonosítóhoz képesti távolság
- városhatártól mért távolság
- kilométerkövektől mért távolság
- kereszteződésektől mért távolság
- nevezetes helytől mért távolság

A különböző szakterületek az üzleti logikából következően sokszor elétőző méretarányú térképről származó geometriai alapra építették az információs rendszerüket. A különböző méretarány, a sokféle üzleti logika, a szelvénybeosztás problémakörét mutatja a 3.9. ábra.

A 3.10. ábrán látható vonallánc leírható a klasszikus georeferencia által, az n_1, n_2, n_3, n_4 nodeok koordinátaival (3.1. táblázat), és ugyanazon helyek lineáris referencia (szelvényezés) szerinti leírásával (pl. az adott hely távolsága a kezdő szelvény számtól, 3.2. táblázat).



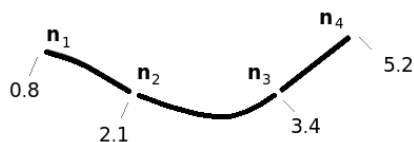
3.9. ábra. A különböző mérearányból, és az eltérő üzleti logikából származó lineáris referencia alapú megközelítés sematikusán ábrázolva

3.1. táblázat. Az n_1, n_2, n_3, n_4 nodeok georeferencia táblázata

n_id	x_coord	y_coord
n_1	625212	385261
n_2	625230	385286
n_3	625248	385266
n_4	625267	385274

3.2. táblázat. Az n_1, n_2, n_3, n_4 nodeok a lineáris referencia (szelvényezés) szerinti táblázata

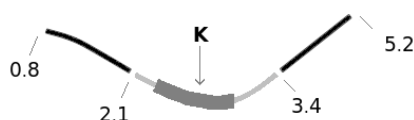
n_id	távolság
n_1	0.8
n_1	2.1
n_1	3.4
n_1	5.2



3.10. ábra. Egy vonalláncot alkotó nodeok (n_1, n_2, n_3, n_4), és a szelvényezés logikája szerinti ugyanazon helyek

3.3. táblázat. A **K** jelű szegmens meghatározása a lineáris referencia szerint

id	vonali_id	start	end
K	67	2.2	3.1



3.11. ábra. Egy vonallánc, amelyen láthatók a szelvényezés helyei, és egy olyan **K** jelű vonalszakasz, amelyet a 3.3. számú lineáris referencia táblázatban definiáltunk

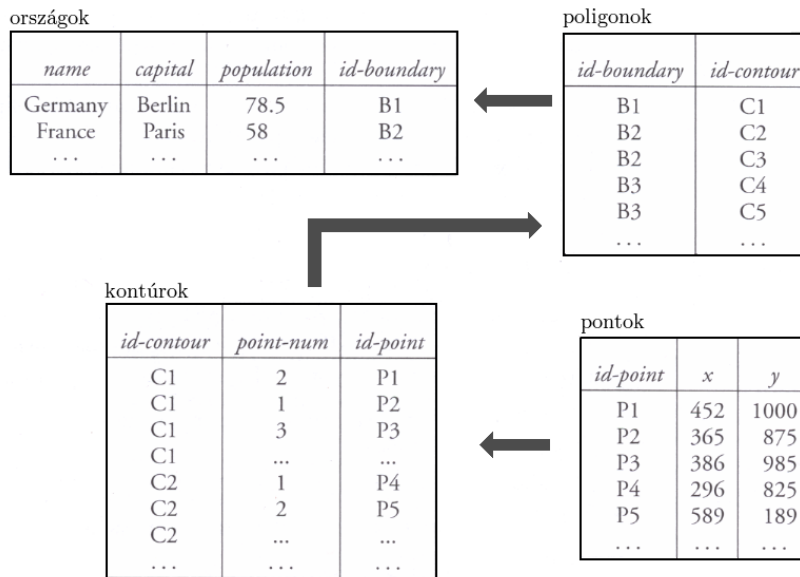
3.2.2. Dinamikus szegmentálás

A rétegszemléletű megközelítésben a grafikus adatok szegmentáltsága, valamely üzleti logika szerinti szakaszoltsága a grafikus adatbázis permanens része. Az objektumorientált szemlélet előretörésével a grafikus adatok valamely (tetszőleges) attribútum szerinti dinamikus szegmentálásának lehetősége rugalmas rendszerépítést tesz lehetővé. Különösen érdekes ez a lineáris referencia szerinti logikájú rendszerekben, ahol a szegmensek megadása a lineáris referencia szerint valósítható meg (3.3. táblázat és 3.11. ábra).

3.3. Relációs modell

A klasszikus vektoros adatmodell megvalósítására kézenfekvő, hogy valamilyen relációs adat reprezentációt alkalmazzunk. A spagetti modell esetében láttuk, hogy a relációs adatrepresentációkkal jól leírhatók poligon struktúrák. Nézzük most meg, hogy az előbbi szerkezetet hogyan finomíthatnánk úgy, hogy a korrekt topológia automatikus biztosítsék.

Konkretizáljuk az előbbi példát. Írjuk le Európa fővárosait úgy térbelileg, mint attribútum adatokat tekintve. A 3.12. ábrán látható négy adattábla, amelyek közül az elemi geometriát, a *pontok* tábla tartalmazza. A *kontúrok* tábla ezekből a pontokból épül fel. A *poligonok* tábla a kontúrokból épül fel. A kontúrok tábla annyi vonalláncból áll, ahány határvonala (szomszédos poligonja) van a poligonnak (felületnek). Végül az *országok* tábla a *poligonok* tábla azonosítóira hivatkozik, mint geometriai összetevőre, a többi adata viszont leíró tartalmú (*name*, *capital*, *population*). Vegyük észre, hogy ebben az esetben egy node megváltoztatása (pl. eltolása) nem rontja el a topológiát, ha eredetileg létezett. Egy ilyen nagy hatékonyságú struktúra már képes biztosítani topológikus tulajdonságokat, bár en-

3.12. ábra. *Pontok, kontúrok, poligonok és országok* nevű relációs adattáblák [31]

nek ellenére mégis relációs modell néven nevezzük.

Példaképpen adjunk meg egy SQL parancsot, amely a táblákból kiválaszthatja Franciaország kontúrjait:

```
SELECT poligonok.id-contour
FROM országok, poligonok, kontúrok, pontok WHERE name='France'
AND országok.id-boundary=poligonok.id-boundary
AND poligonok.id-contour=kontúrok.id-contour
AND kontúrok.id-point=pontok.id-point
ORDER BY poligonok.id-contour, point-num;
```

3.4. Topológikus modell

A topológikus tárolás elemei – az eddiekhez hasonlóan – poligonok, vonalak és pontok. A kérdés az, hogy hogyan tároljuk ezeket az objektumokat úgy, hogy a szomszédsági viszonyok eleve bele legyenek építve az adatszerkezetbe. A relációs modellben a helyes topológiáról magunknak kell gondoskodnunk, az magától nem teljesül (természetesen szoftveres megoldások lehetnek, amelyek arra kényszerítik a felhasználót, hogy ne vétson topológiai hibát).

A topológikus adatszerkezetnek a következőkre kell képesnek lennie:

- tárolja a geometriai elemek topológiai viszonyait

- ne engedjen meg hibás eseteket, azaz a poligonok közt ne legyenek rések (hacsak nem olyan természetűek, mint például városhatárok), azok ne metsszék egymást, stb.
- olyan eseteket is tároljon, amikor egy poligonban egy másik poligon benne van, ugyanis ilyen esetek a térképeken előfordulnak
- tárolja a geometriai entitások attribútumait: egy poligon milyen színű, egy vonal szaggatott-e, stb.
- a tárolt topológiával kapcsolatos kérdésekre könnyen válaszoljon: egy pont hol van, egy polyline mit metsz, egy poligon mivel szomszédos, milyen másik poligonokat tartalmaz, stb.

Mivel a geometriai objektumok típusa pont, vonallánc és poligon lehet, ezek mind külön térképi rétegek leírására használhatók. Amint láttuk, pontok esetében elegendő azok azonosítóit és helyét (koordinátáit) tárolnunk. A pontok nemcsak összetettebb geometriai objektumok elemei lehetnek, hanem önálló pontszerű entitásai egy információs rendszernek. Ilyen esetekben a pontok grafikai megjelenése tematikus tartalmat is hordozhat. A grafikus megjelenés legtöbbször valamilyen egyezményes grafikus jellel történik, amelyet valamely jelkészletből, ábrázolási konvenció gyűjteményből választunk ki (jelkulcs).

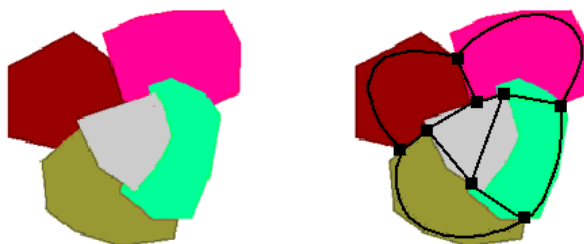
A vonalláncokat (polylineok) célszerű gráfokkal leírni, ahol a vertexek a vonalláncokat alkotó pontok, az élek pedig egy-egy node összekötöttségét reprezentálják. Ezekhez is rendelhetők a megjelenítésre vonatkozó grafikus stílusok (vastag vonal, szaggatott vonal, több párhuzamos vonalból álló egység), de ezek nem érintik a vonalláncokat alkotó nodeok elhelyezkedését.

A poligon réteget egy speciális összetett gráfstruktúrával ábrázoljuk, amelyet síkgráfnak neveznek. A síkgráf háromféle elemtípusból áll: terület, él és elágazási pont. A terület reprezentálja a poligonok területeit, az élek pedig a területeket elválasztó polylineokat, az elágazási pontok pedig a polylineok találkozási pontjai lesznek (hasonlóan a 3.12. ábrán látható táblákhoz). A gráfelmélet alapfogalmait az **A. Függelék** ismerteti (129. oldal).

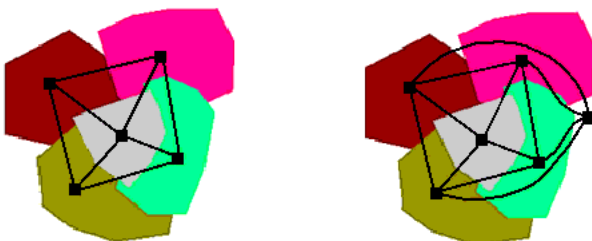
3.4.1. Síkgráfok

A síkbeli struktúrát valójában két gráfban tároljuk. Az egyik gráf csúcsai az elágazási pontok, ezek között a pontok között élek pedig ott lesznek, ahol ezen a pontok között polylineok vannak. Ezekben az élekben tároljuk azt is, hogy tőle jobbra és balra milyen területek vannak. Nevezzük ezt a gráfot síkgráfnak (3.13. ábra).

A másik gráf a síkgráf duálisa, ebben a csúcsok a területek, az élek pedig azt jelentik, hogy egy területnek egy másik a szomszédja. A területhez tároljuk el az őt határoló polylineokat is, egyszóval azt a valódi poligont, amit az reprezentál.



3.13. ábra. Példa egy síkgráfra. Az ábra bal oldalán a poligonok, az ábra jobb oldalán a területek által meghatározott síkgráf látható [11]



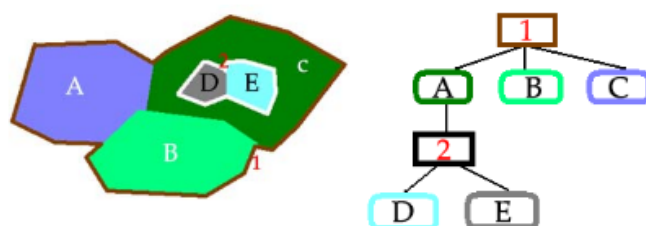
3.14. ábra. A 3.13. ábrán lévő síkgráf duális gráfja, és a duális gráf a befoglaló területekkel kiegészítve [11]

A duális gráf szerepe, hogy a síkgráffal együtt már leírja a poligonok közti topológiai viszonyokat, és azok geometriáját is. Szomszédságra, elérhetőségre vonatkozó kérdésekre így már könnyen válaszolhatunk (ezzel a struktúrával azonban még nem tudunk egymásba ágyazott területeket tárolni).

Definiáljunk ezért a valós területeken kívül egy új típust, a befoglaló területeket. Ezek a valóságban külön poligonként nem megjelenő területek. Legkönnyebb úgy gondolni rájuk, hogy egy befoglaló terület tulajdonképpen egy olyan poligonhalmaz kontúrja, amelynek minden elemét el tudjuk érni az őket leíró síkgráf duális gráfjában (3.14. ábra). A legkülső befoglaló terület tulajdonképp a poligonstruktúra kontúrja, vagyis legfelső szinten az egész kép, a többi szinten pedig egy teljes lyuk egy valódi poligonon belül.

3.4.2. Befoglalási fa

A valós és befoglaló területeket hierarchiába rendezhetjük, és fával ábrázolhatjuk. A fa gyökerében a kép kontúrját jelölő befoglaló terület lesz, a gyerekei azok a poligonok, amelyek a duális gráfban elérhetőek belőle. Így minden valós területet hozzárendeltük az őt közvetlenül tartalmazó befoglaló területhez, másrésztől az egész képet tartalmazó befoglaló területen kívül minden befoglaló terület egy valós



3.15. ábra. A valós és a befoglaló területek kontúros jelöléssel. A valós területek betűvel, a befoglalók számmal címkézve, illetve a struktúrájához tartozó befoglalási fa [11]

terület alá tartozik (3.15. ábra).

A befoglalási fában a gyökérben az egész területstruktúra kontúrja lesz. A páros szinteken befoglaló, a páratlanokon valós területek vannak. Egy befoglaló terület gyerekei azok a valós területek, amelyek belőle a duális gráf bejárásával elérhetők. A befoglalási fát nem egy teljesen külön adatszerkezetben tároljuk, hanem a duális gráfot egészítjük ki úgy, hogy az leírja a tartalmazásokat is. Ha egy poligonon belül van egy beágyazott poligonrendszer, akkor ennek kontúrja lesz a befoglaló területe. A befoglaló területek is megjelennek a duális gráfban, mégpedig úgy, hogy egy befoglaló területnek szomszédja lesz az összes olyan poligon, amely az adott poligonrendszer szélén van. A területekhez tároljuk azt is, hogy befoglaló vagy valós területet írunk-e le.

Ha valós területet írunk le, akkor tartalmazza a határoló irányított éleket negatív körüljárási irány szerint, és a beágyazott befoglaló területek listáját. Ha az adott terület befoglaló terület, akkor tartalmazza a határoló irányított éleket pozitív körüljárási irány szerint, és a beágyazott valós területek listáját. A síkgráfban a szélén lévő éleknek eddig csak az egyik oldalán voltak területek, mostantól a másik szomszédját is beállítjuk, mégpedig a befoglaló területre. Így a síkgráf és duális gráf élei meg fognak egyezni abban az értelemben, hogy a síkgráf éleinek a duális gráf egy másik szerepét adja meg, azaz, hogy az él által reprezentált polyline két területet választ el.

Mivel az élek megegyeznek a síkgráfban és duális grájában, ezért azokat nem is tároljuk kétszer. A duális gráfba azzal teszünk be egy élt, hogy a bal és jobboldalára eső területeket beállítjuk. Ez azért jó, mert könnyű a használat során egyikről másira áttérni, valamint egy vonallánc változás egyszerre jelenik meg mindenütt, így ilyen változtatással nem képzelhetők el topológiailag érvénytelen állapotok. Ezért mondjuk azt, hogy a duális gráf reprezentáció topológikus adatmodellt eredményez.

A síkgráf élei irányítatlanok, de egy élben a polyline mindkét irányítás szerint eltároljuk, hogy az építés során tudunk hivatkozni mindkét irányításra. A repre-

zentációban a pontokhoz tároljuk a pont koordinátáját, és a kimenő irányított éleket pozitív körüjárás szerint. Az élek irányítatlan éleket írnak le, de hozzárendelünk egy tetszőleges irányítást, ami szerint leírjuk a kezdő- és végpont azonosítóját, valamint a bal és jobb terület azonosítóját. Ha nincs jobb vagy bal terület, akkor a befoglaló terület azonosítóját használjuk. Ezenkívül tároljuk az élhez tartozó polylinet is.

Az eddig leírt adatszerkezet előnye, hogy támogat minden számunkra fontos műveletet, megőrzi a topológiai helyességet. A következőkben áttekintjük az előállítás módját.

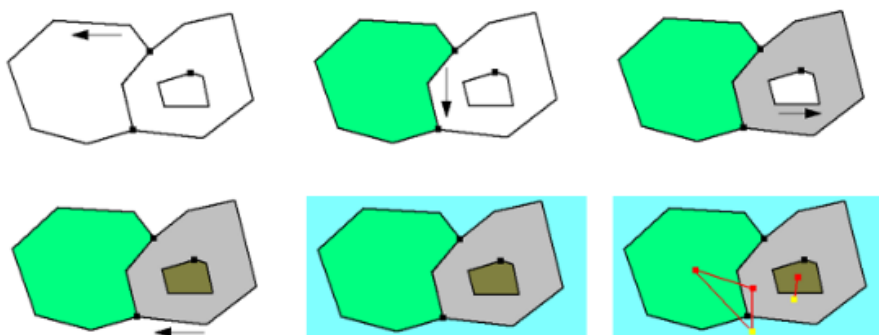
3.4.3. Síkgráf és duális gráf előállítása

A síkgráf előállításának a bemenete egy helyes polyline halmaz, amelyek a poligonok topológiáját meghatározzák.

- kigyűjtjük a vonalláncok végpontjait, ezekből létrehozuk a gráf pontjait és tároljuk, hogy a síkon melyik pontot jelölik.
- az összes vonalláncot egy éllel reprezentáljuk
- az él két végpontjához létrehozott pontok éllistába besúrjuk az élt, figyelve arra, hogy az élek sorrendjének rendezettsége megmaradjon

A síkgráf előállítása után a duális gráf előállítása következik. Ehhez először az élek feldolgozásával előállítjuk a területeket. Ehhez dolgozzuk fel az éleket valamilyen $e_1, e_2 \dots e_{|E|}$ sorrendben. Az i -edik lépésben a következőket tesszük (3.16. ábra):

1. nézzük meg, hogy az e_i mindkét oldalán van-e már terület
2. ha igen, akkor menjünk tovább
3. ha nem, akkor:
 - (a) hozzunk létre egy új T területet az e_i megfelelő oldalára
 - (b) járjuk körbe a területet úgy, hogy az e_i a körüjárás első éle olyan irányítás szerint, hogy T az e_i bal oldalára esik
 - (c) a pontokban mindig azon az élen menjünk tovább, ami a körüjárási irány szerint rendezett éllistában azután következik, amin bejöttünk
 - (d) a bejárt élek mindegyikének bal oldalára állítsuk be a T területet
 - (e) a T -hez tároljuk az így kapott körbejárás szerint azt a polylinet, ami határolja. Tulajdonképp ez lesz a poligon, amit a T jelöl



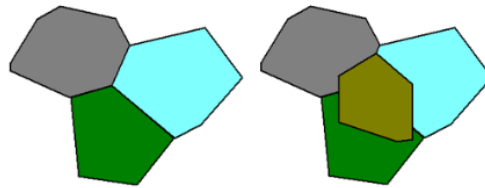
3.16. ábra. A kiinduló polyline halmaz; első lépés iránya; a síkgráf létrehozásának lépései; a végső síkgráf és a duális gráf területei; Duális gráf csúcsai és élei (sötétek a valós területeket jelölő csúcsok, világosak a befoglalók) [11]

A területek felépítésének algoritmusának eredményeképp nem csak a területek és az azok közti szomszédságok alakultak ki, hanem a poligonok is, amelyek a területet határolják. A körüjárás irányából kiderül, hogy valós, vagy befoglaló területről van szó. Ha ennek a poligonnak a csúcsai helyes körüjárás szerint tárolódnak a határoló poligonban, akkor a terület valós terület, különben határoló. Ezt a poligon előjeles területének kiszámításával tudjuk ellenőrizni. Ha az összes befoglaló területet megkaptuk, akkor nincs más dolgunk, mint a befoglalási fát megkonstruálni. Ezt a következőképp tehetjük:

1. rendezzük a befoglaló területeket területük szerint csökkenő sorrendbe
2. az első befoglaló terület, azaz a teljes kép, kivételével megkeressük a fában azt, hogy egy befoglaló terület melyik valós területbe tartozik, ezek listáját tegyük bele a duális gráfban a területet reprezentáló csúcsba
3. az aktuális befoglaló területből duál gráf bejárással elérhető területeket rendeljük hozzá a befoglaló területhez, ezek listáját tegyük bele a duális gráfban a befoglalási területet reprezentáló csúcsba

A rendezés miatt nem fordulhat elő olyan, hogy egy befoglaló terület tartalmaz egy megelőző befoglaló területet, így nem tudjuk elrontani az egymásba ágyazást. Ennek az adatszerkezetnek sok előnyös tulajdonsága van:

- Mivel semelyik élt nem tároljuk kétszer, ezért a topológiát nem tudjuk elrontani egy vonallánc egyszerű megváltoztatásával, ha átmetszünk vele egy másik vonalláncot. Ha azonban erre figyelünk, akkor ezen túl egy él változtatása mindkét poligonra vonatkozik, melyet elválaszt.



3.17. ábra. Példa a síkgráf megváltoztatására: a középső poligon beszúrása a gráfba új csúcsok és élek létrehozásával és törlésével jár, érdemesebb az elejétől felépíteni, mint az eredeti struktúrát változtatni [11]

- Az adatszerkezetben könnyű lekérdezni egy poligon szomszédait, hiszen ahhoz csak a duális gráfban kell a poligont reprezentáló terület szomszédait lekérdezni.
- Támogatja az akárhány szinten beágyazott területeket, és a tartalmazást is könnyű vizsgálni a kiegészítő befoglalási fa vizsgálatával.

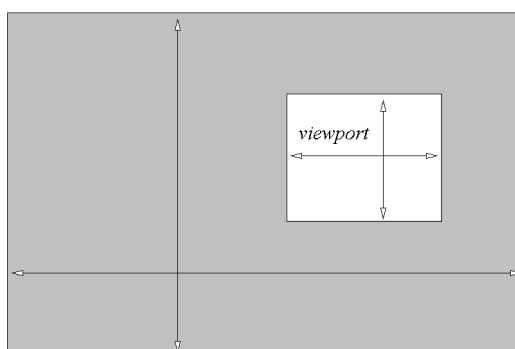
3.4.4. Síkgráf megváltoztatása

A felépített adatszerkezettel kapcsolatosan meg kell vizsgálni, hogy a változtatásokat mennyire nehéz elvégezni benne. Éleket törölni könnyű feladat, ilyenkor azt kell megtenni, hogy megnézzük, milyen területeket határolt el az él. Ha valós területeket, akkor azokat össze kell vonni egy közös területté. Ha valós és befoglaló területet, akkor a valós területet a befoglaló területhez csatoljuk abban az értelemben, hogy a szomszédai inntől azokon az éleken keresztül, amikkel eddig hozzá csatlakoztak, most a befoglaló területhez fognak. Miután ezt megtettük, még meg kell nézünk, hogy van-e olyan csúcs, amely csak két él találkozási pontjában van. Ha igen, akkor ezeket kitöröljük, és a két élt, ami a szomszédsági listájában van, összevonjuk.

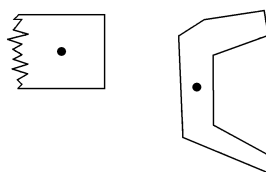
A síkgráfok hátránya, hogy a felépített struktúrába új csúcsok beszúrása esetén a duális gráfban többszörös élek lesznek, ezért a csúcsok beszúrását önmagában nem szabad megengedni, csak akkor, ha azt élek beszúrása is követi, és új területek jönnek létre. Még nagyobb körültekintést igényel új poligonok beszúrása területek közé a lokális topológia megváltoztatásával, mert ez szinte az egész gráf újraépítését követeli (3.17. ábra).

3.5. Térbeli indexelés

A digitális térképek – úgy a raszteres, mint a vektoros adatmodellt követők – általában nagy erőforrás igényűek. Megjelenítéskor megfeleltetést hozunk létre az adatbázis tartalma és a megjelenítő eszköz között.



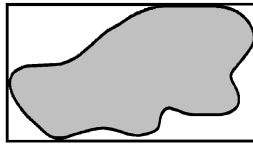
3.18. ábra. A viewport és az adatbázis térbeli kiterjedésének viszonya [8]



3.19. ábra. Konvex és konkáv poligonok centroidjai [8, 39]

A 3.18. ábrán a szürke terület az adatbázisunk teljes síkbeli kiterjedését mutatja, míg a fehér terület az u.n. viewport, amely a megjeleníteni kívánt területet jelzi. A megjelenítés legegyszerűbb módja lenne beolvasni az egész adatrendszer a memóriába, és onnan végezni a kirajzolást. Ez az eljárás több szempontból sem követendő. Egyrészt minek beolvasni olyan területek adatait, amik kívül esnek a viewporton, másrészt a memória mérete sem korlátlan, és célszerű csak a látni kívánt terület adatait betölteni. Komplikálja a helyzetet, hogy a háttértár (diszk), amiről a memóriába való betöltést végezzük, lényegesen lassúbb működésű, mint a memória, ezért célszerű az objektumokat lapokra (pages) osztva csoportosítani, és laponként beolvasni. Egy input/output művelet egy page beolvasását (írását) jelenti. Másfelől viszont a háttértárból való olvasások számát a viewport méretével összefüggően kell megállapítanunk. E két ellentétes szempontnak való megfelelés a térképi adatok indexelése révén lehetséges. Így le tudjuk csökkenteni az objektumok térbeli elhelyezkedését célzó keresések számát.

A viewportba eső objektumok gyors megjelenítéséhez tehát ki kell tudnunk zárni a keresésből a viewporton kívüli területeket. Sajnos a hagyományos adatbázis-kezelőkben megszokott indexelési technikák, mint pl. a B-tree a térbeli indexelésre nem alkalmasak Ennek a problémának a megoldására valók a térbeli indexelés algoritmusai. Még mielőtt rátérnénk néhány térbeli indexelési algoritmus ismertetésére, ismerkedjünk meg a centroid és a befoglaló négyszög fogalmával.



3.20. ábra. A befoglaló négyyszög [39]

A centroid egy olyan helyettesítő objektum, egy pont, amely a poligon belsőjében van, és alkalmas a poligon térbeli elhelyezkedésének reprezentálására. Konvex poligonok esetében a centroid kézenfekvő definíciója a poligon súlypontja, míg konkáv esetben egy a súlyvonalon lévő belső pont (3.19. ábra). A befoglaló négyyszög az a legkisebb területű négyyszög, amely teljes egészében tartalmazza a kérdéses poligont (3.20. ábra).

Mindkét fogalom a poligonok, polylineok (vonalláncok) helyettesítésére való. Azért használjuk őket, mert a térbeli indexeléskor ezen egyszerű objektumokkal reprezentáljuk az időnként rendkívüli összetettségű poligon és polyline objektumokat.

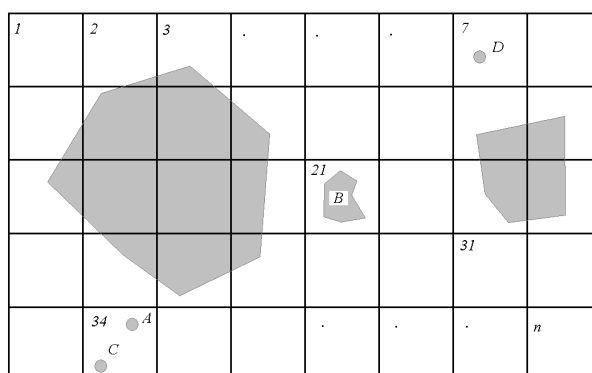
A térbeli adatok indexelési módszerei vagy a térbeli objektumokat, vagy pedig a vizsgált térrészt osztják fel partíciókra. A következőkben bemutatandó módszerek (Grid index, Quad-tree) a második kategóriába sorolhatók, tehát a tér valamely felosztásával próbálják gyorsítani az objektumok elérését, míg az utolsóként bemutatandó eljárás, az R-tree, egy adat vezérelt szerkezet.

3.5.1. Grid index

Osszuk fel a síkbeli objektumainkat tartalmazó síkot szabályos négyzethálóra, ahogy a 3.21. ábrán látható.

Az objektumok gyors megkeresésének első művelete az úgynevezett első szűrés, amikor – kihasználva, hogy a koordinátákra tett feltételekből gyorsan meghatározhatók a szóba jöhető gridek, valamint hogy a grid index a grid azonosítója szerint rendezett és gyorsan kereshető – jelölteket állítunk a lekérdezés feltételeit kielégítő objektumok halmazára. Az index objektumazonosítói segítségével beolvassuk ezt a jelölthalmazt, ami a teljes adatbázisnál várhatóan jóval kevesebb objektumot tartalmaz. A második szűrés dolgozza fel az így megtalált objektum részletes geometriáját (pl. poligonok esetén a vertex koordináták beolvasása és megjelenítése itt történik).

Amint a 3.21. ábrán megfigyelhető, nem mindig van összhangban a grid mérete az objektum méretével. Ha túl nagy a grid, akkor túl sok objektum esik egy adott cellába, ami szélsőséges esetben nem gyorsít a keresésen. Ha túl kicsi a grid, akkor a nagyobb objektumok túl sok gridet fednek le, ami szintén nem előnyös. Az eljárás valamelyest javítható több, eltérő rácsállandójú grid-szint bevezetésével.



3.21. ábra. Különböző méretű objektumok egy adott rácsállandójú négyzethálón [8, 17]

GridID	ObjectID	...	ObjectID	Attributumok
7	D	...	A	Attributum1
21	B	...	B	Attributum2
34	C	...	C	Attributum3
34	A	...	D	Attributum4

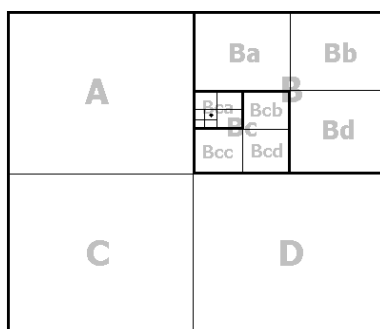
3.22. ábra. Az indexelt tábla és kapcsolata az eredeti táblával

A grid index egyszerű és hatékony módja a térbeli indexelésnek. Számos kereskedelmi szoftver alkalmazza is, annak ellenére, hogy hiányosságai nyilvánvalóak, különösen olyan esetekben, amikor az objektumok térbeli eloszlása nem egyenletes. Az volna ugyanis a kívánatos, hogy lehetőleg minden cellába azonos számú objektum essen, ami az állandó rácsméret miatt gyakorlatilag nem lehetséges. Inhomogén térbeli adateloszlás esetén a második szűrési fázis hatékonysága erősen lecsökkenhet, ezért jobb tulajdonságú indexelési algoritmusok szükségesek. Homogén térbeli adateloszlás esetén, mint amilyenek a raszteres adatszerkezetek, hatékony eszköz lehet a grid index is, bár a soron következő eljárás, a quadtree, sokkal hatékonyabb és elterjedtebb.

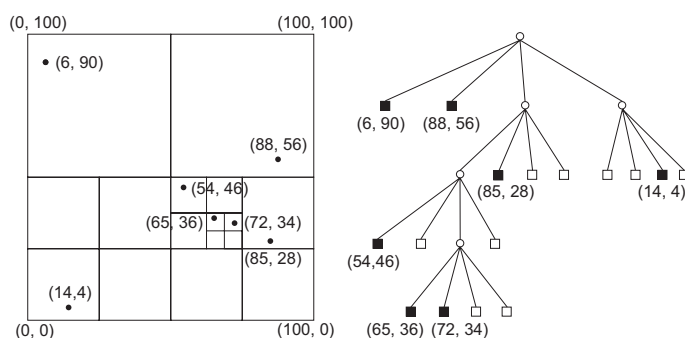
3.5.2. Négy-fa (Quadtree)

A négy-fa egy hierarchikus adatszerkezet, amely az oszd-meg-és-uralkodj elvhez hasonló rekurzív dekompozíció elvén alapszik. A sokféle négy-fa változathoz most az úgynevezett pont-tartomány négy-fát (point-region, vagy PR-quadtree) vázoljuk fel. Az indexelés alapja az adott síktartomány rekurzív felosztása négy síknyegyedre (3.23. ábra), amelyre az angol elnevezése is utal: space-driven structure.

Az indexeléshez használt struktúra egy olyan fa szerkezetként ábrázolható,



3.23. ábra. Egy pont és a quadtree térfelosztás [8, 31, 39]



3.24. ábra. A rekurzív térfelosztás és a felosztás szintjeit ábrázoló fa vonalak esetében [17, 31, 39]

melynek minden belső csúcsa egy-egy sík tartományt reprezentál, leveleiben pedig az objektumok helyezkednek el. A fa gyökere megfelel a kezdeti tartománynak. Minden belső csúcs tartalmazza saját felosztásának középpontját, valamint négy mutatót a felosztás után keletkező négy új tartománynak megfelelő csúcsokra (ezeket az égtájak angol rövidítéseinek megfelelően *NW*, *NE*, *SE* és *SW* jelöljük). Ha egy csúcs levél, akkor a csúcs egy ezt jelző mezőjét igazra állítjuk. A levelek vagy üresek – ezt egy külön mezőben feljegyezzük –, vagy pedig egy objektumot tartalmaznak, koordinátaival és egyéb attribútumaival. A levelek *NW*, *NE*, *SE* és *SW* mezői definiálatlanok. Megjegyzendő, hogy a belső csúcsok felosztást reprezentáló pontját csúcsonként tárolni nem feltétlenül szükséges. Mivel a felosztás mindig egyértelmű, ezért a fa műveleteinek megfelelő implementációjával a felosztásra vonatkozó információk tárolása a csomópontokban elkerülhető. A 3.24 ábra egy hét objektumot tartalmazó négy-fát szemléltet, feltüntetve az objektumok koordinátáit.

A fa felépítése az objektumok egyenkénti beszúrását jelenti. A fa gyökerétől indulva egészen addig haladunk lefelé, amíg levélbe nem érkezünk. A bejárás

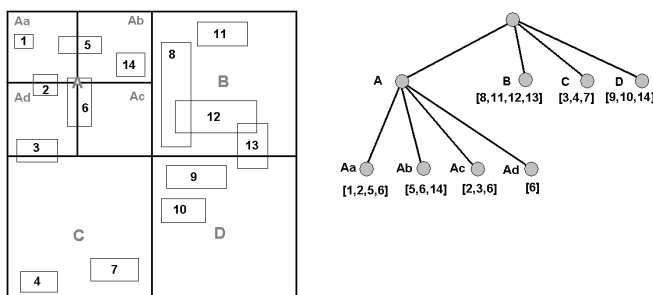
során a beszúrandó pont koordinátái alapján választjuk ki az adott csomópont megfelelő gyermekét. Ha levélbe érkeztünk és az üres, beszúrjuk az új pontot. Ha a levél már tartalmaz egy objektumot, akkor az aktuális tartományt újra és újra fel kell osztanunk, egészen addig, amíg a régi és az új csúcs már nem esik azonos tartományba. Ez a felosztás sok új altartomány létrehozását is jelentheti, ha a régi és az új pontok távolsága kicsi. A felosztásokat megszüntethetjük pontok törlések. Ha egy pont törlésével az adott partíció már csak egy pontot tartalmaz, akkor az alpartíciói összevonhatók.

A fa mérete, és alakja fontos a keresés, beszúrás és törlés műveletek hatékonyságának szempontjából. A pont-tartomány négy-fa szerkezete a fix térfelosztás miatt független a beszúrások sorrendjétől, alakja és mérete viszont függ a beszúrt objektumoktól. A fa műveleteinek futásideje alapvetően a fa mélységétől függ. A fa minimális mélysége M objektum esetén $\lceil \log_4(M - 1) \rceil$, ekkor minden objektum azonos szinten helyezkedik el, a fa kiegyensúlyozott. A fa azonban általában nem rendelkezik ezzel az előnyös tulajdonsággal, sőt felső korlátot sem tudunk adni a mélységére – az függ ugyanis az objektumok páronkénti távolságától. Ahhoz ugyanis, hogy a fába beszúrjunk két pontot, mindenképpen fel kell építenünk a fát legalább olyan mélységben, hogy a két pont közé essen egy tartomány-határ. Minél kisebb viszont a pontok távolsága, várhatóan ez annál később (annál mélyebb fát felépítve) következik be. Ha feltesszük, hogy az objektum-halmazunkban két pont távolsága legalább d valamint a tartományunk kiterjedése kezdetben r , akkor a fa mélységére adható egy $\lceil \log_2(\sqrt{2}(r/d)) \rceil$ korlát. Legrosszabb esetben ugyanis egy d hosszú szakasz valamely koordináta-tengelyre eső vetülete $d/\sqrt{2}$ hosszú, amiből maximum $r/(d/\sqrt{2})$ darab illeszthető egy r hosszú szakaszra.

Indexeléshez célszerű lehet egy olyan négy-fa változat használata, ahol a tartományok felosztásának feltételét a lap mérethez igazítjuk, azaz csak akkor osztjuk tovább az adott síkrészt, ha a síkrészbe eső pontok tárolásához szükséges lemezterület túlszordul a lap méretén. A fa ezen változatában a levelek nem egy-egy objektumot, hanem objektumok halmazát tartalmazzák, melyeket azonos lapon tárolunk.

Poligonok esetére a helyzet nem olyan egyszerű, mint pontokra. A poligon egy komplex objektum, amelyet számos pont alkot. A poligonok indexeléséhez felhasználhatjuk a centroid vagy a befoglaló négyszög fogalmát. Mindkét helyettesítő objektum alkalmas arra, hogy valamely térnegyedbe történő besoroláshoz megfelelően reprezentálja a poligont. A centroid egy és csak egy térnegyedbe való besorolást enged meg, míg a befoglaló négyszög alkalmazásával egy-egy objektum több térnegyedbe is eshet, amint az a 3.25. ábrán látható.

Az eddigi példákban vektoros modellt követő adatokra vizsgáltuk a négy-fa algoritmust. Raszteres adatmodellre is jól alkalmazható, mivel az adatok térbeli eloszlása teljes mértékben egyenletes, tekintettel a raszteres modell természetére. Raszteres adatmodell esetében célszerű lehet az úgynevezett tartomány négy-fák használata. Ezek felépítése hasonló, mint a pont-tartomány négy-fáké, csak a levelek az adott altartomány egészére vonatkozó információkat tárolnak. Ez a fa jól



3.25. ábra. A négy-fa és a poligonok befoglaló négyszögei [8, 31, 39]

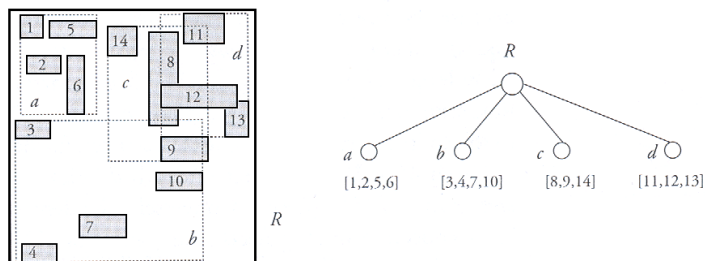
használható például képek tömörítésére: végezzük a felosztást addig, amíg az altartomány homogén területet nem reprezentál, ekkor a tartomány intenzitását tároljuk a levelekben.

3.5.3. R-fa (R-tree)

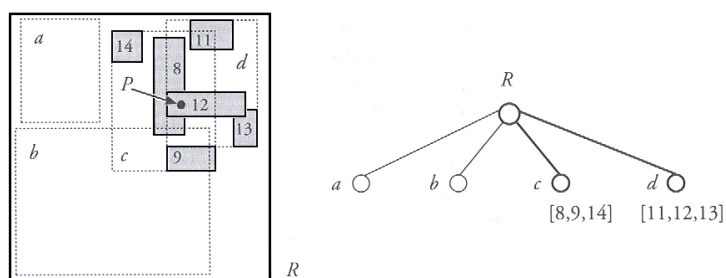
Az eddig bemutatott eljárások a teret osztották fel partíciókra, függetlenül az adatok térbeli eloszlásától. Az effajta felosztások hátrányai olyankor mutatkoznak meg, amikor az adatok térbeli eloszlása nagyon egyenetlen, mint például egyes vektoros adatrendszerek esetében. Az R-fa adatvezérelt fa szerkezet (data-driven structure). Többféle algoritmus is létezik, itt azonban csak az alapváltozatot ismertetjük. Az R-fa hierarchikusan egymásba ágyazott, esetleg egymást átfedő befoglaló négyszögek rendszere. A fa nodejai megfelelnek egy disk page-nek (a diskról egyszerre beolvasandó adathalmaz). A fa nodejaiban vagy levelek, vagy további nodeok helyezkednek el. Az olyan nodeokat, amelyek nem levelek, directory nodenak nevezzük (az elnevezés is utal arra tényre, hogy ezek a befoglaló négyszögek további nodeokat vagy leveleket tartalmaznak). Egy levél tartalmaz egy befoglaló négyszöget a koordinátaival (nemcsak 2D-s négyszögek lehetnek), és azonosítójával. A nem levél típusú nodeokban további nodeok vannak. A fa gyökere (root) R , kiinduláskor legalább két felosztást tartalmaz.

Egy egyszerű esetet mutat a 3.26. ábra. Az a, b, c, d nodeok által szimbolizált befoglaló négyszögek további nodeokat tartalmaznak, amelyek vagy további nodeokat vagy már leveleket tartalmaznak.

A legegyszerűbb keresés egy pont keresése (3.27. ábra), amely két lépésben valósul meg. Először a megkeressük a root összes olyan directory befoglaló négyszögét, amely tartalmazza a kérdéses pontot (ne felejtjük el, hogy a négyszögek átfedők lehetnek). Ezután sorra vesszük a megtalált directory négyszögeket, és megvizsgáljuk, hogy melyik levélbe (vagy további directory befoglaló négyszögbe) esik a pont. A pont keresésből felépíthetők a vonal és poligon keresések is.



3.26. ábra. Egy R-fa [31]. Az a, b, c, d befoglalók tartalmaznak számos további befoglaló négyszöget, amelyek vagy már levelek, vagy további nodeok, amelyek vagy további nodeokat vagy leveleket tartalmaznak



3.27. ábra. Pont keresés az R-fában [31]. Amint látható a P pont a c és d jelű directory befoglaló négyszögben is benne van, viszont a -ban és b -ben nincs, ezért csak c -ben és d -ben folytatjuk a keresést. Folytatva az eljárást azt kapjuk, a P pont a 8 és a 12 jelű levél által szimbolizált négyszögben van

4. fejezet

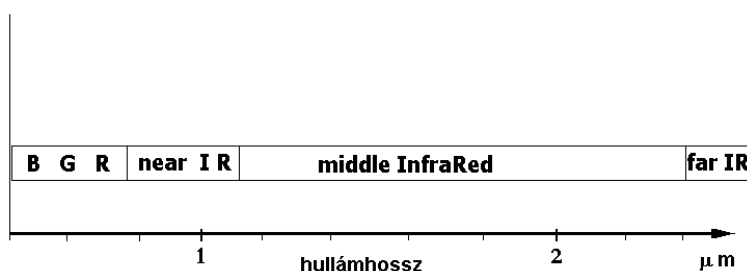
A raszteres adatmodell

A raszteres adatmodell a geoinformatika másik nagy adatmodellje. A vektoros és a raszteres modellek adatszerkezetében, a felszín leképezésének módjában, alapvető a különbség. A leképezett felszín részletgazdagsága is jelentősen eltér. Az adatfeldolgozás eszközei is teljesen különbözőek. Míg a vektoros adatmodellben a relációs adatbáziskezelők fogalomrendszerét használjuk, addig a raszteres adatmodellben a képfeldolgozás különböző módszereit használjuk. Ezek nagymértékben támaszkodnak olyan matematikai módszerekre, mint például a Fourier-transzformáció (lásd a Függelék), vagy a többváltozós matematikai statisztika olyan eszközei, mint a klaszter-analízis, főkomponens analízis.

4.1. Raszteres alapfogalmak

4.1.1. Az elektromágneses spektrum

Először is vessünk egy pillantást a fényre, mint elektromágneses sugárzásra (4.1. ábra). A raszteres GIS a fenti hullámhossz tartományokat használja. A Napból a Föld felszínére eső, és onnan visszaverődő fény révén a képalkotó



4.1. ábra. Az elektromágneses spektrum a számunkra érdekes hullámhossz tartományokban. RGB a látható tartományokat jelöli, míg near IR a közeli, middle Infrared a középső, és far IR a távoli infravörös tartományt jelöli

eszközök leképezik a felszínt, és digitális képet hoznak létre valamely adathordozón. Vizsgáljuk meg, hogy hogyan jellemezhetünk egy digitális képet.

Pixel, spektrális felbontás, hullámhossz

A digitális kép legelemibb objektuma a pixel, azaz a legkisebb képpont, amit a képalkotó eszköz még képes létrehozni. A képalkotó eszköz nemcsak műhold lehet, hanem akár egy egyszerű szkennert, vagy egy digitális fényképezőgépet. A pixel optikailag homogén, azaz színe, fényereje ugyanaz. A kép által átfogott föld terület alapján a pixelnek valós méret is megfelleltethető (ha tetszik méretarány).

A térbeli felbontás fogalma szorosán összefügg a pixel és a teljes kép méretével. Egy digitális kép felbontása annál nagyobb, minél több pixelből áll egy valós terület egység. Ha tehát a felszín egy kis területét egy $n \times m$ (n a sorok, m az oszlopok száma) pixelből álló digitális képpel képezzük le, akkor azt mondjuk, hogy nagy felbontású a kép, de ha ugyanez az $n \times m$ -es kép egy nagy felületet képez le, akkor a felbontás kicsi, noha a kép bytokban kifejezett értéke ugyanaz.

Fontos tudnunk, amikor egy adott digitális képet szemlélünk, hogy az eszköz, amely létrehozta (digitális kamera), az elektromágneses spektrum mely tartományára érzékeny. Mint tudjuk, egy egyszerű fényképezőgép a látható fény tartományában érzékeli a spektrumot, míg más eszközök az infravörös tartományban dolgoznak. A műholdak világában a spektrum különböző intervallumainak az érzékelésére számos eszközt fejlesztettek ki. Egyesek csak a látható fény frekvencia tartományában mérnek (pl. *SPOT*), míg mások, egészen széles frekvencia tartományban dolgoznak (pl. *LANDSAT TM*), amely a három látható tartományon (RGB) kívül még négy infravörös sávban is érzékel. Azt mondjuk, hogy egy eszköz spektrális felbontása annál jobb, minél több frekvencia sávban érzékel.

Vannak továbbá a rádió frekvenciás tartományban dolgozó eszközök is, amelyek *RADAR* elven működve képezik le a felszínt. Lényeges különbség a *RADAR* és az előbbi eszközök között, hogy a *RADAR* eszközök maguk „világítják meg” a leképezendő felszínt, míg az előbbieket a Napból érkező, visszavert fényt mérik.

4.1.2. Színmélység és átlátszóság

A színek digitális reprezentációja a számítástechnika fejlődésével párhuzamosan változott. Kezdetben 4, majd 8 biten ábrázolták a színeket. Ennek megfelelően a kapott színes kép tónusokban szegény, inkább csak színes ábrák megjelenítésére volt alkalmas. Mára már 24, sőt 36 bites színábrázolással dolgoznak a számítógépek, és ennek megfelelően a digitális képek is sok millió színárnyalatot tartalmaznak. Az 4.1. táblázatban áttekintjük a biteket és az ábrázolható színeket.

Az úrfotók is ezeket a színábrázolási módokat használják, sőt az adatfeldolgozás során gyakran átalakítják a képeket egyik modellből a másikba (pl. tematikus ábrázolások). Fontos hangsúlyozni, hogy az egy bites színmélységben ábrázolt kép nem azonos a köznyelvben fekete-fehérnek mondott képpel, ugyanis ott 256 (8 bit) szürkeségi fokozat ábrázolja a leképezett tárgy fényerősségi viszo-

Bit/pixel	Szín	Megjegyzés
1	2	Fekete-fehér
2	4	
4	16	VGA
8	256	
16	32 768	High color
24	16 777 216	True color
32	16 777 216	

4.1. táblázat. A színek pixelenkénti numerikus ábrázolása

nyait, míg az egybites modellben valóban csak két állapotú a kép minden egyes pixele, fekete vagy fehér. Speciális esetekben alkalmazzuk ezt a színmodellt, például éldetektálásakor, amikor a képen ott van nullától különböző érték, ahol egy él fut. Lényegében bármilyen logikai (boolean) érték tárolása is megvalósítható vele (pl. maszkok, célterületek, stb.)

Felmerülhet a kérdés, hogy minek 32 biten ábrázolni a képeket, amikor már 24 bit is több színt tartalmaz, mint amit a szemünk érzékelni képes. Nos, a negyedik bájt a kép átlátszóságát adja meg. Ez egy igen fontos paraméter. A raszteres képek hézag és átfedésmentesen fedik le a pixelek által meghatározott területet, vagyis nem látszik, ami alattuk van. A negyedik bájjal 256 átlátszósági érték ábrázolható, amivel meglehetősen finoman lehet szabályozni, hogy milyen mértékben látszódjék a kép alatti háttér. Ennek nemcsak raszteres képek esetén van jelentősége, hanem a vektoros térinformatika fejlett szoftverei is kiterjedten használják a poligonkitöltés átlátszóvá tételére.

4.1.3. Színmodellek

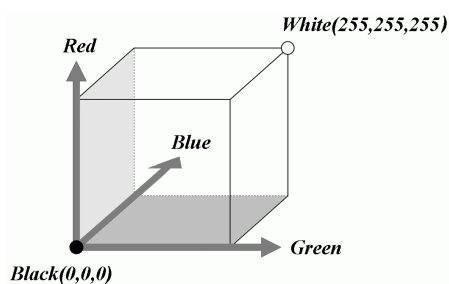
RGB színmodell

Képzeljünk el egy három dimenziós koordináta rendszert (4.2. ábra), ahol a három tengely a három színnek felel meg.

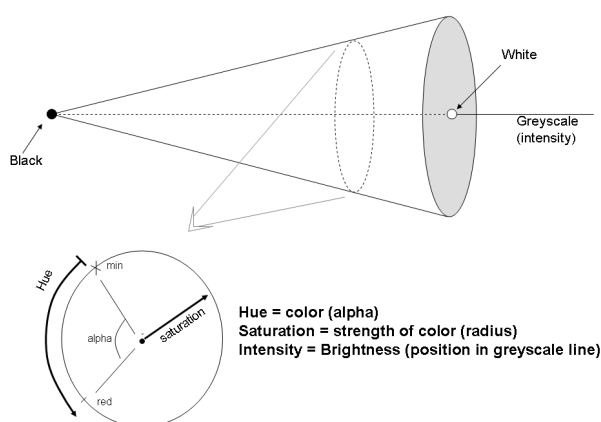
Az RGB színmodell a színeket a vörös (Red), a zöld (Green) és a kék (Blue) kompozíciójaként állítjuk elő a következő módon:

$$Color = a \cdot Red + b \cdot Green + c \cdot Blue \quad (4.1)$$

ahol a, b, c együtthatók az egyes alapszínek részaránya az adott színben. Alkalmazzuk erre az esetre a 24 bites színmodellt. 2^{24} féle színárnyalatot tudunk megjeleníteni, ami azt jelenti, hogy 2^8 vörös, 2^8 zöld és 2^8 kék árnyalat jeleníthető meg, vagyis alapszínenként 256 fényerősség érték adható meg.



4.2. ábra. Az RGB színekocka [8, 15]



4.3. ábra. A HSI színekúp [8, 15]

HSI színmodell

Az RGB színmodelltől alapvetően eltér a HSI modell. Míg az RGB modell derékszögű koordinátarendszerre épül, addig a HSI polárkoordinátákra, amint a 4.3. ábrán látható.

A HSI a hue (színárnyalat), saturation (szinkitöltés), intensity (fényerősség) szavak kezdőbetűiből ered. Az úrfotók feldolgozását végző szoftverek mindkét modellt használják. Vannak még további színmodellek is (pl. CMYK), amelyek főként a kiadványszerkesztés, nyomdatechnika területén érdekesek, az úrfotók elemzésében azonban nem lényegbe vágó a szerepük.

4.1.4. Raszteres adatszerkezetek

A raszteres adatszerkezetek sokkal egyszerűbbek, mint a vektorosok, mivel valamennyi közös tulajdonsága, hogy egy táblázat sorai és oszlopai tartalmazzák az egy egy pixel állapotát leíró adatokat (intenzitás értékek, magasság adatok, stb.). A 4.4. ábrán nézzünk meg egy egyszerűbb esetet, ahol például egy panchromatikus

(szürke árnyalatos, vagy gray scale) kép táblázatos megjelenését láthatjuk.

I_{11}	I_{12}	\cdots	I_{1n}
I_{21}	I_{22}	\cdots	I_{2n}
\vdots		\ddots	
I_{m1}	I_{m2}	\cdots	I_{mn}

4.4. ábra. I_{mn} az m -edik sor n -edik oszlopában lévő képpont intenzitás értékeit mutatja ($0 < I_{mn} < 255$)

Sokféle raszteres fájl formátum létezik, de valamennyi a fenti táblázatos logikára épül, éppen ezért csak vázlatosan foglalkozunk konkrét fájl formátumok bemutatásával. A 4.4. ábra táblázata csak egy sematikus bemutatása a digitális képek raszteres reprezentációjának, mivel valamennyi szoftvergyártó, aki úrfotók feldolgozásával foglalkozik, saját munkaformátumába konvertálva dolgozza fel a képeket. Ezeket meg kell majd ismerni, amint konkrét szoftverekkel kívánunk dolgozni. A 4.4. ábra egyetlen intenzitás sávot ábrázol, azonban ha van egy RGB képünk, akkor természetesen három ilyen táblázat (intenzitás mátrix) írja le a képet. Sőt, ha 32 bites a képünk, akkor egy negyedik mátrix tartalmazza az átlátszóság értékeit.

Multispektrális úrfotók esetében az RGB sávokon kívül még további, infravörös sávok is egy-egy intenzitás mátrixszal lesznek reprezentálva. Ha a ma egyre gyorsabban terjedő hiperspektrális távérzékelést is figyelembe vesszük, akkor akár többszáz intenzitás mátrix is kell egyetlen kép tárolásához. Ennek az adatmennyiségnek a kezelése már önmagában jelentős probléma. Ha figyelembe vesszük ezen többszáz sávok értelmezésének kérdéseit is, akkor belátható, hogy a hiperspektrális távérzékelés óriási kutatási terület.

Ismertebb raszteres adat formátumok

Számos raszteres adatformátum létezik, úgy a hétköznapi képfeldolgozásban mint a távérzékelésben. Egyesek operációs rendszerhez kötöttek (pl. bmp), mások ettől függetlenek. Egyesek tömörítettek, veszteséges (jpg, egyes tif formátumok) vagy veszteségmentes (tif,gif, png, jpg2000) algoritmusokkal készültek, mások nem tömörítettek. Csak néhány ismertebb formátumot ismertetünk, ezeket is csak vázlatosan, mivel számos szakkönyv foglalkozik velük.

Tiff. Általános célú formátum, a professzionális képfeldolgozás kedvelt képfarmátuma. Operációs rendszerektől, programoktól független. Az úrfotók feldolgozásában is ismert, és gyakran használt adatcsere formátum, amely a digitális kép pixelenkénti leírásán kívül számos képparamétert is tárol, szöveges és számszerű adatokat. Támogatja sekélyebb színmélységek kezelését éppúgy, mint

a professzionális 24 vagy 32 bites színmodelleket, egészen a nyomdai rendszerekben kedvelt CMYK színmodellig. Tömörített – veszteséges és veszteségmentes – valamint tömörítetlen változata egyaránt létezik. Geotiff nevű változata a georeferálás kontroll pontjait is tartalmazza, így a fájlt beolvassa valamely űrfotó elemző rendszerbe, georeferált, vagyis koordináta rendszerben elhelyezett adatállományt kapunk.

Jpg. Kiemelkedően hatékony tömörítő algoritmust használó formátum, amelyet részletgazdag fényképek tömörítésére fejlesztettek ki. Mára a digitális fénykép kezelés kedvelt formátuma lett. A veszteség mértéke a felhasználó által beállítható. Nagyobb mértékű tömörítéshez nagyobb képminőség romlás tartozik. Csak részletgazdag állományok, azaz fényképek tárolására alkalmas. Vonalas ábrákat tartalmazó raszteres állományok tömörítésére nem megfelelő, mert feltűnő lehet a veszteségből származó minőség romlás. Létezik a georeferálás kontroll pontjait is tartalmazó változata. Ilyenkor ezek az adatok külön fájlban kapnak helyet (jpw). A jpg2000 veszteségmentes formátum, de széleskörű elterjedése még nem történt meg.

Bil. Űrfelvételek tárolására kifejlesztett raszteres fájlformátum, amely alkalmas a kép frekvenciasávonkénti tárolására, a kalibrációs pontok, sarokpontok koordinátáinak, a vetületi rendszer paramétereinek tárolására. Igen elterjedt formátum, valamennyi űrfelvétel elemző és feldolgozó rendszer ismeri, sőt azok a vektoros rendszerek is felismerik, amelyek fel vannak készítve raszteres állományok fogadására.

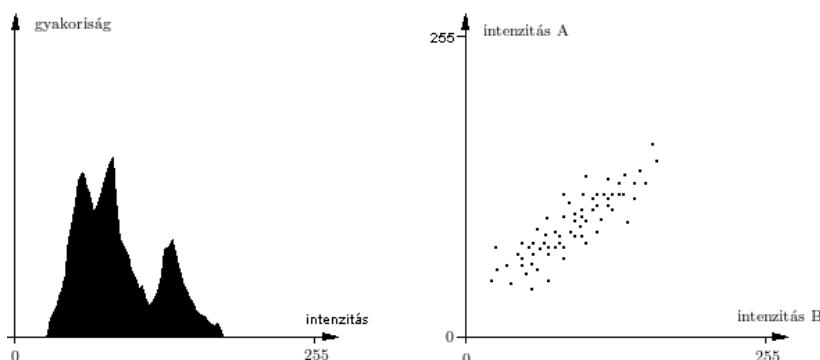
4.1.5. Intenzitás transzformációk

A hisztogram kiegyenlítés

Bármely digitális kép különböző fényerejű pixelekből áll. Elméletileg ezen képpontok fényereje a feketétől a fehérig terjed. Ez azonban a gyakorlatban soha nincs így. A digitális leképező módszerek, és a környezeti körülmények miatt a digitális képek intenzitásának dinamika tartománya kisebb, mint a lehetséges.

Természetesen minden képnek van egy legvilágosabb és egy legsötétebb pontja, de a legvilágosabb pont nem azonos az RGB színekockán látható elméleti fehér ponttal, illetve a legsötétebb képpont az elméleti fekete ponttal. Ha egy koordináta rendszerben ábrázoljuk a fényerősséget és ezek előfordulásának gyakoriságát, akkor kapunk egy hisztogramot, amely a digitális képfeldolgozás egyik alapvető ábrája (4.5. ábra).

A hisztogram az egyes pixelek intenzitás értékeinek előfordulási valószínűségét adja meg (P), hiszen azt számoljuk ki, hogy az összes ponthoz képest egy adott intenzitás érték (i) hányszor fordul elő:



4.5. ábra. Az ábra bal oldalán egy *A* nevű sávjának histogramja, vagyis a pixelek intenzitás értékeinek gyakorisága látható. Az ábra jobb oldalán az *A* és *B* nevű sávok intenzitásainak cross-plotja látható, amely a sávok közötti összefüggést mutatja grafikus formában [7]

$$P_i = \frac{\sum_{I(x,y)=i}(x,y)}{\sum(x,y)}$$

A hisztogram alapján meghatározható a kép minimális (*a*) és maximális (*b*) intenzitású pixeleinek értéke. A hisztogram transzformáció során valamely transzformációs függvény segítségével e két érték közé transzformáljuk a közbülső intenzitású pixeleket. Így voltaképpen széthúzzuk a kép hisztogramját (4.6. ábra), ezáltal megnöveljük az egyes pixelek intenzitás értékei közötti különbséget, vagyis a kontrasztot.

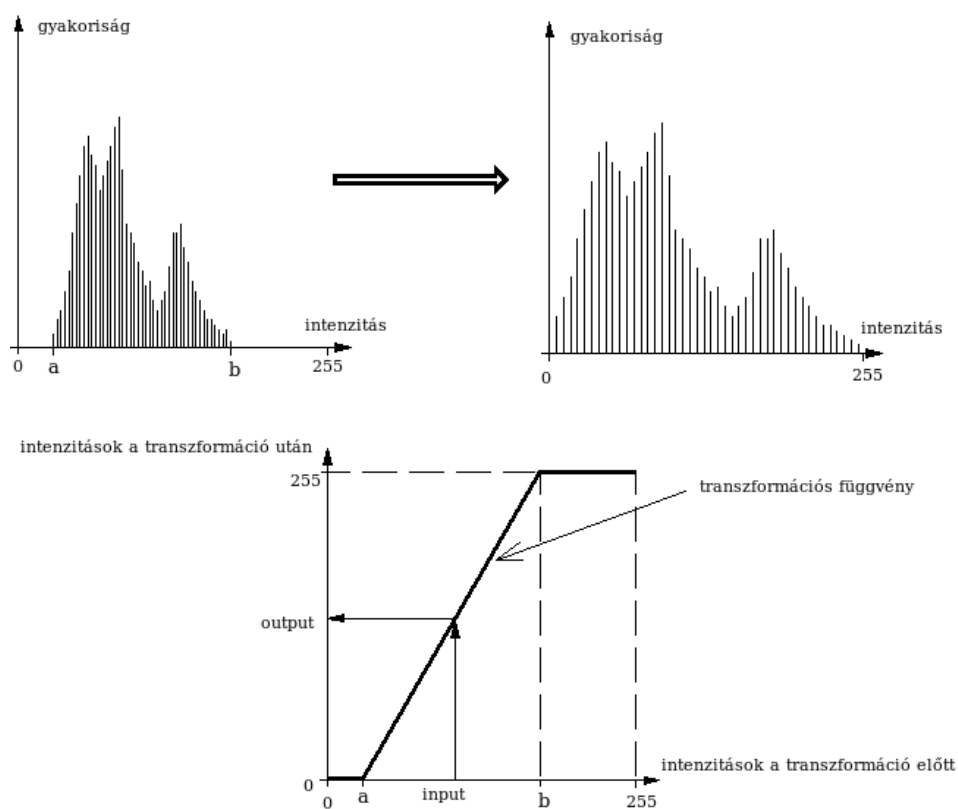
Nemcsak lineáris intenzitás transzformációk léteznek. Tetszőleges függvényt is alkalmazhatunk az intenzitások transzformálására, mint ahogy a 4.7. ábrán látható. Az *i*-edik képpont transzformált intenzitása megadható a következő formulával:

$$T(i) = \frac{I_{max} - I_{min}}{I_{maxBad} - I_{minBad}}(i - I_{minBad}) + I_{min}$$

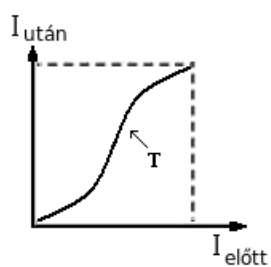
ami egy képmátrixra alkalmazva, 24 bites RGB színmodellt követve a következő:

$$g(x,y) = \frac{f(x,y) - \min}{\max - \min} 255,$$

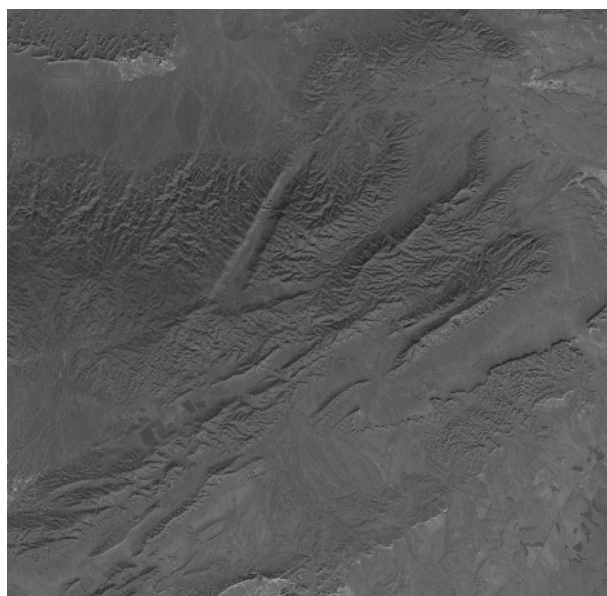
ahol $f(x,y)$ az eredeti képet jelöli az x,y helyen, \min, \max az eredeti kép minimális és maximális intenzitás értéke, $g(x,y)$ a transzformálás után kapott kép, amely a 0 – 255 közötti intenzitás tartományra terjed ki. Ezzel megnöveltük a pixelek közti intenzitás különbséget, ezáltal kontrasztosabbá tettük a képet. Ez az eljárás a hisztogram kiegyenlítés.



4.6. ábra. A hisztogram kiegyenlítés folyamata: az eredeti kép hisztogramja (bal felső ábra rész), a transzformáció utáni hisztogram (jobb felső ábra rész), valamint a transzformációs függvény (alsó középső ábra rész) [7]



4.7. ábra. Teszőleges T függvénnyel adhatjuk meg az intenzitás transzformációt

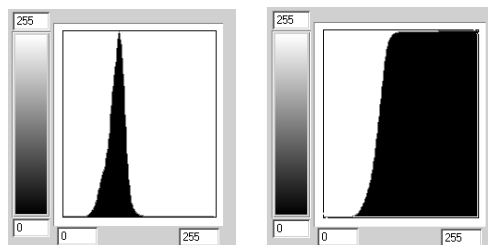


4.8. ábra. Az eredeti SPOT űrfotó [8]

Tekintsük meg az 4.8. ábrát, amely egy fekete fehér képet – egy űrfotót – mutat. Készítsük el a hisztogramot, amely a képen előforduló fényerősségek sűrűségfüggvényét mutatja egy 0-255-ig terjedő skálán. Amint látható, a kép legsötétebb és legvilágosabb pontja nem 0 és nem 255. Gyakran nem hisztogramot, hanem eloszlás függvényt ábrázolunk (4.9. ábra jobb oldali része).

Alakítsuk át az eredeti képet úgy, hogy a dinamika tartományt megnöveljük, vagyis a kép legsötétebb pontja legyen 0, és a legvilágosabb 255. Ennek eredményét mutatja a 4.10. ábra.

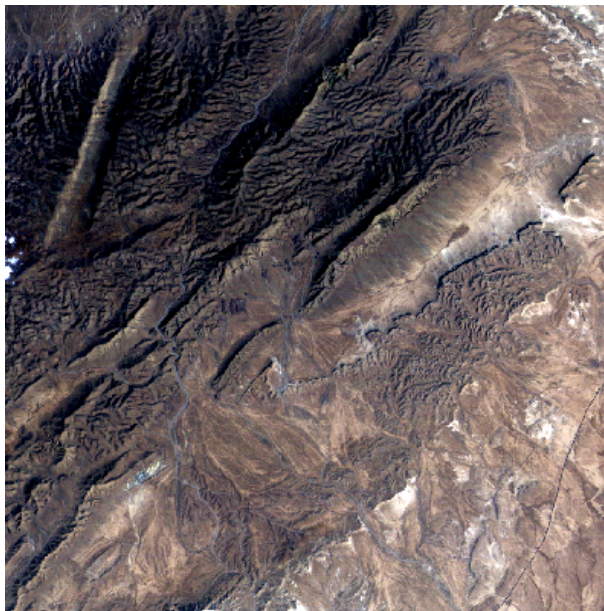
Képzeljük el, hogy mi történik egy színes képpel, ha ugyanezt az eljárást alapszínenként végezzük el, vagyis külön-külön az RGB frekvencia sávokra. A



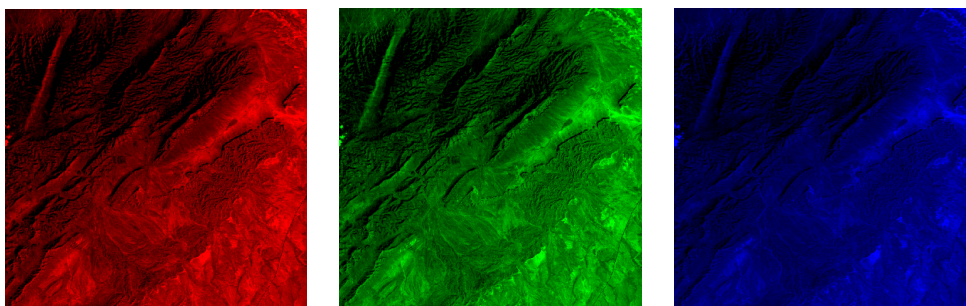
4.9. ábra. Az űrfelvétel hisztogramja és eloszlásfüggvénye. Az ábráról leolvasható, hogy a lehetséges dinamika tartomány lényegesen keskenyebb, mivel a legsötétebb pont valahol 60, míg a legvilágosabb 120 körül van [8]



4.10. ábra. Az úrfotó histogram szerint átalakított képe. A kontrasztnövekedés mértéke feltűnő, mivel a kép legvilágosabb része fehér lett, míg a legsötétebb fekete [8]



4.11. ábra. A nyers LANDSAT kép [8]



4.12. ábra. A nyers LANDSAT kép és az RGB alapszínek szerinti felbontása [8]

4.11. ábrán egy LANDSAT képet láthatunk. A 4.12. ábrán a 4.11. ábrán látható kép RGB összetevőit szemlélhetjük alapszíneként, amelyekből az ún. RGB kompozíció szerint áll elő a kép (lásd 4.1. formula).

Ha a hisztogram kiegyenlítést színenként alkalmazzuk előfordulhat, hogy a képünk jellege, színeinek karaktere, egymáshoz viszonyított aránya megváltozik, megjelenése el fog térni a természetestől. Ennek persze a jelentősége az űrfotók esetében nem túlzottan nagy, de normál fényképek esetén fontos lehet az eredeti kép színhűségének megőrzése. A 4.13 ábrán a hisztogram-kiegyenlített LANDSAT képet láthatjuk, míg a 4.14. ábrán a színcsatornánkénti kiegyenlített képeket.

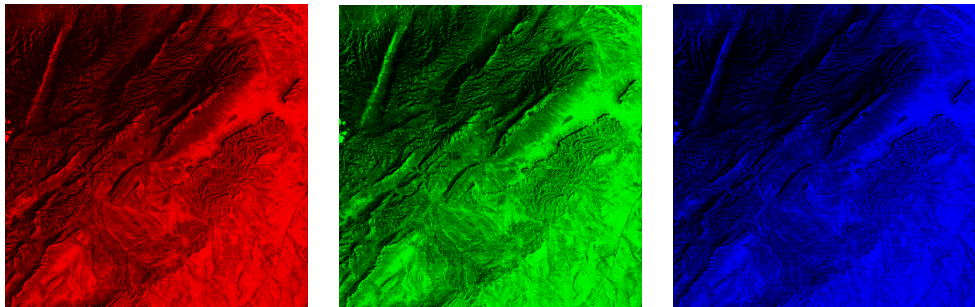
Az egyes alapszínek szerepét is felerősítjük azáltal, hogy intenzitásukat 0 és 255 közé transzformáljuk, így a nyers kép természetes színei mesterséges karaktert nyerhetnek a beavatkozásunknak köszönhetően, vagyis előfordulhat, hogy megbontjuk a kép színegyensúlyát. Ezzel csak arra szerettük volna felhívni a figyelmet, hogy a kép színeinek, színösszetevőinek megváltoztatásával óvatosan kell bánni. Mindíg tisztában kell lennünk egy-egy beavatkozás következményével, mert például az előbbi esetben kontrasztosabbá tettük a képet, de színegyensúlyát megbontottuk. Lehetséges úgy is a hisztogram kiegyenlítés, hogy a színegyensúly nem borul fel, vagyis a színek egymáshoz viszonyított arányát megőrizzük. Ebben az esetben viszont nem lesz kihasználva mindhárom színcsatornára a teljes rendelkezésre álló dinamika tartomány.

Binarizáció

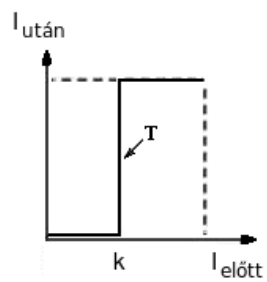
Az intenzitás transzformációk egy speciális esete a binarizáció, amely a képet úgy alakítja át, hogy egy k küszöbértéktől függően állapítja meg a transzformált értéket (4.15. ábra). Egy speciális binarizáló a szeletelő szűrő, ami a nullázást egy intenzitás sávban végzik. Vagy egy intenzitás tartományban engednek át, vagy azon belül nulláznak (4.16. ábra).

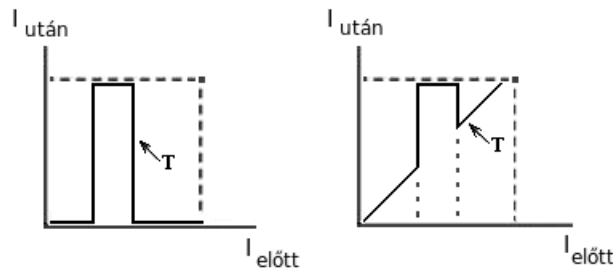


4.13. ábra. A hisztogram-kiegyenlített LANDSAT kép [8]



4.14. ábra. A hisztogram-kiegyenlített LANDSAT fotó RGB sávjai [8]

4.15. ábra. A binarizáció a k között alatti intenzitás értékeket nullává, az a feletti-eket eggyé alakítja



4.16. ábra. A szeletelő szűrők egy sávon belül vagy helybenhagyják a képet, vagy lenullázzák. Ezzel lényegében egy intenzitás sávot kivonnak a képből

4.2. Digitális szűrési eljárások

A digitális szűrők matematikai hátterének áttekintése a Függelékben található (169. oldal). A digitális szűrések általában azok a képfeldolgozó eljárások, amelyek vagy az időtartományban, vagy a frekvencia tartományban történő műveletekkel manipulálják a képet. Az időtartományt történeti okokból nevezzük annak, helyesebb volna inkább tértartománynak hívni. A frekvenciatartománybeli képet spektrumnak is nevezzük.

Jelöljük a képet $s(t)$ -vel, a kép spektrumát $S(f)$ -el és a Fourier-transzformációt \mathcal{F} -el. Az idő- és a frekvenciatartományt a Fourier-transzformáció köti össze:

$$S(f) = \mathcal{F}[s(t)] \quad (4.2)$$

illetve

$$s(t) = \mathcal{F}^{-1}[S(f)] \quad (4.3)$$

A 4.2 formula a direkt Fourier-transzformáció, míg a 4.3 az inverz Fourier-transzformáció.

Bizonyos szűrési eljárások hatását függvények formájában (átviteli függvény) adjuk meg a frekvencia tartományban (pl. konvolúciós szűrők), míg mások csak az időtartományban értelmezhetők (pl. medián szűrő). Ha például egy kép spektrumát megszorozzuk egy olyan négyzetfüggvénnyel, amely $[-f_f, f_f]$ intervallumon kívül nulla, akkor a spektrumból eltávolítjuk az f_f felső határfrekvenciánál nagyobb frekvenciájú összetevőket (vagyis simítjuk). A megváltozott spektrum inverz Fourier-transzformációjával megkapjuk a simított képet. A simítás mértéke természetesen függ a felső határfrekvenciától, mely minél alacsonyabb, annál erősebb a simítás.

Általánosságban tehát a digitális szűrők működése a következő:

1. legyen egy $s(t)$ függvényünk (ez jelenti a digitális képet)
2. Fourier-transzformáljuk, vagyis számítsuk ki a spektrumát: $S(f) = \mathcal{F} s(t)$

3. szorozzuk meg a spektrumot az átviteli függvénnyel: $S'(f) = S(f)A(f)$, ahol $A(f)$ az átviteli függvény
4. inverz Fourier-transzformáljuk a kapott spektrumot, és ezzel megkaptuk a szűrt képet: $s'(t) = \mathcal{F}^{-1}S'(f)$

A legtöbb „jól viselkedő” függvény Fourier-transzformálható. Mivel ismert a szűrési műveletünk átviteli függvénye (magunk adjuk meg, attól függően, hogy mit akarunk csinálni a képpel), akkor annak inverz Fourier-transzformálása előre kiszámítható. Az ismert konvolúciós azonosság szerint két függvény konvolúciója az időtartományban, megegyezik ezen függvények spektrumainak szorzatával:

$$s(t) * a(t) = S(f) \cdot A(f) \quad (4.4)$$

Ezt az összefüggést kihasználva, és az átviteli függvény inverz Fourier-transzformációjának ismeretében, az időtartományban is elvégezhetjük a szűrést, mégpedig úgy, hogy az eredeti képet (az időtartományban) konvolváljuk az átviteli függvény inverz Fourier-transzformáltjával:

$$s'(t) = s(t) * \mathcal{F}^{-1}[A(f)], \quad (4.5)$$

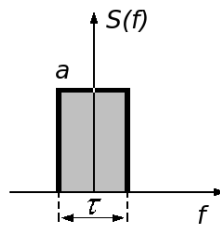
ahol $A(f)$ az átviteli függvény.

Az eddig okfejtések folytonos esetekre vonatkoztak. Diszkrét esetben a Fourier-transzformáció neve diszkrét Fourier-transzformáció (DFT), amelynek gyors és hatékony algoritmus a gyors Fourier-transzformáció (FFT). $s(t)$ a digitális kép, és az átviteli függvény ($A(f)$) is diszkrét, beleértve annak inverz Fourier-transzformáltját is, vagyis az $a(t)$ függvény mintavételezett értékeivel fog történni az időtartománybeli konvolúció (4.5. formula).

Az átviteli függvény inverz Fourier-transzformáltjának diszkrét változatára bevezették a kernel elnevezést, amely mára önálló fogalommá vált. Nemcsak olyan esetekben használják, amikor a művelet hatása megadható a spektrum valamely függvénnyel történő szorzataként, hanem olyan esetekben is, amikor az időtartománybeli művelet hatása nem adható meg a frekvencia tartományban (pl. rangszűrők).

A legtöbb digitális szűrő kernelt használ. Ezek a szűrők alapvetően úgy működnek, hogy egy kernelt, vagyis egy $[2k + 1 \times 2k + 1]$ méretű ablakot, futtatunk végig a szűrni kívánt kép minden pontján, úgy, hogy az aktuális képpont a kernel közepére esik, majd a kernel alá eső értékekből valamilyen eljárással kiszámolják az aktuális pixel szűrt értékét.

Általában ez a következő módon történik: legyen $g(x, y) = F\{f(x, y)\}$, ahol $g(x, y)$ a szűrt kép x, y koordinátájú pontját, f az eredeti képet, $F\{\}$ azt az operátort jelenti, amely az eredeti kép x, y koordinátájú pontjának szomszédjaiból kiszámolja a szűrt értéket. Azt a megfigyelést használjuk ki, hogy az egymáshoz közeli képpontok értékei jobban összefüggenek, mint az egymástól távoli pixelek. Ezeknek a szűrőknek egy másik fontos jellemzője, hogy nem rekurzívak. Ez azt jelenti,



4.17. ábra. A négyzögimpulzus a felülvágó szűrő átviteli függvénye a frekvencia tartományban (az egyszerűség kedvéért egy egydimenziós időfüggvényt látunk)

hogy az eljárás csak az eredeti intenzitásértékektől függ, azaz mindig az eredeti képből vesszük a képpont szomszédságát.

4.2.1. Konvolúciós szűrők

Jelölje $f_1(t)$ a konvolválandó függvényt, és $f_2(t)$ a kernelt. A konvolúciós, vagy lineáris szűrők úgy működnek, hogy a kernelben szereplő értékekkel konvolválják az időfüggvényt, és ez lesz a konvolúció értéke a t -edik helyen:

$$h(t) = \sum_{\tau=-\infty}^{\infty} f_1(\tau)f_2(t - \tau) \quad (4.6)$$

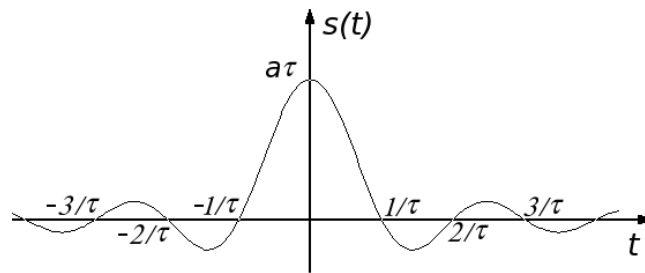
Kétdimenziós esetre, mint amilyen a digitális kép, a 4.6 formulával megadott konvolúciós a következőképpen alakul:

$$h(x, y) = \sum_{u=-\infty}^{\infty} \sum_{v=-\infty}^{\infty} f_1(u, v)f_2(x - u, y - v) \quad (4.7)$$

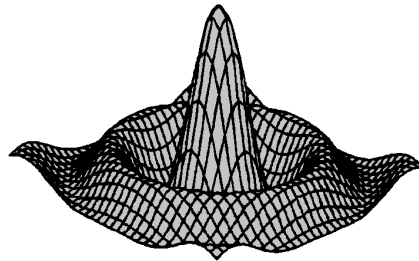
A szűrő átviteli karakterisztikáját a kernelben lévő értékek határozzák meg, (amelyek egyébként az átviteli függvény inverz Fourier-transzformáltjának diszkrét értékeiből állnak).

A szemléletesség kedvéért nézzünk meg egy f_f felső határfrekvenciájú felülvágó szűrőt. A szűrő átviteli függvényét mutatja a 4.17. ábra, illetve ennek inverz Fourier-transzformáltját a 4.18. ábra, amely megfelelően mintavételezve adja a kernelbe töltendő együtthatókat.

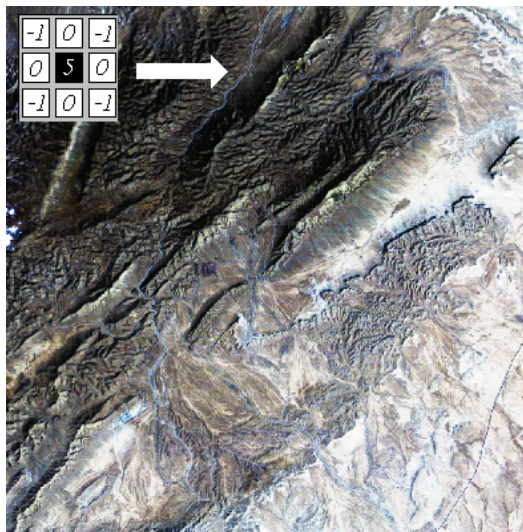
Ezzel a kernellel végrehajtva a konvolúciót kapjuk az ún. konvolúciós szűrőket. Átviteli karakterisztikájuk a kernel tartalmától (vagyis a frekvenciatartományban definiált átviteli függvényétől) függ. Kétdimenzióban, mint amilyen a digitális kép, a kernel a 2D-s *sinc* függvény, amely a 4.19. ábrán látható. A kernel képen történő mozgását mutatja szematikusan a 4.20. ábra.



4.18. ábra. A négyzög impulzus inverz Fourier-transzformáltja a *sinc* függvény az időtartományban



4.19. ábra. A kétdimenziós *sinc* függvény, amelynek mintavételezett értékeiből áll a simító szűrő kernelje



4.20. ábra. Az erősen felnagyított kernel mozgása a képen (LANDSAT képrészlet)

Szeperábilis szűrők

Ha a kernelmátrix speciális alakú, akkor a konvolúció végeredményét lényegesen gyorsabban számolhatjuk ki. Ha fennáll, hogy a w kernelmátrix felbontható egy oszlop- és egy sorvektor szorzatára, azaz $w(i, j) = u(i) * v(j)$, akkor megtehetjük azt, hogy először kiszámoljuk a kép u -val vett konvolúcióját, majd az eredmény v -vel vett konvolúcióját, azaz

$$f'(x, y) = \sum_{i=x-k}^{i=x+k} f(i, y) * u(i + k),$$

és ebből

$$g(x, y) = \sum_{j=y-k}^{j=y+k} f'(x, j) * v(j + k)$$

Box szűrő

A box szűrő olyan speciális kernelű szűrő, ahol a kernelmátrixban szereplő összes érték ugyanannyi, vagyis a kernel alá eső pixeleket átlagoljuk. A box szűrő hatása a kép simítása. Önmagában nem ad túl jó eredményt, de más szűrőkkel kombinálva (pl. élmegőrzők) hasznos eszköz lehet. Egyszerűségének köszönhetően igen népszerű szűrő, annak ellenére, hogy a spektrumra gyakorolt hatása meglehetősen kedvezőtlen. (Ha meggondoljuk, ilyenkor az időtartományban egy négyzetfüggvénnyel végezzük a konvolúciót, aminek az átviteli függvénye, a frekvencia tartományban egy *sinc* függvény, amiről minden elmondható, csak az nem, hogy értelmezhető lenne rá felső határfrekvencia).

Gauss szűrő

A Gauss szűrő kernelében az értékek a kétdimenziós Gauss eloszlás értékei szerepelnek. A Gauss eloszlás a következőképp számolható:

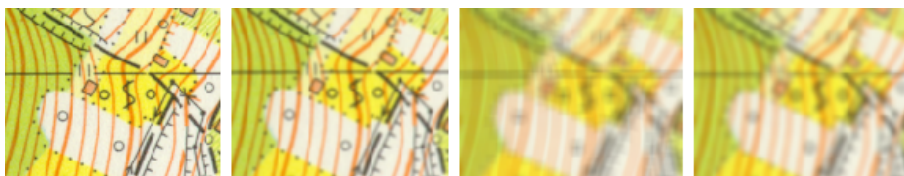
$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}}$$

Így a 0 várható értékű, σ szórású Gauss-görbét kapunk, amelyből a kernelt úgy számoljuk, hogy a rácspontokon mintavételezzük a G függvényt. Mivel

$$\frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}} = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}} * \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{y^2}{2\sigma^2}}$$

ezért a Gauss-szűrő is szeperábilis, az u és v vektor a következőképp számolható:

$$u(x) = v(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-(k+1))^2}{2\sigma^2}}$$



4.21. ábra. Balról jobbra: Eredeti kép. A kép 3x3-as, Gauss-szűrt változata, $\sigma=1.0$. A kép 7x7-es, box-szűrt változata. A kép 7x7-es Gauss szűrt változata, $\sigma=2.0$

A Gauss szűrőnek létezik egy speciális, még gyorsabb implementációja. Ha a haranggörbe értékeit 2 hatványokkal közelítjük, akkor nem kell szoroznunk, amikor a konvolúciót végezzük, hanem megfelelő számú bittel kell csak eltolni az értékeket. Ekkor az u és v vektorok

$$u(x) = v(x) = 2^{(k+1-|x-k|)}$$

alakúak lesznek.

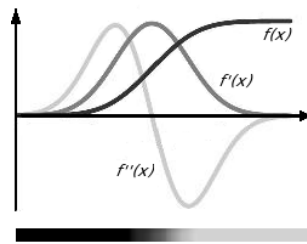
A Gauss szűrő, mint látható simító jellegű. A simítás mértéke a szórás nagyságától függ, természetesen nagy szórás esetén a kernel méretét is növelni kell (4.21. ábra). A Gauss szűrőnek szintén nem önmagában, hanem más algoritmusokban van szerepe, pl. a Canny-féle éldetektorban használjuk. Átviteli függvénye, ha nem is ideális, de lényegesen jobb a box szűrőnél. Gyakran használják „csonkító” függvényként, amivel például az ideális felülvágó kerneljének a hosszát csökkentik le, ezáltal javítva a futásidőt. (Ne feledjük, hogy ebben egy *sinc* függvény van mintavételezve.)

Hátránya, hogy a simítás miatt a képen található élek elmosódottá válnak a felső határfrekvencia függvényében. Ha meggondoljuk, nem meglepő ez a eredmény, hiszen a spektrumból eltávolítjuk a nagyfrekvenciás részeket, márpedig az éleken éppen a nagyfrekvenciás összetevők játszik a legfőbb szerepet.

A Gauss szűrő segítségével lehetséges a képek méretének egyszerű csökkentése (felezése). Az algoritmus úgy működik, hogy a kiinduló képre alkalmazzuk a Gauss szűrőt, majd elhagyjuk minden második sort és oszlopot. Az így létrejövő folyamatosan feleződő képeket Gauss-piramisnak is nevezik. A Gauss-piramisban az egymás feletti képeket egymásból kivonva a heterogén részek felerősödnek, ez a tulajdonság jól használható a textúrák detektálásánál.

Nemseparábilis szűrők

A nem separábilis szűrők azok, melyek kernelmátrixa nem írható fel két vektor szorzataként. Ilyenkor az eredeti konvolúciós képletben szereplő összegzést kell megvalósítani.



4.22. ábra. Az éleken az első deriválnak maximuma, a második deriválnak zérushelye van

Éldetektorok

Az éldetektorok olyan lokális szűrők, melyek a kép egy pontjában a szomszédos elemek segítségével leírt intenzitásfüggvény deriváltjával dolgoznak (4.22. ábra). Feladatuk, hogy az éleket kiemeljék, a hasonló pixelekből álló csoportokat pedig eltüntessék.

Gradiens szűrő

A gradiens szűrő használatával a kép, mint felület, pontjaiban vett deriváltak x és y irányú gradiensét közelítjük a differencia hányadossal.

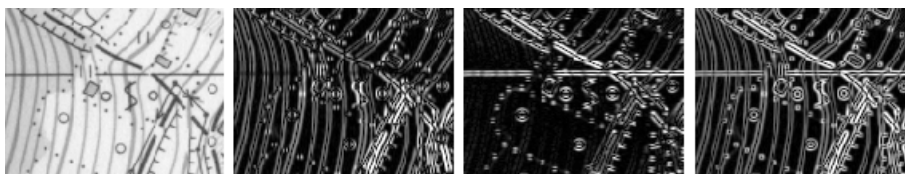
$$\nabla f(x, y) = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right)$$

A szűrő nagy intenzitásváltozásokra reagál, a homogén területekre 0-t ad eredményül. Kernelje a következő:

$$\mathbf{G}_x = \begin{pmatrix} -1 & 0 & 1 \\ 2-p & 0 & p-2 \\ -1 & 0 & 1 \end{pmatrix}$$

$$\mathbf{G}_y = \begin{pmatrix} -1 & 2-p & -1 \\ 0 & 0 & 0 \\ 1 & p-2 & 1 \end{pmatrix}$$

A p érték szabadon választható, gyakran használt értékek a $p = 2$, $p = 3$, $p = 2 + \sqrt{2}$. Az első esetben Prewitt, a másodikban Sobel, a harmadik pedig izotropikus operátorról beszélünk (4.23. ábra). Az eredményt p -vel normalizálni kell.



4.23. ábra. Balról jobbra: Eredeti kép. A kép x és y irányú gradienseinek közelítése izotropikus gradiens szűrővel. Az előző két képből számolt élkiterjedés

Laplace szűrő

A Laplace operátor definíciója:

$$\Delta f(x, y) = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

Ha a Laplace operátort diszkrétizáljuk, akkor a következő egyenletet kapjuk az x, y pontbeli második deriváltra:

$$f''(x, y) = f(x - 1, y) + f(x + 1, y) + f(x, y + 1) + f(x, y - 1) - 4 * f(x, y)$$

Így a Laplace szűrt értékt az (x, y) pontban a következő konvolúciós kernellel számolhatjuk:

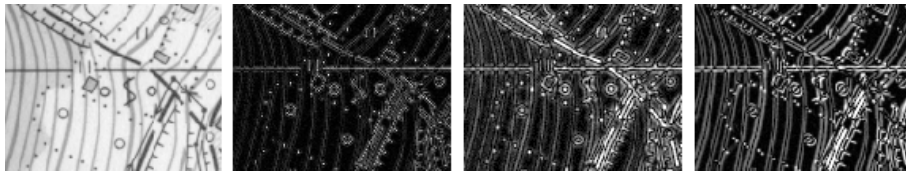
$$\begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

A Laplace szűrő a gyakorlatban a kép simított és eredeti változatának különbsége, ezért az intenzitásváltozásokra, így a hibákra is nagyon erősen reagál. Simítás nélkül, önmagában nem túl eredményes (4.24. ábra).

LoG szűrő

A Laplace szűrő előtt egy Gauss simítást alkalmazva jó éldetektort kaphatunk (4.24. ábra). Mivel a konvolúció asszociatív, ezért megtehetjük, hogy a Laplace és Gauss szűrő kerneljét konvolváljuk és az eredményként kapott mátrixot használjuk. Ezt a szűrőt szokták „Laplacian of Gaussian” (LoG) nevezni. A LoG-szűrő kevésbé érzékeny a zajra, jó éldetektor. A kernelmátrix a következőképp számolható:

$$LoG(x, y) = -\frac{1}{\pi\sigma^4} \left[1 - \frac{x^2 + y^2}{2\pi\sigma^2} \right] e^{-\frac{x^2 + y^2}{2\sigma^2}}$$



4.24. ábra. Balról jobbra: Az eredeti kép. A kép Laplace-szűrt változata. A kép LoG-szűrt változata. A kép emboss szűrt változata: ÉK-DNY irányú élek megtalálására beállítva

Emboss szűrő

Az emboss szűrők célja speciális irányú élek detektálása (4.24. ábra). Ehhez olyan kernelt alkalmazunk, amelynek két átellenes szélén +1 illetve -1 található. Attól függően, hogy milyen irányú átlóban vannak az értékek, az arra merőleges élekre reagál érzékenyen a szűrő. Példa egy lehetséges emboss szűrő kernelre:

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -1 \end{pmatrix}$$

Ez a kernel az ÉK-DNY irányú éleket fogja detektálni, míg az erre merőlegeseket észre sem fogja venni.

Canny-féle éldetektor

Az éldetektálás különösen fontos szerepet játszik az alakfelismerésben, a raszteres térképek vektorossá alakításában. Az élek a képnek azon helyei, ahol az intenzitás megváltozása a legnagyobb. Először is döntsük el, hogy mennyire kifinomult élek kimutatását szeretnénk. A legtöbbször érdemes előzetesen simító vagy medián szűrésnek alávetni a képet, hogy ne mutassunk ki minden apró, jelentéktelen élt. Ismert és egyszerű módja a simításnak a kép és egy Gauss-függvény konvolúciójának alkalmazása. Legyen h az f és g függvények konvolúciója. Kimutatható, hogy

$$h' = (f * g)' = f * g',$$

vagyis egy kép (jelöljük f -el) Gauss-függvénnyel (g) való konvolúciójának a deriváltja egyenlő a kép és a Gauss-függvény deriváltjának a konvolúciójával. Ezek alapján az éldetektálás a koncepciója következő:

1. Konvolváljuk f -t g' -vel.
2. Számítsuk ki h' abszolút értékét.
3. Definiáljuk éleknek mindazokat a helyeket, ahol a h gradiensének értéke meghalad egy előre meghatározott küszöb értéket.



4.25. ábra. Balról jobbra: 1. Az eredeti kép. 2.-3. A kép x és y irányú gradienseinek közelítése a gradiens szűrő segítségével. 4. Az előző két képből előállított élkiterjedés

A Canny-féle éldetektor nem érzékeny az élek állására, minden élt helyesen detektál, egy élt egy vonallal rajzol meg. Lássuk kissé részletesebben a működését:

$$\text{grad}f = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right) = (f_x, f_y)$$

ahol (f_x, f_y) a kép gradiense az (x, y) pontban,

$$M(x, y) = \sqrt{f_x^2 + f_y^2}$$

ahol $M(x, y)$ az él erőssége az (x, y) pontban, valamint

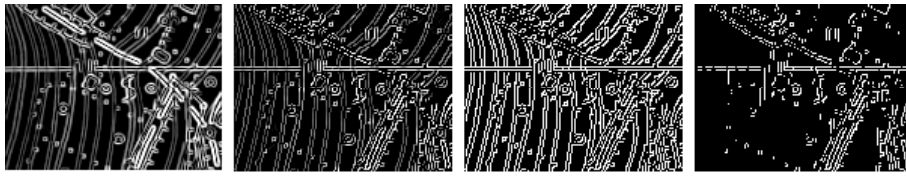
$$\Theta(x, y) = \arctan\left(\frac{f_x}{f_y}\right)$$

az (x, y) pontban vett derivált irányvektora, az él normálvektora.

Az f_x, f_y értékeket közelíthetjük úgy, hogy az x és y irányú izotropikus gradiens szűrővel vett konvolúcióját számoljuk a képnek az adott pontban, majd ebből kiszámoljuk minden pontban az $M(x, y)$ értékeket. Ezen a képen az élek látszanak, de minden él több pixel vastag, hiszen a képeken az élek általában nem tökéletesek, valamint ezzel a módszerrel még egy tökéletes él szomszédságában is 0-nál nagyobb értékeket kapunk.

Ezt a problémát oldja meg a nem-maximális élek kiküszöbölése. Ezt úgy tehetjük meg, hogy az $M(x, y)$ -t lemásoljuk egy kimeneti képbe, a $\Theta(x, y)$ -ban adott normálvektor által meghatározott szögben lépünk mindkét irányba az $M(x, y)$ képen, és ha bármelyik intenzitásérték nagyobb az aktuálisnál, akkor töröljük a pixelt a kimeneti képen. A nem-maximális élek kiküszöbölésének eredményeképp egy olyan képet kapunk, amelyen minden él rajta van, és mindegyik egyetlen vonalként jelenik meg.

Mivel így minden él, még a leggyengébbek is rajta lesznek a kimeneti képen, szükségünk lehet arra, hogy a gyengébb éleket eltüntessük és az erősebb, globális éleket tartsuk meg. Ha valamilyen küszöbölési eljárást használnánk, akkor az nem lenne tekintettel az élek folytonosságára, mi pedig nem szeretnénk, ha az élek megszakadnának. Erre a megoldást az élküszöbölés jelenti. Ennek alapötlete, hogy



4.26. ábra. Balról jobbra: 1: Az élkiterjedést tartalmazó kép. 2: A nem-maximális éleket kiküszöbölve kapott kép. 3-4: Az élek küszöbölése, 10, 40 illetve 40, 100 paraméterekkel. A második esetben csak a legerősebb élek maradtak meg

egy határ alatt minden pontot hagyjunk el, egy határ fölött mindent tartsunk meg, a köztes pixeleket pedig aszerint tartsunk meg, hogy eljutunk-e belőle biztosan jó pixelbe köztes pontokon. Ehhez a legjobb, ha mélységi bejárás egy változatát implementáljuk, kiegészítve azzal, hogy egy biztosan jó pontba érve az egész útvonalon megtartjuk a pixeleket (4.26. ábra).

A Canny szűrő ideális olyan esetekre, mikor additív, Gauss típusú zaj található a képen. Egy egyszerű közelítő módszer, hogy Gauss-szűrővel simítsuk el a zajokat a képen, és ezután alkalmazzuk a gradiens maszkját. Mivel a két szűrő lineáris, ezért a szűrés megvalósítható egyetlen lépésben, amint azt a szűrő koncepciót bemutató bekezdésben vázoltuk.

Kép élesítése

Az eddig tárgyalt szűrők segítségével lehetséges a képek élesítése, minőségének javítása. Működésének alapelve, hogy a kép eredeti és simított változatának különbségét hozzáadja az eredeti képhez:

$$g(x, y) = f(x, y) + [f(x, y) - f_s(x, y)]$$

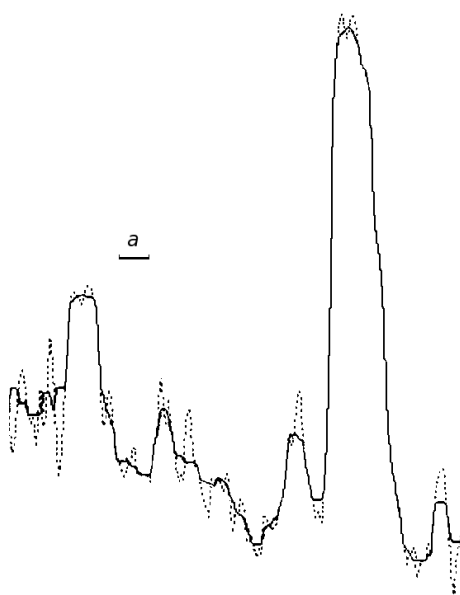
ahol f_s a simított képet jelenti.

4.2.2. Nemlineáris szűrők

A nemlineáris szűrők azok az eljárások, amelyek ugyancsak egy adott pixel szomszédos pixelei alapján számolják ki a szűrt értéket, de nem a szomszédos pixelek értékeinek valamilyen lineáris kombinációjaként, hanem más módon. Itt nem beszélünk kernelmátrixról, hanem csak kernelről, vagy kernelablakról.

Rang szűrők

A rang szűrők alapvető ötlete az, hogy a kernelablak alá eső pixelek intenzitásértékeit állítsuk nagyság szerint sorba, majd ez alapján a sorrend alapján válasszunk új intenzitásértéket a szűrendő pixelnek. A leggyakrabban használt



4.27. ábra. Egy zajos görbe (szaggatott vonal) és medián szűrt változata (folytonos görbe). A kernel hossza a

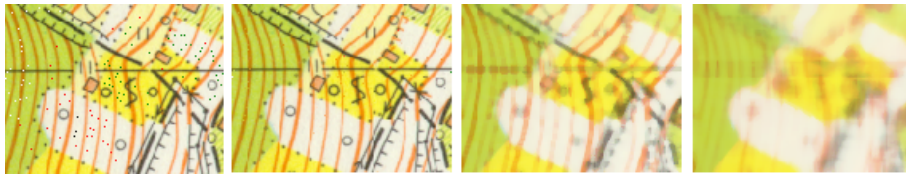
rang szűrő a medián szűrő, mely a nagyság szerinti középső értéket választja szűrt értékül (4.28. ábra).

A szűrés eredménye valamiféle simítás, amelyhez azonban átviteli függvény nem rendelhető. A szűrő a lokális zajokat hatékonyan eliminálja. A „salt and pepper” típusú hibákat eredményesen eltünteti, mert amikor egy ilyen pixelhez érünk, a környező pontok színétől kiugróan eltérő (sötét vagy világos) színű pontokat a rendezett kernel szélére sorolja. Az 4.27. ábra a medián szűrőnek egy zajos görbére gyakorolt hatását mutatja.

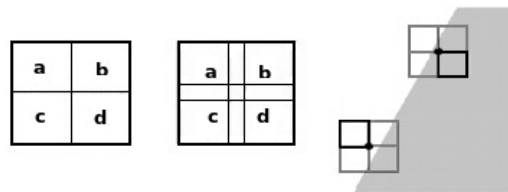
Digitális képekre a medián szűrő fontos tulajdonsága, hogy $2k+1$ méretű kernel esetén a k -nál vékonyabb vonalakat eltünteti a képről. Ez kívánatos eredmény, amikor a nagy területeket próbáljuk kiemelni. Sajnos az éleket eltolhatja és a sarkokat lekerekíti, de az algoritmus kiegészíthető úgy, hogy ez a hiba ne forduljon elő.

Olimpiai szűrő

Az olimpiai szűrő, a medián szűrőhöz hasonlóan, a kiugró intenzitás értékeket zajforrásból származónak tekinti. Egyes sportok olimpiai pontozási módszerét követi. Sorba rendezi a kernel alatti elemeket, majd a középsőtől legjobban elütőket eldobja. Paraméterként megadható, hogy a legnagyobb és legkisebb elemekből mennyit hagy figyelmen kívül.



4.28. ábra. Balról jobbra: Az eredeti kép, „salt and pepper” típusú hibákkal terhelve. Konzervatív simítással eltüntetve a hibák. 5x5 méretű mediánszűrővel tisztított kép. 11x11 méretű mediánszűrővel szűrt kép [11]



4.29. ábra. A Kuwahara-szűrő ablakbeosztása, és a kernel viselkedése egy élen [11]

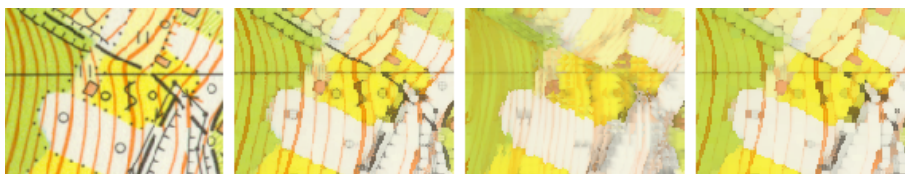
Konzervatív simítás

A konzervatív simítás zajszűrő eljárás, mely leginkább a „salt and pepper” típusú zajt képes eliminálni. Stratégiája, hogy a kernelablakba eső pixeleket nagyság szerint sorba rendezi az aktuális pixel kivételével. Így kapunk egy $[min..max]$ intervallumot, és megnézzük, hogy az aktuális pixel ebbe az intervallumba esik-e (4.28. ábra).

- ha a $[min..max]$ intervallumba esik, akkor nem változtatunk a pixel intenzitásán
- ha a maximumnál nagyobb, akkor az új érték a maximum lesz
- ha a minimum alá, akkor az új érték a minimum lesz.

Kuwahara szűrő

A Kuwahara-féle szűrési eljárás zajszűrő, simító hatású. Fontos tulajdonsága, hogy az éleket megtartja, nem tolja, vagy mossa el. Osszuk fel a kernel ablakot négy átfedő $(k + 1) \times (k + 1)$ méretű részre, melyek a négy sarokból indulnak (4.29. ábra). A szűrt pixel értéke azon négyzet átlagintenzitása legyen, ahol legkisebb a szórás.



4.30. ábra. Balról jobbra: Az eredeti kép. A kép 5x5 méretű Kuwahara szűrővel szűrt változata. A kép 11x11 méretű Kuwahara szűrővel szűrt változata. A szűrt képet újrászűrve 5x5 méretű Kuwahara szűrővel [11]

$$\begin{pmatrix} a & ab & b \\ ac & abcd & bd \\ c & cd & d \end{pmatrix}$$

A betűjelek azonosítják azt, hogy a pixelek melyik négyzetbe tartoznak. A középső pixel mindegyik négyzetben benne van. Miután a pixeleket besoroljuk a négy halmazba, kiszámoljuk mindegyik halmazban a pixelértékek átlagát és a empirikus szórást. Ezután annak a halmaznak az átlagát adjuk értékül a pixelnek, amelyikben a legkisebb a szórás (4.30. ábra).

4.3. Szegmentálás, küszöbölés

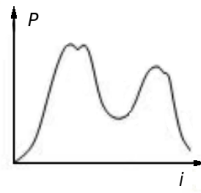
A küszöbölés a szürkeárnyalatos képek szegmentálásának egyik módja. Ilyenkor megadunk néhány küszöbértéket, amelyek intervallumhatárokat fognak jelölni. A küszöbölés speciális esete a kétszintes küszöbölés, a binarizáció, amikor egyetlen küszöbértéket adunk meg, így a pixeleket két osztályba soroljuk (4.15. ábra, 80. oldal).

A küszöbérték meghatározására több stratégia létezik, attól függően, hogy milyen célt tűztünk ki, mi a mértéke annak, hogy mennyire jó egy küszöbérték. Általában azt szeretnénk, ha a képen az objektumok jól eltérjenek a háttérüktől. Mivel két osztályba sorolhatunk be minden pixelt, ezért ez nem sikerülhet mindig tökéletesen, de a jó küszöbölés ezt a lehető legjobban közelíti.

Otsu-féle küszöbölés

Tekintsük meg az 4.31. ábrát, amely egy kép hisztogramját mutatja. Látható, hogy az eloszlás két intenzitás érték körül csoportosul, vagyis a hisztogram két csúcús. A cél a kép szegmentálása, mégpedig úgy, hogy az intenzitás eloszlás csúcsainak megfelelő osztályok jöjjenek létre.

Otsu szerint az a jó osztályozási eredmény, ha a két osztály közötti szórás a lehető legnagyobb. Ehhez kiszámolja a kép pixeljeinek empirikus várható értékét és szórásnégyzetét.



4.31. ábra. A kép hisztogramja két csúcsú. A jó szegmentálás a csúcsoknak megfelelő osztályokat állítja elő

$$\mu = \sum_{i=0}^{255} iP(i)$$

$$\sigma^2 = \sum_{i=0}^{255} (i - \mu)^2 P(i)$$

Egy t küszöbérték mellett az egyes osztályokon belüli szórás és várható érték:

$$\mu_1(t) = \frac{1}{q_1(t)} \sum_{i=0}^t iP(i)$$

$$\sigma_1^2(t) = \sum_{i=0}^t (i - \mu_1)^2 P(i)$$

valamint

$$\mu_2(t) = \frac{1}{q_2(t)} \sum_{i=t+1}^{255} iP(i)$$

$$\sigma_2^2(t) = \sum_{i=t+1}^{255} (i - \mu_2)^2 P(i)$$

ahol

$$q_1(t) = \sum_{i=0}^t P(i)$$

és

$$q_2(t) = \sum_{i=t+1}^{255} P(i).$$

Az osztályokon belüli szórás a két osztály szórásának súlyozott összege, azaz

$$\sigma_w^2(t) = q_1(t)\sigma_1^2(t) + q_2(t)\sigma_2^2(t)$$

Az osztályok közti szórást a következőképp definiáljuk:

$$\sigma^2 = \sigma_w^2(t) + \sigma_b^2(t)$$

vagyis

$$\sigma_b^2(t) = \sigma^2 - \sigma_w^2(t).$$

Fejessük ki $\sigma_b^2(t)$ -t:

$$\sigma_b^2(t) = q_1(t)q_2(t)(\mu_1(t) - \mu_2(t))^2 = q_1(t)(1 - q_1(t))(\mu_1(t) - \mu_2(t))^2$$

Az optimális szórás számunkra az, ami a lehető legjobban elkülöníti az osztályokat. Mivel a kétféle szórás összege egyenlő a teljes szórással, ezért két, egymással ekvivalens célunk lehet az optimum megtalálásához: minimalizáljuk az osztályokon belüli szórást, vagy maximalizáljuk az osztályok közöttit. Ha az utóbbit választjuk, akkor az egyes t értékekre a $q_1(t+1), \mu_1(t+1), \mu_2(t+1)$ értékeket számolhatjuk a $q_1(t), \mu_1(t), \mu_2(t)$ értékek felhasználásával:

$$\begin{aligned} q_1(0) &= 0 \\ \mu_1(t+1) &= \frac{q_1(t)\mu_1(t) + (t+1)P(t+1)}{q_1(t+1)} \\ \mu_1(0) &= 0 \\ \mu_2(t+1) &= \frac{\mu - q_1(t+1)(\mu_1 + 1)}{1 - q_1(t+1)} \end{aligned}$$

Íly módon a következő algoritmust kapjuk:

1. számoljuk ki P -t, μ -t és σ -t
2. $t = 0$ -tól 255-ig számoljuk ki minden értékre $q_1(t), \mu_1(t), \mu_2(t)$ értékeket, majd ebből a σ_b^2 értéket
3. válasszuk $t_{optimal}$ -nak az $argmax(\sigma_b^2)$ -t

Az Otsu-féle küszöbölés eredményeit mutatja a 4.32. ábra. Az Otsu-féle küszöbölés általánosítható, azaz több szintű küszöbölést is lehet ezzel a stratégiával előállítani.

Itt több küszöbölési problémát nem tárgyalunk, mert külön fejezetben foglalkozunk adatbányászati módszerekkel, osztályozási eljárásokkal, ahol még foglalkozunk különböző szegmentáló, osztályozó algoritmusokkal.



4.32. ábra. Balról jobbra: Eredeti szürkeárnyalatos kép. A kép Otsu eljárásával küszöbölt változata. A kép halványabb változatát az eljárás ugyanolyan jól küszöböli. Egy előre beállított küszöbérték az első képet jól, a másodikat teljesen rosszul küszöbölte volna

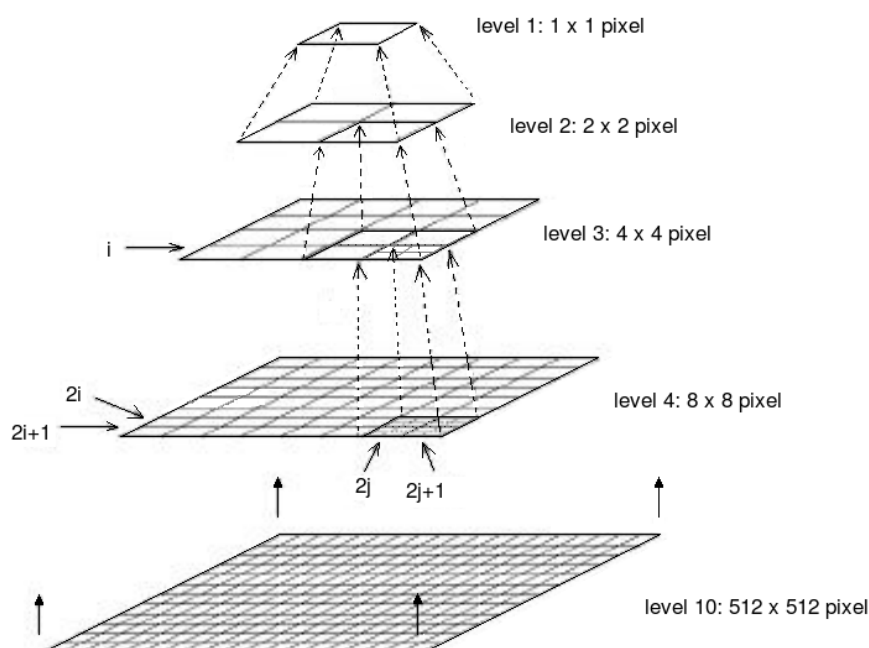
4.4. LOD algoritmusok

Úrfelvételek, légifotók vagy digitális magassági modellek kezelése, megjelenítése jelentős nehézséget jelenthet a térinformatikai szoftverek fejlesztőinek. Gondoljunk csak el, hogy amikor nagy felületet kívánunk megjeleníteni, amit sok nagyfelbontású kép fed le, elképesztő adatmennyiséget kell kezelnünk, kirajzolnunk, miközben a monitorok felbontóképessége (pl. 1024×768 , vagy 1400×1050) lényegesen kisebb adatmennyiséget képes megmutatni. Ekkora területre értelmetlen eredeti felbontásban rajzolni a pixelek értékeit (még TIN modellt használva is), vagyis a kirajzolásra szánt területre vonatkozó adatokat ajánlatos ritkítani. (A ritkítás elméleti háttéréről szól a függelék mintavétellel foglalkozó fejezete.)

A képek, domborzat modellek leegyszerűsítését végző algoritmusokat gyűjtőnéven LOD (level-of-detail) algoritmusoknak hívják. Alapvetően két fajtája létezik, a statikus, és a folytonos. A statikus változat lényege, hogy előre elkészítjük egy modell különböző részletességű változatait, és ezeket cserélgetjük attól függően, hogy milyen részletességre van szükségünk az adott felhasználás szempontjából (pl. változó nagyítás miatt). Úrfotókra vagy magassági színezett raszteres állományok megjelenítésére ez a módszer megfelelő.

A terepmodellek esetén nem túlzottan jók a statikus algoritmusok. A terepmodellek ugyanis általában nagyok, és ezért könnyen előfordulhat, hogy egy részét jelentősen közelebről szemléljük. Így aztán nem lehet egyszerre megváltoztatni a teljes modell részletezettségét. A folytonos LOD (CLOD) algoritmusok pont erre a problémára nyújtanak megoldást. Lényegük, hogy darabokra bontják a kérdéses modellt, és a nézőponttól (vagy egyéb más, előre meghatározott mértéktől) függően minden alkalommal elemzik, hogy mit lehet egyszerűsíteni.

A LOD algoritmusok folyamatosan változtatják az egyes részek felbontását. A kérdés az, hogy mi alapján lehet megmondani, hogy egy adott részt milyen minőségben kell megjeleníteni, milyen részletezettségi szintre van szükség. A hiba fajtájára és mértékére sok megoldást lehet találni, csak a felhasználási módtól függ, hogy mikor melyik lesz a célravezető.



4.33. ábra. A Gauss-piramis különböző felbontású szintjei. A képméreték csökkenése kettő hatványai szerint történik, vagyis például az eredeti kép 512×512 méretű, amelyből az első ritkítás után 256×256 méretű kép lesz. Ebből 128×128 -as lesz a, majd 64×64 -es kép lesz, és így tovább

4.4.1. A Gauss-piramis

Az egyik legismertebb, és talán leggyakrabban alkalmazott statikus LOD algoritmus a Gauss-piramis. Segítségével igen nagyméretű képek is gyorsan jeleníthetők meg. Ezt a sebesség növekedést azzal éri el, hogy előzetesen több, különböző felbontású képet állít elő, és mindig azt jeleníti meg, amire egy adott nagyításban szükség van. Alapelve eléggé egyszerű.

Elméletileg a következőképpen működik az algoritmus: vegyünk egy nagy felbontású képet. Hajtsunk végre rajta simítószűrést az eredeti kép felső határfrekvenciájának felére, majd ritkítsuk meg ennek megfelelően a képet, vagyis újramintavételezzük az eredeti mintavételi távolság kétszeresével. Az eredmény képre folytassuk az eljárást, és ismét hajtsunk végre simító szűrést a már egyszer ritkított kép felső határfrekvenciájának felével, majd ennek megfelelően ritkítsuk meg a képet, és így tovább (4.33. ábra).

A gyakorlatban a Gauss-piramis kiszámítása a következő módon történik. Jelölje az $l - 1$ -edik szintnek megfelelő képet g_{l-1} . Az l -edik szint pixel értékeit a



4.34. ábra. A Gauss-piramis kiszámítása (egy dimenziós illusztráció)

$$g_l(i, j) = \sum_{m=-2}^2 \sum_{n=-2}^2 w(m, n) g_{l-1}(2i + m, 2j + n)$$

formulával számoljuk ki, amely mint látható, egy konvolúció. A 4.34. ábrán láthatjuk, hogy (egy dimenzióra egyszerűsítve az esetet) hogyan számoljuk ki a ritkított kép pixel értékeit a sorban alatta lévő, nagyobb felbontáshoz tartozó értékekből.

5. fejezet

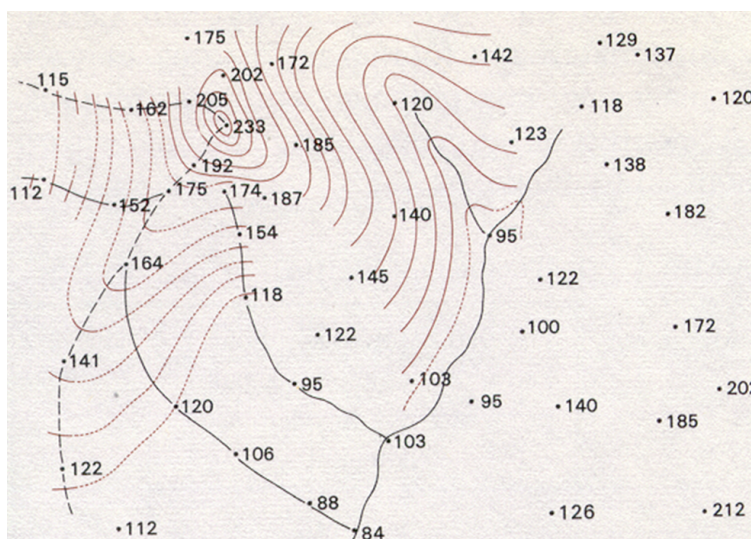
3D grafika, felület modellezés

A geoinformatikában használatos 3D-s felület modellezési algoritmusok, valamint a 3D-s megjelenítés módszerei ugyanazok, mint amelyek a legtöbb 3D-s szoftverben megtalálhatók. Ezek az eljárások működnek valamennyi 3D-s megjelenítő, mérnöki, építészeti tervező rendszerben, sőt a játékprogramok valóság-hű megjelenítő moduljaiban is. Hangsúlyozni kell azonban, hogy nemcsak megjelenítésről van szó, hanem egy általános objektum leírási módról, amely a pontokhoz három egyenrangú koordinátát rendel (x,y,z) , szemben a kétdimenziós térinformatikával, ahol az objektumok leírása alapvetően kétdimenziós (x,y) , még akkor is, ha a domborzati viszonyok leírása a cél (pl. szintvonalak).

5.1. A 3D-s domborzat modellezés áttekintése

A Föld felszíne, mint tudjuk nem sík, ezért teljesértékű leírása csak három dimenzióban lehetséges. Mivel e felszín nem adható meg analitikus alakban, ezért pontos leírása minnél több pontjának x, y, z koordinátákkal történő jellemzésével lehetséges. Alapvetően kétféle leírási mód létezik. Az egyikben a terep jellegzetes pontjainak megadásával, illetve mérési adatok alapján írjuk le a felszínt, míg a másikban a domborzati jellemzőkre való tekintet nélkül egyenletes távolságra lévő rácspontokban adjuk meg a terepfelszín pontjait. A hagyományos térinformatikai adatmodellek szóhasználatával élve a felszín leírható vektoros és raszteres logikájú adatszerkezetekkel egyaránt. Mindkét megközelítési módnak van létjogosultsága. Mindkét módszer fel tud mutatni előnyöket a másikkal szemben, éppen ezért nincs értelme kitéüntetni egyik leírási módszert a másikkal szemben.

Tekintsük meg a 5.1. ábrán látható klasszikus szintvonalas térképet. A tradicionális térképész a magassági pontokból kiindulva a saját tudása, tapasztalata alapján szintvonalakat rajzol. A munkafolyamat sok szubjektív elemet tartalmazott, amelynek számos előnye is volt. Kevés adatpont alapján is elkészült a szintvonalas térkép, és egy jó térképész sosem rajzolt lehetetlen vagy valószínűtlen terepi képződményeket. A terepi bejárás is segíthetett a szintvonalak megrajzolásának menetében.



5.1. ábra. A magassági pontok és a kézi szintvonal szerkesztés eredménye

Ma már a szintvonalak rajzolásának útja a fotogrammetria, amely egzakt elveken alapulva, szubjektív elemek nélkül képes megalkotni a felszín szintvonalas képét.

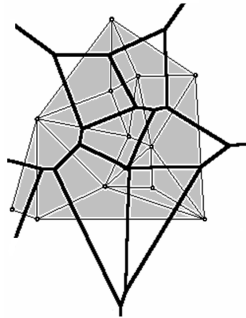
5.1.1. Vektoros domborzat leírás, a TIN modell

A TIN a triangular irregular network szavak kezdőbetűiből származik, utalva arra, hogy ez a leírási mód szabálytalan rácspontok hálózatából épül fel, amelyeket úgy kötünk össze, hogy a felszínt háromszögekkel fedjük le. Nem triviális, hogy a rendelkezésre álló pontokból miként fog előállni a háromszögek rendszere.

Voronoy-sokszögek, Delaunay-háromszögek

Tegyük fel, hogy szabálytalan elrendezésű pontjaink vannak a síkon. Minden pont köré szerkeszthető egy olyan sokszög, melynek belső pontjai közelebb vannak a kérdéses ponthoz, mint az összes többi ponthoz. Az ilyen tulajdonsággal rendelkező sokszögek konvexek és folytonosan töltik ki a síkot (ez is egyfajta tesszelláció).

A meghatározásból következik, hogy a sokszög oldalai merőlegesek a körülvevő pontot a többi ponttal összekötő egyenesekre és felezik azokat. Természetesen nem minden sugár felező merőleges egyenes lesz a sokszög része hanem csak azok, melyek metsződéseiből a zárt konvex sokszög létrejön. A sokszög oldalak egyben meghatározzák az egy-egy pontból figyelembe veendő szomszédos pontokat is, hiszen csak azok a pontok befolyásolják a sokszög kialakulását, amelyekre menő sugarak felező merőlegesei részei a sokszögnek. Ha minden pontot összekötünk a



5.2. ábra. A magassági pontokat kis fekete karikák jelzik. A vastag vonal mutatja a Voronoy-sokszögeket, és vékony vonalak a pontokat összekötő Delaunay-háromszögeket. Ismerjük fel, hogy ha gráfokkal reprezentáljuk a magassági pontokat, és az őket összekötő Delaunay-háromszögeket, akkor ezek duálisai a Voronoy-sokszögek.

róla a fentiek szerint figyelembe veendő szomszédos pontokkal, úgy síkbeli esetben, egyértelmű és bizonyos szempontból optimális háromszög-felbontást kapunk (5.2. ábra). Ezt a felbontást nevezik Delaunay háromszögelésnek. Térbeli esetben a szomszédosként meghatározott pontok összekötése egyértelmű, optimális tetraéder felbontást eredményez.

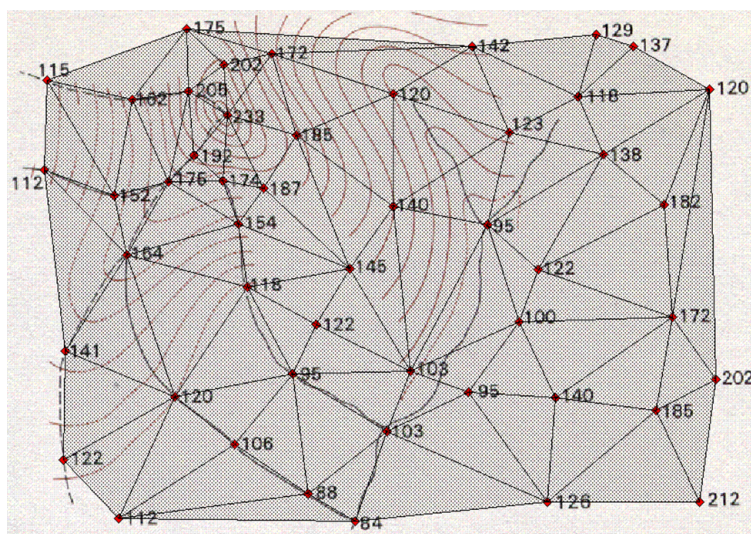
Alkalmazzuk a fenti módszert az előbbi, kissé tankönyvíví példára (5.3. ábra). Kössük össze háromszöghálóvá a magassági pontokat. Az így összeállított terepmodell egy kiragadott részletét mutatja a 5.4. ábra.

Tekintsünk egy valós példát, amelyben szabálytalan elhelyezkedésű, nagyszámú nagassági pontunk van. Ezekből szerkesszünk Delaunay-háromszögeket, majd jelenítsük meg perspektívikusan ábrázolva az eredményt (5.5. ábra)

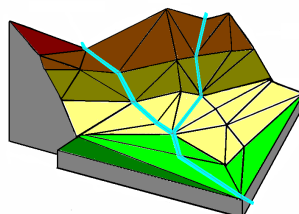
A TIN modell előnye, hogy nem használ interpolált pontokat, csak mérési eredményeket – éppen ebből következik szabálytalan mivolta –, így interpolációból eredő hiba nem terheli. A szabályos mintavételezéshez képest kevés adattal dolgozik, ami számos szempontból kifejezetten előny, ellenben megjelenítéskor tapasztalható, hogy az előállt terepfelszín, mesterséges, művi kinézetű. (Ne felejtsük el, hogy a pontok számát nem mi határozzuk meg, mint a 3D-s test modellezés esetében, hanem a magassági pontok száma dönti el a háromszög felbontás részletességét). A megjelenésnél nagyobb hátrány, hogy állandó felbontóképesség nem rendelhető hozzá, ezáltal nem tudjuk megmondani, hogy mi az legkisebb objektum, ami még kimutatható a felszínen.

5.1.2. Raszteres domborzat leírás, a DEM modell

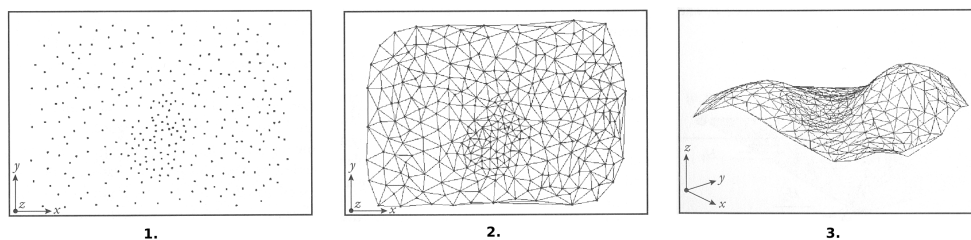
A DEM mozaikszó a a digital elevation model szavak kezdőbetűiből áll, amely digitális magasságmodellt jelent. Írjuk le a felszínt szabályos rácshálózatban meg-



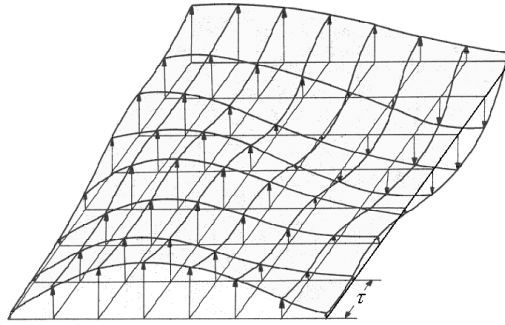
5.3. ábra. A magassági pontok alapján összeállított szabálytalan rács (TIN).



5.4. ábra. A háromszökből felépített felszín modellje. A színezés a kartográfiai hagyományoknak megfelelő, de az erős nagyítás miatt kicsit durva a kép. A világos, vastag vonalak a völgyekben futó folyókat mutatják



5.5. ábra. Magassági pontok (1), belőlük alkotott Delaunay-háromszögek (2), és ezek perspektívikus megjelenítése (3)

5.6. ábra. A τ rácállandójú Dirac- δ sorozattal mintavételezett felszín

adott magasságértékekkel. A magasság értékek egy a felszín leíró függvénynek egy τ rácállandójú Dirac-impulzus sorozattal való szorzásával kaphatók meg (5.6. ábra).

A τ mintavételi távolság megválasztása kulcskérdés a létrejövő adatbázis gyakorlati felhasználhatóságára nézve. Vizsgáljuk meg, hogy mi történik a 5.1. ábrán bemutatott térképpel, ha mintavételezzük a magasság értékeket (5.7. ábra).

Várható volt, hogy nem lesz egyetlen magassági pont sem, ami bármelyik rácspontra ráesne. Márpedig most olyan felszínmodellt építünk, ahol minden rácspontban kell legyen magasság értékünk.

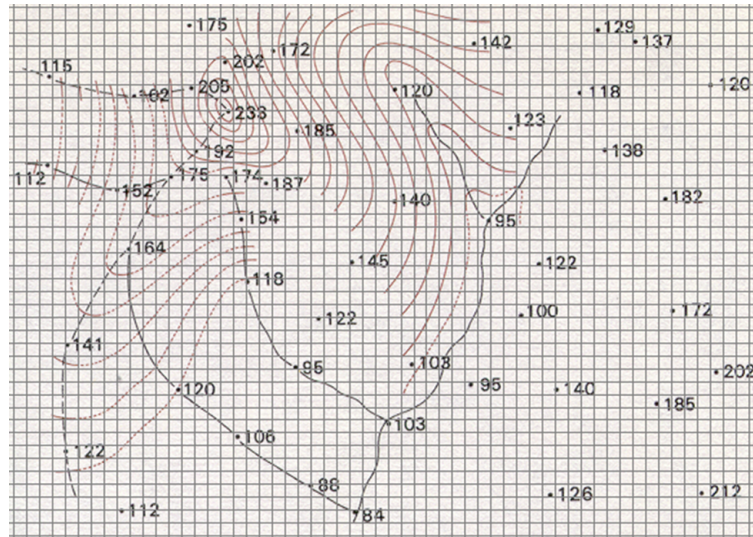
Grid számítási módszerek

Az IDW módszer. Az IDW (inverse distance weighting) inverz távolságok módszere egy gyors és egyszerű módja a grid pontok kiszámításának. A módszer azon alapul, hogy a grid értékének megállapításához a környező pontok magasság értékeit használjuk fel, de nem egyenlő súlyokkal, hanem a távolsággal csökkenő mértékben. Egy bizonyos hatástávolságon kívüli pontokat már nem is vesszük figyelembe (5.8. ábra). Hatása a spektrumra nem ideális, de előnyös tulajdonságai miatt előszeretettel használják.

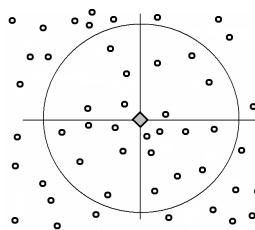
$$\hat{Z}_j = \frac{\sum_{i=1}^n \frac{Z_i}{h_{ij}^\beta}}{\sum_{i=1}^n \frac{1}{h_{ij}^\beta}}$$

$$h_{ij} = \sqrt{d_{ij}^2 + \delta^2}$$

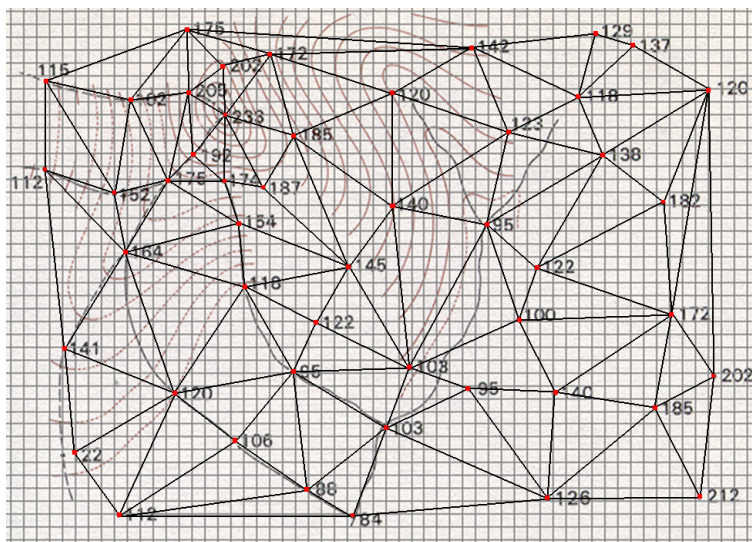
ahol h_{ij} a távolság a j -edik grid pont és az i -edik adatpont között, \hat{Z}_j a j -edik node interpolált értéke, Z_i a szomszédos pontok, d_{ij} a távolság a grid pont és a szomszédos pont között, β a súlykitevő, δ egy a simításra jellemző faktor.



5.7. ábra. τ rácállandójú rácsot fektettünk a térképre. Amint látható egyetlen magassági pont sem esik rá egyik rácspontra sem



5.8. ábra. A súlyozott inverz távolság módszere. A kis fekete körök mutatják a magassági pontokat, a kis szürke rombusz pedig az adott rácspontot. Ennek magassági értéknek a kialakulásába a közeli pontok nagyobb súllyal, míg a távolabbiak kisebb súllyal esnek latba



5.9. ábra. τ rácsállandójú rácsot fektetünk a TIN-t is tartalmazó térképre. A rácsponatok magasság értékeit az adott Delaunay-háromszögben elfoglalt pozíció adja meg. Ez az érték is interpolált

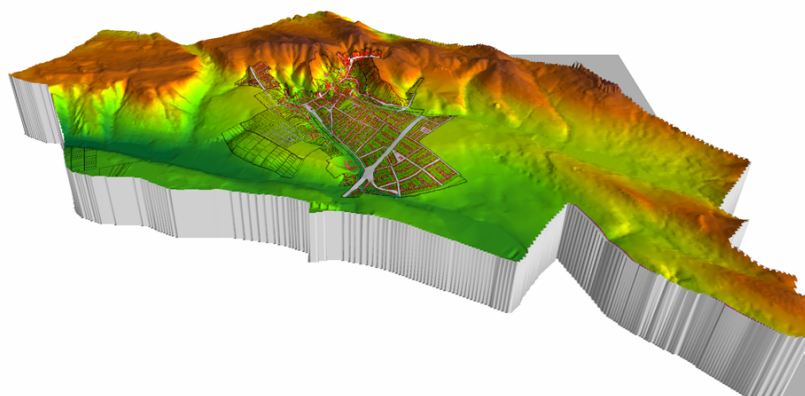
A TIN módszer. Más módszerrel is kaphatunk magassági értékeket a rácsponatok számára. Szerkesszük meg például a 5.3. ábrán látható háromszögrácsot.

Fektessük erre rá a 5.9. ábrán látható négyzetrácsot. Minden rácsponat be fog esni egy háromszög belsejébe. A rácsponatok magasság értékét egyértelműen meghatározza az a hely, amelyet egy adott háromszög belsejében elfoglal.

A DEM modell tulajdonságai

Valamelyik fenti módszerrel kiszámítottuk egy szabályos rácshálózatban a felszín magasságértékeit. Ezt nevezik DEM-nek, vagy digitális magasság modellnek. A DEM több előnyös tulajdonsággal is rendelkezik. A terep felszínét a DEM alapján megjelenítve, meglehetősen valóságos képet kapunk. Ennek ára, hogy igen nagy tömegű adatra van szükségünk. Előny, hogy állandó felbontóképességet rendelhetünk egy adott modellhez, ezáltal meg tudjuk mondani, hogy mekkora az a legkisebb objektum, ami még meglátszik a felszínen (ez a megállapítás persze erősen függ az adatnyerés pontosságától, amelyről itt nem ejtettünk szót). Hátrány, hogy szinte csak interpolált adatokkal dolgozik, ezért a rácsponatok adatai az interpolációs módszer hibáival terheltek. Előny, hogy az adatbázis szerkezete rasteres típusú, tehát úrfotó elemző és feldolgozó szoftverekbe is betölthető, sőt a képfeldolgozás eljárásai is alkalmazhatók rá.

A 5.10. ábrán egy valóságos domborzati modellt láthatunk, amelynek felbontóképessége kb. $5m$, amivel a mintavételi tétel miatt $10m$ -nél kisebb horizontális kiterjedésű felszín egyenetlenség nem mutatható ki. Ezért csak építészeti,



5.10. ábra. Egy valóságos terület 3D-s DEM modellje. Az ábrán melleleg egy település vázterképe is látható a felszínre, textúraként ráfeszítve

várostervezési célú felhasználást enged meg. Ha például a felhőszakadások alkalmával lezúduló csapadékvíz lefolyását kívánnánk modellezni, akkor pontosabb (nagyobb felbontású) domborzati modellt kellene megalkotni.

5.2. A 3D-s megjelenítés alapelvei

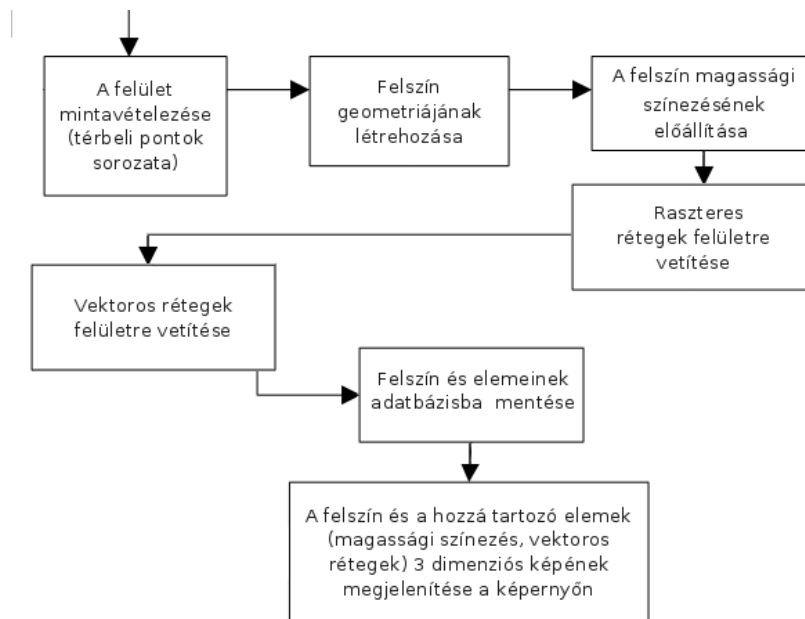
A felszín 3D-s modelljének megalkotásához felhasznált matematikai eljárások az eredmények 3D-s megjelenítéséhez nem elegendőek. Eddig ugyanis csak azt vizsgáltuk egy pontsorozatból miként alkothatunk háromszögeket, amelyekkel optimálisan lefedhetjük a modellezendő felszín. A megjelenítés ennél bonyolultabb feladat, elvégre nemcsak a két dimenziós felszín írjuk le, hanem annak a három dimenziós térben való elhelyezkedését, a rábocsátott fénysugár útját, amely visszaverődés, elnyelődés révén texturális tulajdonságokkal is rendelkezik.

Vizsgáljuk meg a 5.11. ábrát, amelyen a monitoron megjelenő 3D-s kép előállításának folyamatát láthatjuk a nyers magassági pontoktól a perspektívikus, textúrát is tartalmazó képig.

5.2.1. A virtuális színtér

A virtuális színtér elemei

- a tárgyak geometriai modelljei, úgy mint pontok, vonalak, vagy háromszögekből felépülő térháló, amelyek valós tárgyak felszínét közelítik
- a virtuális kamera, amelyen keresztül „láttatjuk” a virtuális színteret a képernyőn



5.11. ábra. A 3D-s kép kialakulásának folyamata a pontszerű magasság adatoktól a perpektívikus képig

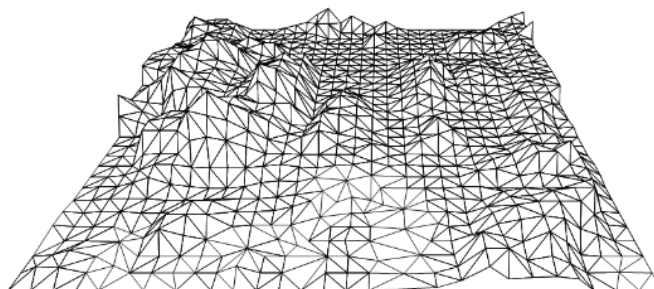
- a megvilágítási modell, amely a felületek színezését, árnyalását határozza meg. Elemei anyagtulajdonságok, fényforrások, és a felületek normálvektorai

5.2.2. A geometriai modellezés

A tárgyak felületét, geometriáját a virtuális térben háromszöghálókkal írjuk le (5.12. ábra). A hálók úgy adhatók meg, hogy felsoroljuk a hálót alkotó háromszögek csúcspontjait. Ezeket a csúcspontokat vertexeknek nevezzük, amelyek tartalmazzák a pont térbeli koordinátáit (x, y, z) a tárgy saját koordinátarendszerében megadva. Ennek a rendszernek az origója általában a tárgyon belül helyezkedik el, tengelyeinek iránya pedig a tárgy orientációjával párhuzamos (a Föld esetében ez egy geocentrikus koordináta rendszer). Egy vertex nemcsak a pontok koordinátáit, hanem egyéb segédkoordinátákat is tartalmazhat, mint például a felület adott pontbeli normálvektorát (nx, ny, nz) , amit a megvilágítási modell használ, valamint a textúra koordinátákat (u, v) , amelyek a textúra felületre feszítéséhez szükségesek.

5.2.3. Modellek elhelyezése a színtérben

Színtér építéskor a modelleket – amelyeket saját koordinátarendszerükben adtunk meg – el kell helyezni a virtuális világ koordinátarendszerében. Ezt a koordináta



5.12. ábra. A felszín közelítése háromszögekkel

transzformációt az affin transzformációkkal végezhetjük el (bevezetése a 2.1.3 fejezetben olvasható). Az affin transzformáció műveletei eltolás, tengelyek körüli forgatás, skálázás.

Geometriai transzformációk

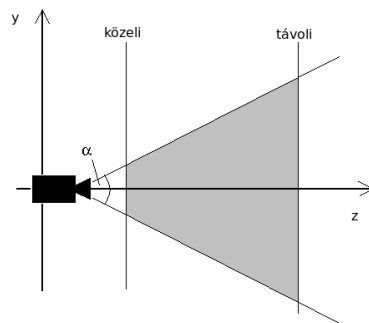
A modellek elhelyezése a virtuális világban és perspektivikus vetítése transzformációk alkalmazásával történik, amelyeket vertexenként kell végrehajtani. Egy-egy ilyen transzformációt egy 4×4 -es mátrix reprezentál, amely csak úgy alkalmazható egy vertexre, ha a térbeli koordinátáit homogén, négy koordinátás alakra hozzuk (amelyre azért van szükség, hogy az eltolás is leírható legyen mátrixokkal). Az Euklideszi (x, y, z) pont négyes vektorként $(x, y, z, 1)$ alakban adható meg. Jelölje valamely \mathbf{M} transzformáció eredményét a következő mátrix szorzás: $\mathbf{v}' = \mathbf{M} \cdot \mathbf{v}$.

Ha több egymás utáni transzformációt hajtunk végre, akkor az első transzformáció mátrixát jobbról szorozzuk a következő transzformáció mátrixával ($\mathbf{M} = \mathbf{T}_1 \cdot \mathbf{T}_2 \cdots \mathbf{T}_n$). Ha például egy eltolást egy forgatás követ, akkor $\mathbf{v}' = \mathbf{M} \cdot \mathbf{v}$, az \mathbf{M} transzformációs mátrix a forgatás (\mathbf{R}) és az eltolás (\mathbf{T}) transzformációs mátriszából áll: $\mathbf{M} = \mathbf{R} \cdot \mathbf{T}$.

A virtuális kamera

A virtuális színtérben elhelyezett kamera képezi le a színteret egy kétdimenziós képre. A kamera paraméterei a következők:

- a kamera pozíciója a virtuális térben
- a kamera orientációja: a felfelé mutató és a nézeti irányvektorokkal adható meg
- a kamera nyílásszögének fele (α)
- közeli és távoli vágósíkok



5.13. ábra. A virtuális kamera és paraméterei (y, z, α) . A kamera a teljes térből csak egy csongagúla alakú területet érzékel (szürke terület). A kép előállításakor a kamera perspektivikusan vetíti a virtuális teret, amelynek következtében a látótér csongagúlaból origó középpontú téglatestté transzformálódik. A perspektivikus vetítést egy négy dimenziós mátrix végzi, amelynek paraméterei a közeli és távoli vágósík, valamint a kamera nyílásszöge (α)

- A kamera saját koordinátarendszerében a kamera van az origóban, a felfelé mutató vektor y irányú, a nézeti vektor z irányba mutat. (5.13. ábra)

5.2.4. A grafikus szerelőszalag

A grafikus szerelőszalag feladata, hogy előállítsa a 3 dimenziós virtuális színtér raszteres képét egy virtuális kamera adott beállításával, amely aztán megjelenik a képernyőn. A bemenő adatok a színtér tárgyainak felszíni geometriáját leíró háromszögháló, amelyeket vertexek sorozatai reprezentálnak. Egy adott tárgy vertexeinek a szerelőszalagra küldése előtt be kell állítani a kamerát, a fényrendszert, az adott tárgy anyagi tulajdonságait, és a transzformációit. A kimenő adat egy $n \times m$ -es képmátrix (ahol n, m a kép méretei), amelynek elemei RGB színmodellben megadott pixelek. A folyamat sorrendje tehát a következő: modellezési transzformáció, nézeti transzformáció, perspektív (projektív) transzformáció, képernyő transzformáció, vágás, homogén osztás, raszterizáció, megjelenítés a képernyőn.

Modellezési transzformáció

Ez a művelet, amint az előbbiekben említettük egy affin transzformáció (4×4 -es mátrix), amelyet egy objektum felületi hálójának vertexeire alkalmazva a tárgyat a saját koordinátarendszeréből a színtér koordinátarendszerébe viszi át. $\mathbf{v}' = \mathbf{M} \cdot \mathbf{v}$, ahol \mathbf{M} a modellezési transzformáció, \mathbf{v} pedig az objektum egy vertexe.

Nézeti transzformáció

Ez egy szögtartó és távolságtartó transzformáció, amely beforgatja és eltolja a színteret a virtuális kamerának megfelelő állásba, vagyis a kamera koordinátarendszerébe transzformálja azt. Egy forgatási és egy eltolási mátrix szorzatából épül fel. $\mathbf{v}' = \mathbf{N} \cdot \mathbf{v} = \mathbf{T} \cdot (\mathbf{R}_{xyz} \cdot \mathbf{v})$, ahol \mathbf{N} a nézeti, \mathbf{T} az eltolási, \mathbf{R}_{xyz} pedig a három tengely körüli forgatási transzformáció

Perspektív transzformáció

Ez a transzformáció végzi el a perspektivikus vetítést, ami egy középpontos vetítés az origóra nézve. Ennek következtében a látható csonkagúla alakú térrész téglatestté transzformálódik. $\mathbf{v}' = \mathbf{P} \cdot \mathbf{v}$, ahol \mathbf{P} a kamera által megadott vetítómátrix

Homogén osztás

A négyes mátrixokkal reprezentált transzformációk elvégzése után a vektorokat vissza kell alakítani a homogén koordinátás alakjukból 3 dimenziós euklideszi alakjukba, vagyis $(x, y, z, w) \Rightarrow (x/w, y/w, z/w)$, ahol $w \neq 0$.

Vágás

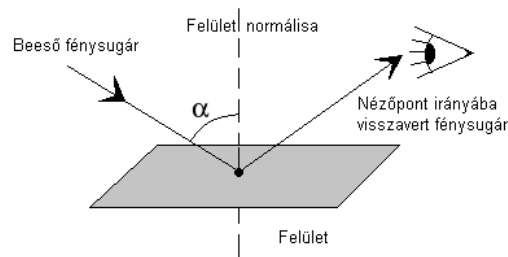
A homogén osztás után le kell vágni a transzformált háromszögeknek azokat a részeit, amelyek kívül esnek a láthatósági tartományon. A láthatósági tartomány egy téglatest, melynek oldalai a tengelyek által meghatározott síkokkal párhuzamosak. Osztályozza a háromszögeket aszerint, hogy hogyan helyezkednek el a tartományhoz képest. Ha egy háromszög teljes egészében belül van, akkor változatlan marad, ha teljes egészében kívül van, akkor eldobjuk, ha pedig metszi a tartomány határát, akkor lecserélődik a belülré eső részére, amely egy, vagy két háromszögből állhat.

Képernyő transzformáció

A képernyő koordinátarendszere: a képernyő mögötti egységnyi mélységű téglatest, a képernyő bal felső sarkában van az origó, a jobb alsó sarokban pedig az $(n, m, 0)$ pont, ahol $n \times m$ a képernyő felbontása. A z értékek 0 és 1 közé eshetnek, ahol 0 a képernyő síkjában, az 1-es pedig a képernyő mögötti síkban van. A képernyő transzformáció ebbe a térbe viszi át a téglatest alakú láthatósági tartományt.

Raszterizáció

Ebben a fázisban rá kell rajzolni a háromszög képét a képernyő háttérpufferére, amelynek tartalma a képszintézis befejezése után megjelenik a képernyőn. Ehhez a



5.14. ábra. A beeső fénysugár visszaverődése a felületről a nézőpont irányában. A fénysugár érkezik közvetlenül a fényforrásból, vagy másik felületről visszaverődve

háromszöget fel kell bontani képpont méretű darabokra, úgynevezett fragmentekre. A fragment egy homogén színezetű, 3 koordinátával rendelkező képpont méretű elem, melynek színét a megvilágítási modell és az esetleges textúrázás adja. A képernyő egy képpontjában az a fragment fog megjelenni, amelyik az adott pont mögötti fragmentek közül a legkisebb z -értékű.

5.2.5. A megvilágítás modellje

A megvilágítási modell a tárgyak színezetét, árnyalását határozza meg. A következő elemekből áll:

- Absztrakt fényforrások: olyan elemek, amelyek fényt bocsátanak ki, ami megvilágítja az egyes tárgyakat. A színtér elemei visszaverik a fényforrások fényét, ami aztán a szemünkbe jut (vagyis jelen esetben a virtuális kamerába), és emiatt látjuk azokat. A fényforrások típusai:
 - Pontszerű fényforrás: a virtuális tér egy adott pontján található, és sugárirányban bocsátja ki a fényt (pl. villanykörte). A fény intenzitása a fényforrástól távolodva csökken.
 - Környezeti (ambiens) fény: a tér minden pontján ugyanolyan erősségű homogén fény
 - Párhuzamos fénysugarak: adott irányú párhuzamos fénysugarak. (pl. napsugarak)
- Anyagtulajdonságok: meghatározzák, hogy a beérkező fény egyes komponenseit milyen mértékben veri vissza a tárgy felszíne.
- Felületi normálisok: meghatározzák, hogy a felületre beeső fény az egyes irányokba milyen intenzitással verődik vissza.

Diffúz felületi modell

Matt felületek esetén (pl. falfelület) a visszavert fény erőssége független a beeső fénysugár irányától, a diffúz felületek minden irányban egyforma erősséggel szórják a beeső fényt. A diffúz felületek által visszavert fény intenzitása:

$$L = L_{be}k_d \cos(\alpha),$$

ahol L_{be} a beeső fénysugár intenzitása, α a beesési szög, k_d a felület anyagának diffúz visszaverődési tényezője.

A spekuláris visszaverődési modell

Tükröződő felületek esetén a beeső fénysugár tükörirányban verődik vissza a legerősebben, és minél nagyobb mértékű az eltérés a tükörirány és a nézeti irány között, annál gyengébb a visszavert fénysugár intenzitása. A spekuláris felületek által visszavert fény intenzitása:

$$L = L_{be}k_s \cos^n(\delta),$$

ahol L_{be} a beeső fénysugár intenzitása, \mathbf{H} vektor: a nézeti irányvektor és a beeső fénysugár forrásának irányába mutató vektorok összegének az $1/2$ szerese, δ a \mathbf{H} vektor és a felületi normális egymással bezárt szöge, k_s a felület anyagának spekuláris visszaverődési tényezője, és n konstans egész szám

A képlet alapján L akkor maximális, ha a nézeti irány megegyezik a tüköriránnyal, ugyanis ekkor a \mathbf{H} vektor egybeesik a felületi normálissal, tehát a $\delta = 0$, és a $\cos^n(0) = 1$.

Komplex felületi modell

A természetben előforduló anyagok általában nem írhatók le tisztán diffúz, vagy spekuláris modellel, de jól közelíthetők ezek kombinációjával. $M = (k_d, k_s)$, ahol k_d az anyag diffúz, k_s pedig a spekuláris visszaverődési tényezője. A komplex felület által visszavert fény intenzitását úgy kapjuk, hogy összegezzük a diffúz és a spekuláris módon visszavert fény intenzitásokat. Az $M = (k_d, k_s)$ anyagú felület által visszavert fény intenzitása tehát $L = L_{be}k_d \cos(\alpha) + L_{be}k_s \cos^n(\delta)$.

Az árnyalási egyenlet egyszerűsített alakja

Az árnyalási egyenlet megadja, hogy a színtér absztrakt fényforrásainak fényét egy adott $M = (k_d, k_s)$ anyagú felület mekkora intenzitással veri vissza a nézeti irányban. Az egyszerűsített egyenlet csak a közvetlenül a fényforrásokból a felületre érkező fényekkel számol, és nem veszi figyelembe a többszörös visszaverődést (vagyis ha a fény egy másik felületről visszaverődve esik be), és a fényforrás leárnyékolását. A legtöbb grafikai szoftver ma még ezt az egyenletet használja, és külön algoritmussal számítja ki az árnyékokat, és a környezeti tükröződést.

Domborzati modellek megjelenítésekor a fényforrás általában a konvencionális északnyugati irányban van, vagyis a térinformatikában a domborzat árnyékolás térképészeti szabályai az irányadóak. Ettől természetesen el lehet térni, hiszen a fényforrás iránya változtatható. Az egyszerűsített árnyalási egyenlet:

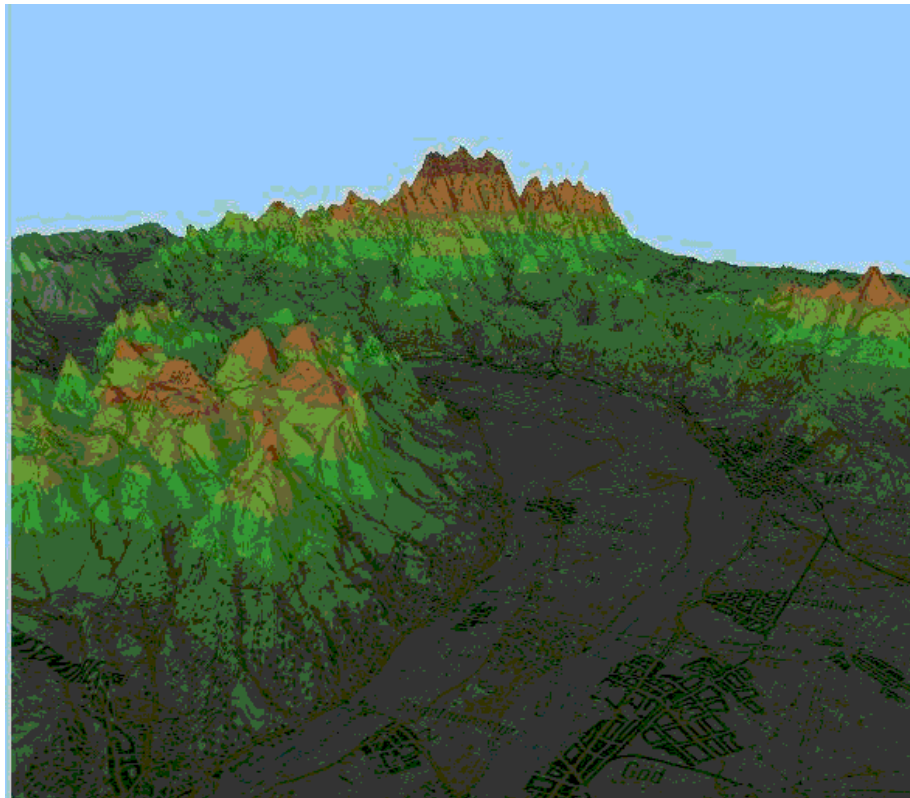
$$L = k_d L_a + \sum_{fi} (L_{fi} k_d \cos(\alpha) + L_{fi} k_s \cos^n(\delta)),$$

ahol $M = (k_d, k_s)$ a felület anyaga, L_a a környezeti fény intenzitása, L_{fi} az i -edik fényforrás intenzitása

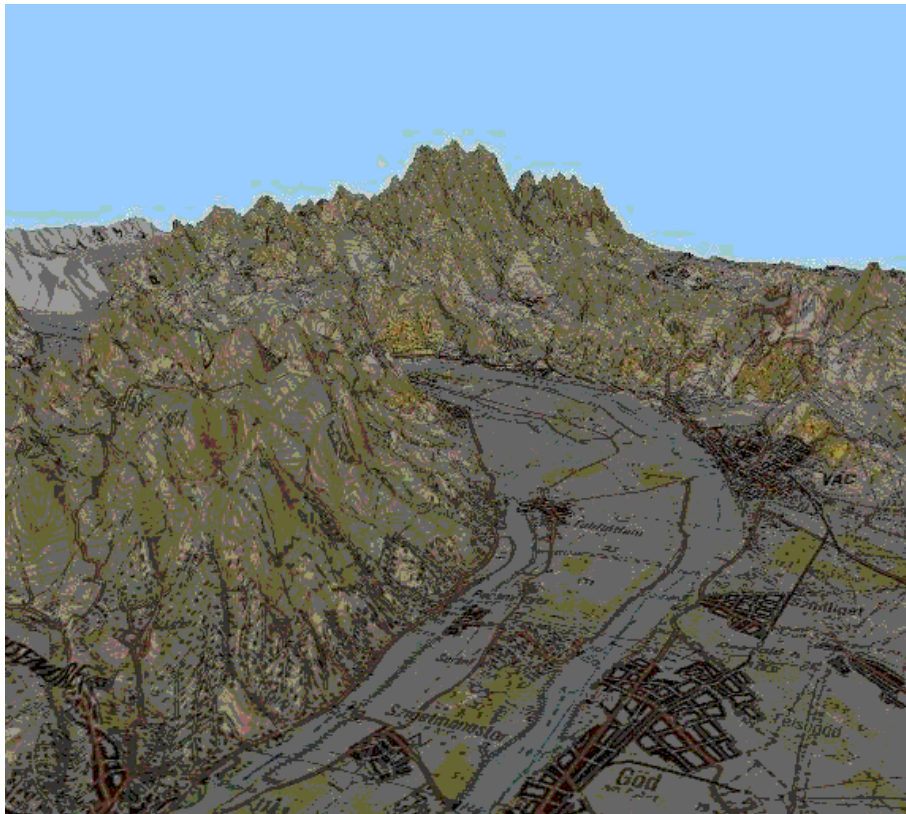
5.2.6. Egy térinformatikai példa

A 3D-s térinformatikai szoftverek a fentebb bemutatott elveken működnek. A 5.15. és 5.16 ábrákon valóságos domborzati adatokból generált felszín modelleket láthatunk. A számos megjelenítési lehetőség közül csak néhány látványosabbat mutatunk be. Mindkét ábra esetén ugyanazok a kamera beállítások érvényesek, amelyek a következők:

- eltolás
 - x irányban: 20
 - y irányban: 100
 - z irányban: 10
- forgatás
 - x tengely körül: 70 °
 - y tengely körül: -20 °
- nagyítás: 160%
- magassági torzítás: hatszoros



5.15. ábra. A Dunakanyar 3D-s megjelenítése a honvédség Térképszeti Kht adatai alapján. A színezés hagyományos magassági színezés szerint történt



5.16. ábra. A színezésen felül a terület 1:10.000 topográfiai térképe is, mint textura, a felületre feszítve látható

6. fejezet

Adatbányászati módszerek

6.1. Osztályozás, szegmentálás

Egy adathalmaz pontjainak az adatrekordok hasonlósága alapján történő diszjunkt csoportokba sorolását klaszterezésnek nevezzük. A csoportosítás jósága alapvetően két dologon múlik: a jó hasonlóság definícióján és egy jó algoritmuson, amely a hasonlóságon alapulva valamilyen kritériumok alapján megállapítja a klasztereket. Sokszor használjuk az osztályozás kifejezést is, ami nem egészen ugyanazt jelenti, mint a klaszterezés. Míg a klaszterezés nem felügyelt csoportosítás, addig az osztályozás felügyelt. Ebben az összefüggésben a felügyelt jelző azt jelenti, hogy a csoportok minőségi paraméterei előre definiáltak, míg a nem felügyelt esetben nem tudjuk, hogy milyen minőségi osztályok fognak létrejönni, milyen minőségi osztályba fognak tartozni az előálló csoportok, sőt ezek határai sem mindig tudhatók előre.

6.1.1. Távolság mátrix

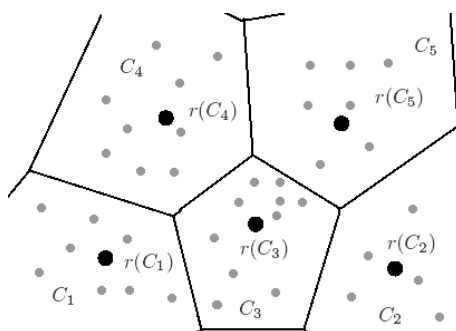
A hasonlóság definiálásának egy kézenfekvő módja az euklideszi távolság fogalom. Jelölje u_i, v_i az adatpontokat, és $d(u, v)$ az adatpontok közötti távolságot.

$$d(u, v) = \sqrt{\sum_{i=1}^d (u_i - v_i)^2}$$

Írjuk be az összes lehetséges adatpont közötti távolságot egy mátrixba, amelyet távolság mátrixnak nevezünk:

$$\mathbf{D} = \begin{pmatrix} d_{11} & d_{12} & \dots \\ d_{21} & d_{22} & \dots \\ \vdots & \vdots & \ddots \end{pmatrix}$$

Első közelítésben azt mondjuk, hogy egymáshoz hasonló pontokat azonos csoportba, klaszterbe sorolunk. A távolság mátrix alapján viszont ismerjük az adat-



6.1. ábra. Adatpontok (szürke körök), klaszter középpontok (fekete körök) és klaszterek határok

pontok páronkénti távolságát, így a távolságok alapján a hasonlóságra is következtetni tudunk. Azt mondjuk tehát, hogy azonos klaszterbe tartozó pontok egymáshoz közel vannak.

Ez a megfogalmazás elég tág határokat enged meg a klaszterek meghatározására, de valamennyi klaszterező eljárás háttérében megtalálható a távolság mátrix, illetve a klaszterek középpontjától való távolság.

6.1.2. Particionáló klaszterezés

Feltesszük, hogy a klaszterek egy vektortérben helyezkednek el. A klasztereket súlypontjukkal reprezentáljuk, vagyis a klaszterekhez tartozó adatpont-vektorok átlagával jellemezzük (6.1. ábra). Az algoritmus olyan C klaszter beosztást keres, ahol az adatpontok saját klaszterük $r(C_i)$ súlypontjától mért távolságának négyzetösszege minimális.

$$E(C) = \sum_{i=1}^k \sum_{u \in C_i} d(u, r(C_i))^2$$

Általában előre meg kell adnunk egy k klaszterszámot (vagyis, hogy hány csoportra szeretnénk bontani az adathalmazt). Válasszunk ezután k darab adatpontot. Ezután minden adatpontot a hozzá legközelebb eső klaszter-súlyponthoz tartozó klaszterbe sorolunk. A besorolás eredményeként kialakult új klaszterek súlypontjai lesznek az új klaszterek reprezentáns pontjai. A besorolás, súlypontszámítás lépéseit addig végezzük, amíg a súlypontok rendszere változik. Akkor állunk meg, amikor a klaszterek elemei és a klaszterek középpontjai már nem változnak az iteráció hatására.

6.1.3. Hierarchikus eljárások

A hierarchikus klaszterező eljárásokban a adatokat hierarchikus adatszerkezetbe (fába, dendogram) rendezzük. Az adatpontok a fa leveleiben helyezkednek el. A fa minden belső pontja egy klaszternek felel meg, és azokat a pontokat tartalmazza, amelyek a fában alatta találhatók.

Két alapvető hierarchikus eljárás létezik: az egyik a felhalmozó, a másik a lebontó. A felhalmozó eljárásban kezdetben minden adatelem egy klaszter, majd a legközelebbi klasztereket egyesíti az algoritmus, és a hierarchiában egy szinttel feljebb új klasztert alakít ki.

A lebontó eljárásban kezdetben egyetlen klaszter létezik, amelybe minden adatpont beletartozik, majd ezt tovább osztjuk. Az újabb klaszterek az előző finomításai lesznek. Az eljárások akkor állnak meg, amikor vagy elérnek egy előre megállapított klaszter számot, vagy a klaszterek közötti távolság egy előre megállapított mértéknél kisebbé válik.

6.1.4. Képek klaszterezése

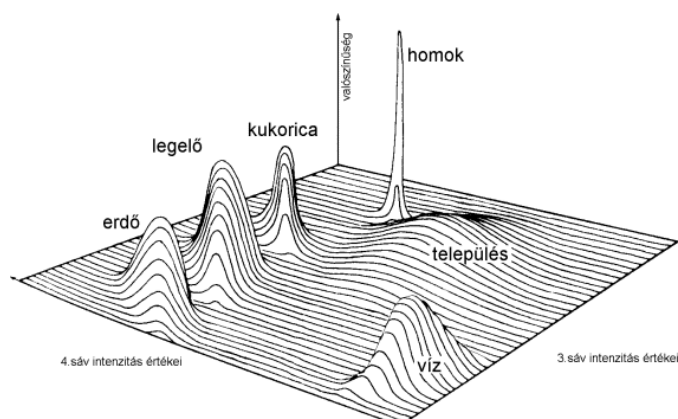
A képek osztályozásakor az a célunk, hogy a pixeleket tematikus kategóriákba soroljuk az intenzitás értékeik alapján, mintegy spektrális osztályokat létrehozva. Kétféle osztályozási módszert különböztetünk meg. Az egyik a nem felügyelt (unsupervised classification), a másik a felügyelt (supervised classification) osztályozás. A nem felügyelt esetben megelégszünk spektrális csoportok létrejöttével, amelyeket megkísérlünk megfeleltetni valamely tematikus kategóriának.

A klaszterezéskor kiindulhatunk fix számú osztályból is, de megadhatunk egy környezetet is, például Euklideszi távolság alapján, amelynek túllépése esetén új klaszter jön létre.

ISODATA eljárás

Az ISODATA eljárás a klaszterek középpontjait keresi meg. Az eljárás a következő módon működik:

1. Válasszunk ki klaszter középpontokat kiindulásul
2. A pixeleket a hozzájuk legközelebb eső középpontú klaszterbe soroljuk be
3. Az újfent besorolt pontok figyelembe vételével kiszámítjuk az új klaszter középpontokat, amik ettől kisebb nagyobb mértékben megváltoznak
4. Az eljárás leállítását a középpontok mozgása határozza meg. Addig folytatjuk az eljárást (a 2.-es pontra ugorva), amíg a középpontok helyzete nem változik, pontosabban a mozgásuk egy bizonyos küszöbérték alatt marad



6.2. ábra. A különböző anyagok reflektanciáinak eltérése alapján következtetni lehet az anyagminőségre [7], feltéve, hogy a tematikus osztályok között nincs átfedés

Felügyelt osztályozás

Felügyelt osztályozáskor a kép pixeleit tematikus kategóriákba soroljuk be a tematikus kategóriák mintáiból gyűjtött adatok alapján (vagyis előre tudjuk, hogy hány osztályunk van, és azoknak mik a minőségi jellemzői). A tematikus kategóriák mintaterületeinek kijelölése történhet terepi bejárás alapján, vagy független, más forrásból származó adatok alapján vizuális interpretációval. A mintaterületeket gyakran nevezzük tanuló területnek.

Összehasonlítva a kétféle osztályozási módot látható, hogy a felügyelt osztályozáskor a tematikus kategóriák meghatározása után osztályozzuk a képet, míg a nem felügyelt osztályozáskor a klaszterezés után feleltetjük meg az egyes klasztereket a tematikus kategóriáknak.

Ezen osztályozások fizikai háttérét az a megfigyelés adja (amelyet akár a távérzékelés alapösszefüggésének is nevezhetünk), hogy egyes tematikus osztályok, minőségi kategóriák pixelei jellegzetes csoportokat alkotnak, amint azt a kategóriák reflektancia értékeinek eloszlása is jól mutatja a 6.2. ábra. (Feltételezhetően annak tudható be a normális eloszlás, hogy a visszaverődés jelensége sokféle folyamat együtteséből tevődik össze. Márpedig ezek akármilyen eloszlást is kövessenek, az összegük eloszlása közelíteni fog a standard normális eloszláshoz. Ez a centrális határeloszlás tétele.)

A tapasztalat azt mutatja, hogy a különböző frekvenciasávokra másként reagálnak ezen minőségi kategóriák anyagai, így minél több frekvenciasávban állnak rendelkezésünkre képek, annál több minőségi kategória megállapítása válik lehetségessé. Ez a körülmény teremti meg az értelmét a többszáz frekvenciasávban működő hiperspektrális távérzékelés számára.

6.2. Dimenzió csökkentés PCA-val

A klaszterező, osztályozó eljárások valamennyi rendelkezésre álló adatot figyelembe vesznek a csoportosítás elvégzéséhez. Az algoritmusok működéséből világosan látható, hogy a távolságok és a klaszter középpontok állandó számítása rendkívüli számítási igényt támaszt, különösen olyankor, amikor nagyon sok dimenziós az adatrendszer. A multispektrális távérzékelés néhány frekvencia sávja még kezelhető méretű adatrendszer, de ha figyelembe vesszük a ma egyre nagyobb jelentőségű hiperspektrális távérzékelés több száz frekvencia sávját, akkor belátható, hogy az osztályozás hagyományos eljárásainak alkalmazása erre az adatmennyiségre irreális számítási időt eredményezne.

Ilyen esetekben alkalmazunk dimenzió csökkentő eljárásokat. A dimenzió csökkentés egyik lehetséges módja a főkomponens analízis (Principal Component Analysis), amely a többváltozós matematikai statisztika egy széles körben elterjedt eljárása. A következőkben a főkomponens analízis valószínűségi megfogalmazását adjuk meg, de lehetséges algebrai megoldás is, amelyet a fizikusok használnak előszeretettel a mechanikában (főtengely transzformáció néven).

A valószínűségi megfogalmazás a következő: legyen p számú megfigyelési egységünk, amelyek egyenként n számú adatot tartalmaznak (p számú megfigyelési vektorunk van).

\mathbf{x}^1	\mathbf{x}^2	...	\mathbf{x}^p
x_1^1	x_1^2	...	x_1^p
x_2^1	x_2^2	...	x_2^p
\vdots			\vdots
x_n^1	x_n^2	...	x_n^p

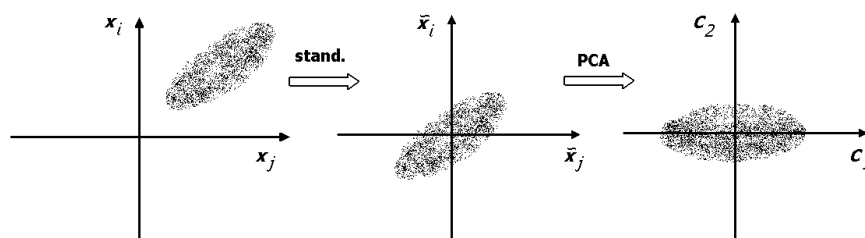
Tekintsük az \mathbf{x}^j vektorokat valószínűségi változóknak, a vektorok elemeit a valószínűségi változók realizációinak. Standardizáljuk a változókat:

$$\tilde{x}_i^j = \frac{x_i^j - \bar{x}^j}{s^j}$$

ahol \bar{x}^j a j -edik vektor elemeinek átlaga (a várható érték becslése), és s^j az empirikus szórása. Így tehát 0 várható értékűvé, és 1 szórásúvá tettük a valószínűségi változóinkat. Ezek után számítsuk ki az adatrendszerünk korrelációs mátrixát:

$$\mathbf{R} = \begin{pmatrix} r_{11} & r_{12} & \dots & r_{1p} \\ r_{21} & r_{22} & \dots & r_{2p} \\ \vdots & \vdots & \ddots & \\ r_{p1} & r_{p2} & \dots & r_{pp} \end{pmatrix}$$

ahol r_{ij} az i és j -edik megfigyelési egységek korrelációs együtthatója. Határozzuk meg a korrelációs mátrix sajátértékeit és sajátvektorait, vagyis oldjuk meg a következő sajátérték egyenletet:



6.3. ábra. A folyamat geometria jelentése: a standardizálás 0 várható értékűvé és 1 empirikus szórásúvá teszi a változókat, vagyis a pontfelhőt betolja az origóba, majd elforgatja a legnagyobb variancia irányába, ami az első főkomponens

$$\underline{\mathbf{R}}\mathbf{v} = \lambda\mathbf{v}$$

A sajátértékek $\lambda_1 < \lambda_2 < \dots < \lambda_p$, a sajátvektorok $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_p$. Számítsuk ki a főkomponenseket a következő módon, legyen a j -edik főkomponens a következő:

$$C_i^j = \sum_p x_i^p v_p^j$$

ahol $i = 1, n$ és $j = 1, p$.

A főkomponensek ortogonális rendszert alkotnak, vagyis korrelálatlanok, azaz korrelációs mátrixuk

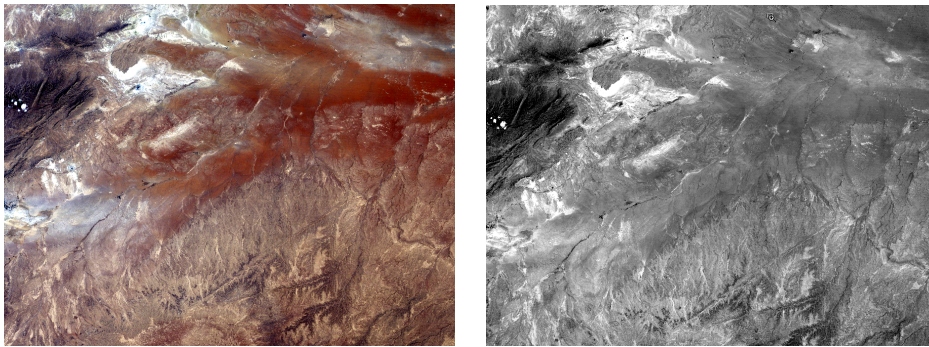
$$\underline{\mathbf{R}}_{\mathbf{C}} = \begin{pmatrix} \lambda_1 & & & 0 \\ & \lambda_2 & & \\ & & \ddots & \\ 0 & & & \lambda_p \end{pmatrix}$$

A $\underline{\mathbf{R}}_{\mathbf{C}}$ fontos tulajdonsága, hogy a főkomponensek és a standardizált változók összvarianciája azonos:

$$\sum_{j=1}^p \lambda_j = \sum_{i=1}^p \tilde{s}_i^2 = \sum_{j=1}^p s_j^2 = p$$

Amint látható, a főkomponensek kiszámításával nagymértékben átrendeztük a varianciákat, mivel (ha ez lehetséges volt), összevontuk őket az első (néhány) főkomponensbe. Az eljárás főbb mozzanatainak geometria jelentését a 6.3. ábra mutatja.

Abban az esetben, ha például az első főkomponens képes magába sűríteni a megfigyelési egységek varianciáinak nagy részét, akkor megtehetjük, hogy az egész adatrendszert csak az első főkomponensével helyettesítjük. Így jelentős mértékben csökkentettük az adatrendszer dimenziószámát, ezzel adatszámát, azaz meggyorsítottuk, megkönnyítettük egy soron következő eljárás, például a klaszter



6.4. ábra. Egy LANDSAT kép RGB sávjainak és az összes sáv első főkomponensének összehasonlítása.

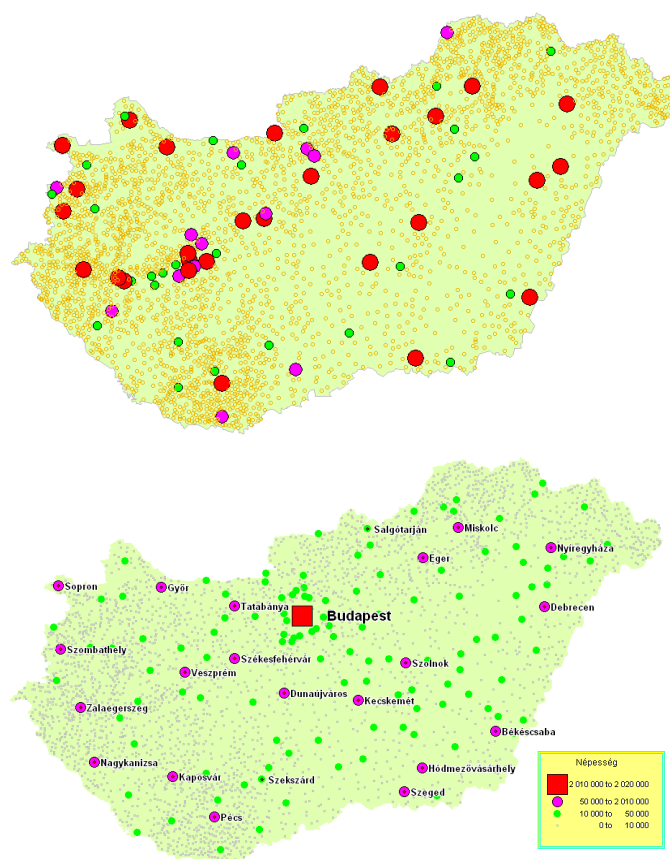
analízis működését. A 6.4. ábrán egy úrfotó példát láthatunk egy LANDSAT kép RGB sávjainak és az összes sáv első főkomponensének összehasonlítására.

Felmerülhet a kérdés, hogy mikor nem használható az első főkomponens a teljes adatrendszer helyettesítésére? Ha a korrelációs mátrix diagonális, vagyis a változók korrelálatlanok, akkor biztosan nem. Ebben az esetben minden további számítás értelmetlen.

Egy másik kézenfekvő kérdés, hogy ha például a megfigyelési egységeink mérési értékek (pl. digitális képek, fizikai mennyiségek), akkor miért tekintjük őket valószínűségi változóknak. Ennek pusztán az az oka, hogy először a többváltozós matematikai statisztika használta ezt az eljárást dimenziószám csökkentésre. A probléma leírható algebrai módszerekkel is, mint fentebb említettük.

6.2.1. Tematikus térkép és osztályozás

Most egy pillanatra visszatérünk a vektoros térinformatikához, mivel a klaszterezésnek és a főkomponens analízisnek ebben a körben is van jelentősége. Az adatok osztályozásának hétköznapi módja a térképszetben, geoinformatikában a tematikus térkép készítése. Ilyenkor a csoportosítás eredményét tematikus térképen jelenítjük meg, vagyis a csoportok állapotát szimbolizáló értékeket (fizikai mennyiségek, kategória változók, stb.) valamilyen grafikus szimbólum vagy stílus (vonaltípus, kitöltési mintázat, szín) formájában mutatjuk meg a térképen. A geoinformatika gyakorlati alkalmazói – statisztikusok, elemzők, szociológusok, stb. – jól ismerik az adatok láttatásának ezt a módját. Az informatikai piacon elérhető szoftverek kiváló tematikus térkép készítő képességekkel rendelkeznek, de az objektumok több dimenziós leírásából származó nehézségeket általában úgy oldják meg, hogy csak egyetlen változó értékeit jelenítik meg, ami persze dimenziócsökkentés, de egyáltalán nem optimális módon, hiszen az adatrendszer varianciáira nincs tekintettel, önkényesen hagy figyelmen kívül adatokat. Bizonyos esetekben természetesen ez a módszer is szolgáltat megfelelő eredményeket, de

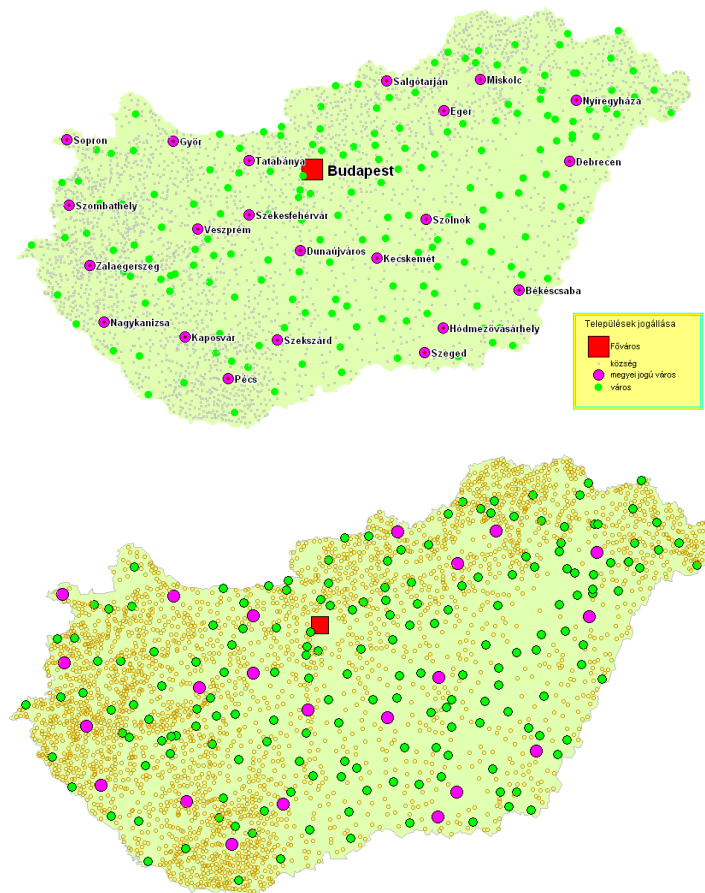


6.5. ábra. A településenkénti vendégéjszakák számát mutató tematikus térkép (felső térkép). Tematikus csoportosítás a települések népessége szerint (alsó térkép) [12]

általános osztályozási célra nem alkalmas.

Nyilvánvalóan megbízhatóbb az a csoportosítás, amely a tematikus felosztást nem egy önkényesen kiragadott adatféleség alapján végzi, hanem optimális adatvesztés révén hoz létre egy „szintetikus” adatsort, és ez alapján hozza létre a hagyományos tematikus térképet. Hasonlítsuk össze a csak egyetlen megfigyelési egység adatai alapján történő különböző tematikus csoportosítások eredményét (6.5. és 6.6. ábrák).

Amint látható volt, a főkomponensek dimenziótlan számok, emiatt fizikai jelentésük sem nyilvánvaló. Egyes esetekben hipotézist állíthatunk fel arra vonatkozóan, hogy az első főkomponens az adatok háttérében meghúzódó valamilyen fizikailag is azonosítható, közös ok változó. Máskor meg kell elégednünk az első főkomponens variancia tartalmára épülő előnyökkel anélkül, hogy azonosítani tudnánk a háttérben lévő hatókat.



6.6. ábra. Tematikus csoportosítás a települések jogállása szerint (felső térkép). A települések statisztikai adataiból számított első főkomponens szerinti tematikus térkép (alsó térkép). Nem véletlen, hogy az így kapott tematikus térkép megegyezik a települések jogállását mutató térképpel, elvégre a várossá nyilvántás számos település paraméter együttes mérlegelése után történik meg [12]

A. Függelék

Matematikai összefoglaló

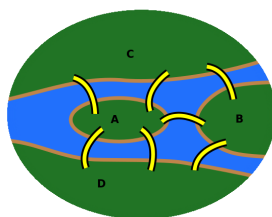
Ebben a részben tömören összefoglaljuk azokat a legfontosabb matematikai fogalmakat, amelyek a térinformatikában valamilyen formában szerephez jutnak. Célunk voltaképpen az, hogy ha az olvasó nem emlékezne a szöveg törzsében felbukkanó valamelyik matematikai fogalomra, annak rövid bemutatását itt könnyen megtalálja. Éppen azért a függelék kalkulus logikájú, így nem tartalmaz néven nevezett tételeket, bizonyításokat. Akit komolyabban érdekel valamelyik matematikai szakterület, azok az ajánlott irodalomban találhatnak útmutatást az elmélyültebb tanuláshoz.

A.1. Gráfelmélet

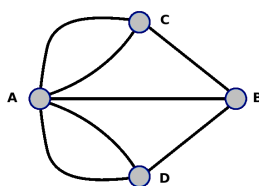
Az első gráfelméleti munka a Szentpétervári Tudományos Akadémia közleményeiben jelent meg 1736-ban. A szerző Leonhard Euler (1707-1783) svájci matematikus, aki ekkor az akadémia meghívására Oroszországban dolgozott. Königsberg (ma Kalinyingrad) a Balti-tenger közelében a Pregel folyó két partján fekszik. A folyón itt két sziget is van, amelyeket hidak kötnek össze a partokkal, amint az A.1. ábrán látható. Akkoriban a folyó A és B szigeteit hidak kötötték össze egymással és a partokkal is (C, D). A Königsbergiek kedvenc kérdése volt, hogy valaki otthonról indulva tehet-e olyan sétát, hogy minden hídon pontosan egyszer menjen át, és a séta végén hazaérjen.

Euler teljes általánosságban oldotta meg a feladatot. A feladatban ugyanis, mindössze a hidakon való áthaladás, valamint a kiindulópontba való visszatérés. Ezért Euler egy olyan ábrát készített, melyben a városrészeknek pontok, a hidaknak a pontokat összekötő vonalak (nem feltétlenül egyenesek) felelnek meg (A.2. ábra).

Az ilyen pontokból és vonalakkból álló alakzatokat gráfoknak nevezzük. A pontok a gráf csúcsai (node, vertex), a vonalak a gráf élei (edge). A kérdés tehát így fogalmazható meg: valamely csúcsból kiindulva bejárhatjuk-e a gráf éleit úgy, hogy minden élen pontosan egyszer haladjunk át, és végül a kiindulási pontba érjünk vissza?



A.1. ábra. A königsbergi hidak sematikusán ábrázolva



A.2. ábra. A königsbergi hidak gráfja

Ha valaki a kérdéses útvonalon halad, és eljut valamely csúcsba, akkor onnan, más útvonalon, ki is kell lépnie, és végül utolsó lépésként belép a kezdőpontba. Így minden pontba a sétálóknak ugyanannyiszor kell belépni, mint kilépni. A feladat megoldhatóságához tehát szükséges, hogy minden pontban páros számú él találkozzon. Ez a königsbergi hidak problémája esetében nem teljesül, ezért a feladat nem oldható meg.

A feladat megoldásában lényeges volt az egy csúcsban találkozó élek száma. Ezt a számot a továbbiakban a csúcs fokának, fokszámának nevezzük.

Jelöljük a G gráf (A.3. ábra) pontjai halmazát $V(G)$ -vel. A pontokat általában kis- vagy nagy betűkkel u, v, \dots , vagy akár egyszerűen számokkal jelöljük. A G gráf élei halmazának szokásos jelölése $E(G)$.

$$V(G) = \{v_1, v_2, v_3, v_4, v_5\}$$

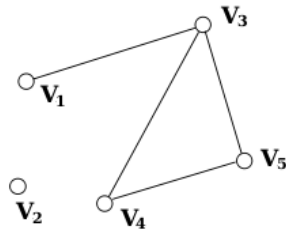
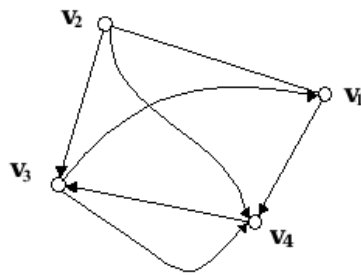
és

$$E(G) = \{(v_1, v_3), (v_3, v_4), (v_3, v_5), (v_4, v_5)\}.$$

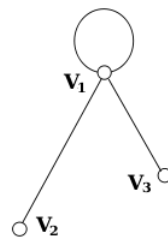
Egy élt két végpontja megadásával jelölünk. Amennyiben a gráf irányítatlan, úgy a végpontok neveinek sorrendje lényegtelen.

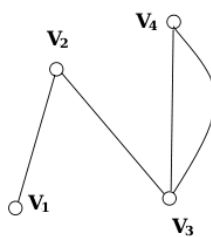
Irányított gráfok

Az ún. irányított gráfok esetében az élek sorrendje lényeges. Az éleket egyenes szakasznak, vonalláncnak, vagy görbe vonalnak is ábrázolhatjuk (A.4. ábra).

A.3. ábra. A G gráf csúcsai és élei

A.4. ábra. Példa irányított gráfra

A.5. ábra. Példa hurkot is tartalmazó gráfra: $V(G) = \{v_1, v_2, v_3\}; E(G) = \{(v_1, v_1), (v_1, v_2), (v_1, v_3)\}$



A.6. ábra. Példa többszörös élt is tartalmazó gráfra

Hurok

Előfordulhat olyan él is, melynek mindkét végpontja ugyanaz a pont. Az ilyen élt huroknak nevezzük (az A.5. ábrán ilyen a v_1 pont).

Többszörös élek

Két csúcs között több él is húzhatunk, ezt többszörös élnek nevezzük. Ilyenek az A.6. ábrán a v_3 -t és a v_4 csúcsokat összekötő élek. A gráfok ábrázolásakor két él az ábrán metszheti egymást, de ez a metszéspont a gráfnak nem csúcsa.

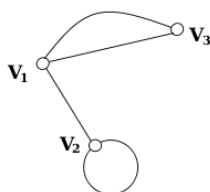
Szomszédsági mátrix

A G gráfhoz hozzá tudunk rendelni egy őt leíró mátrixot. $i = 1, 2, \dots, m$ a mátrix sorainak száma, $j = 1, 2, \dots, n$ a mátrix oszlopainak száma. Ha a mátrixnak m sora és n oszlopa van, $m \times n$ típusú mátrixnak nevezzük. A mátrixban levő a_{ij} számok a mátrix elemei. Az a_{ij} elem az i -edik sor j -edik oszlopának eleme. Ha $m = n$, a mátrixot négyzetesnek mondjuk.

$$\begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & & & \\ a_{m1} & a_{m1} & \dots & a_{mn} \end{pmatrix}$$

Nézzük meg a három csúcspontú G gráfot. A gráfot ismerjük, ha meg tudjuk mondani, hogy melyik pont melyik ponttal van éllel összekötve, és hányszoros éllel. Hozzunk létre egy mátrixot, ahol írjuk be mindegyik mezőbe az azt meghatározó csúcsokat összekötő élek számát. Az A.7. ábrán látható G gráf szomszédsági mátrixának nevezzük az (A.1) számú mátrixot. E mátrix ismeretében meg tudjuk rajzolni a gráfot.

$$\begin{pmatrix} 0 & 1 & 2 \\ 1 & 1 & 0 \\ 2 & 0 & 0 \end{pmatrix} \quad (\text{A.1})$$



A.7. ábra. A G gráf szomszédsági viszonyait az (A.1) számú szomszédsági mátrix írja le

Általában egy n vertexű gráf szomszédsági mátrixa az $A = (a_{ij})$ egy $n \times n$ -es mátrix, melyben a_{ij} a v_i és v_j csúcsokat összekötő élek száma, $i, j = 1, 2, \dots, n$. Így az A elemei nem negatív egész számok. A főátlóban levő elemek összege a gráfban levő hurkok számát adja. Világos az is, hogy az irányítatlan gráf szomszédsági mátrixa szimmetrikus a főátlóra, vagyis $a_{ij} = a_{ji}$, míg az irányított gráf mátrixa nonszimmetrikus (pl. A.4. ábra). Ennek szomszédsági mátrixa az (A.2) mátrix.

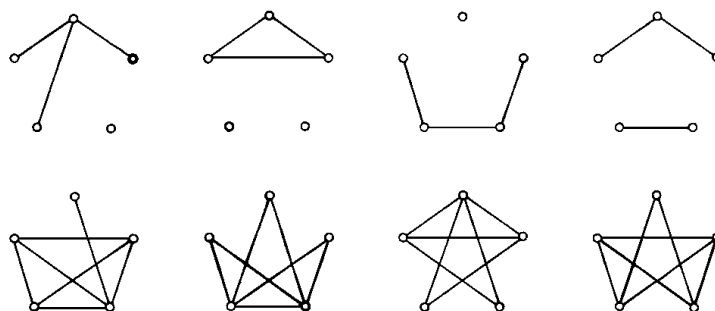
$$\begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix} \quad (\text{A.2})$$

Ha egy gráf pontjait más sorrendben indexezzük, akkor a szomszédsági mátrixban bizonyos sorok és oszlopok felcserélődnek, míg a gráf ugyanaz marad. A mátrix adatszerkezeti szempontból a programozási nyelvekben jól ismert kétdimenziós tömb.

A gráf egy csúcsában található élek számát a csúcs fokának nevezzük. Ez egy nem negatív egész szám (amely 0 is lehet). Ez esetben a pont izolált pont, nincs összekötve egyetlen más ponttal sem (pl. az A.3. ábrán a v_2 pont). Mivel minden él két ponthoz – vagy hurok esetében ugyanazon ponthoz kétszer – csatlakozik, így a fokszámok összege az élek számának kétszerese. A gráf fokainak összege páros szám, tehát a páratlan fokú pontok száma páros.

Élmátrix

A gráfok megadásának másik módja az élmátrix, mely a gráfok éleit tárolja, például egy $G[1..E]$ egydimenziós tömbben, ahol E az élek száma. Matematikai szempontból ez sorvektor, vagyis olyan mátrix, melynek csak egy sora van. A G elemei számpárok, pl. $G[k] = (v_i, v_j)$, vagyis a k -edik él a v_i és v_j csúcsokat köti össze. Itt nehezebb megállapítani két pont összekötöttségét, viszont könnyebb, pl. az összes élt felsorolni. Ritka kapcsolati mátrix esetén ez a forma könnyebben programozható.



A.8. ábra. Gráfok (felső sor) és komplementeik (alsó sor)

Súlyozott gráf

Az élekhez lehet számot is rendelni ez a súlyozott gráf, mátrixa a súlymátrix. Ilyen, pl. egy csővezetékrendszer, ahol megadjuk az egyes csövek áteresztő képességét. Úthálózatokra vonatkoztatva, az élhez rendelt súly lehet akár az élen való áthaladás ellenállása, amely így egyirányú utcák esetben a csúcspontok közötti távolság (megengedett haladási irányban történő mozgás esetén), vagy végtelen nagy (a kötelező haladási irányal szemben).

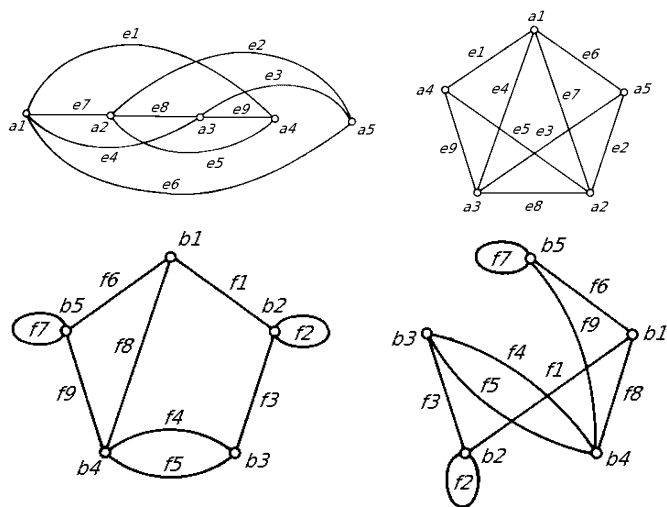
Komplemeneter gráf

Egy adott gráfhoz rendeljünk egy másik gráfot úgy, hogy a pontok halmaza mindkét gráfban ugyanaz, de a második gráfban két pontot akkor és csak akkor kötünk össze, ha az első gráfban a két pont nincs összekötve. Ezt a gráfot az eredeti G gráf komplementer gráfiának nevezzük és \bar{G} -sal jelöljük (A.8. ábra).

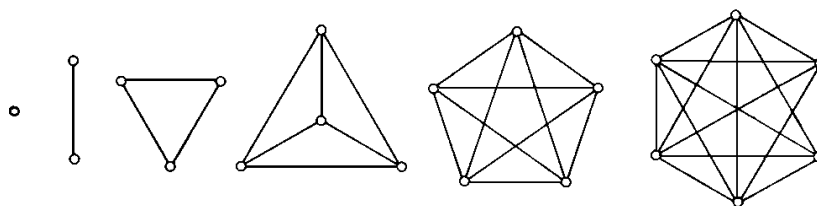
Izomorfia. Ha a G_1 gráf minden pontjának és élének megfeleltethető a G_2 gráfnak pontosan egy pontja illetve éle, valamint a G_2 gráf minden pontjának és élének megfeleltethető a G_1 gráfnak pontosan egy pontja illetve éle, akkor a gráfok izomorfak (A.9. ábra).

Teljes n-gráfok

Vizsgáljuk most meg egy egyszerű gráf pontjainak fokszámát. Csak az az eset érdekes, ha a gráfnak legalább két pontja van. Ha a gráf n -pontú, akkor mindegyik pont fokszáma $0, 1, 2, \dots, n - 1$ lehet. Ha egy pont fokszáma 0 , akkor az a pont nincs más ponttal összekötve, ha pedig egy pont fokszáma $n - 1$, akkor ez a pont minden más ponttal össze van kötve. Ezért nem fordulhat elő, a gráfban 0 -fokú és $(n - 1)$ -fokú pont is legyen. A kettő közül legfeljebb egyik lehet. Így azt kapjuk, hogy mivel n pontunk van, de a fokszámra csak $n - 1$ különböző eset lehetséges,



A.9. ábra. Példák izomorf gráfokra



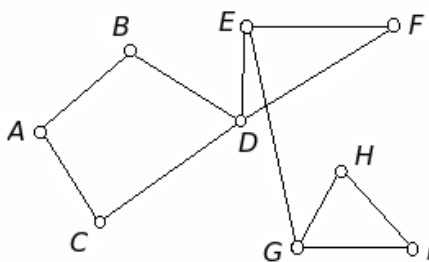
A.10. ábra. Példa teljes gráfokra

ezért legalább egy fokszámnak legalább kétszer kell előfordulnia. Ha egy gráfnak van legalább két pontja, akkor van két azonos fokú pontja.

Hány éle van az n -pontú egyszerű gráfnak, amelynek bármely két különböző pontját él köti össze? Az ilyen gráf neve teljes n -gráf (A.10. ábra). A teljes n -gráf minden pontjának foka $n - 1$. A teljes n -gráf éleinek száma $n(n - 1)/2$. Egy gráf és komplementere teljes gráfot alkotnak.

Egyszerű gráfok

Az általános gráfban két pont többszörösen is össze lehet kötve, illetve a hurokél is megengedett. A továbbiakban csak olyan gráfokról lesz szó, melyekben nincs sem többszörös él, sem hurok. Az ilyen gráfokat egyszerű gráfoknak nevezzük. Az egyszerű gráfok szomszédsági mátrixában csupa 0 és 1 áll, és a főátló minden eleme 0.



A.11. ábra. Az ábrán lévő gráf esetében $ABDEGH$ út, de az $ABDEFDEGIH$ nem. AB, BD, DE, EF, FD, DC vonal, de nem út. Az AB, BD, DC, CA élsorozat kör, vagyis a kiindulási pontba visszatérő vonal.

Út, vonal, kör, séta

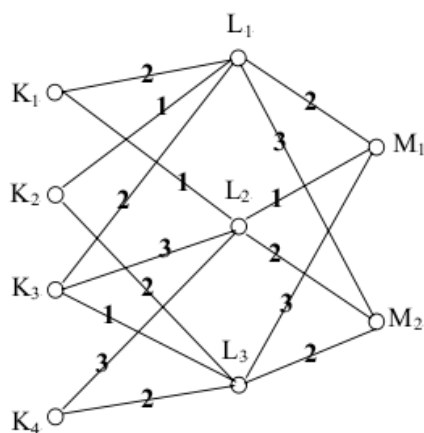
Ábrázolja a G gráf egy városrész úthálózatát. A gráfelméletben útnak nevezzük az éleknek olyan egymáshoz csatlakozó sorozatát, mely egyetlen szögpontra sem halad át egynél többször (A.11. ábra). Ez természetesen azt is jelenti, hogy minden élen legfeljebb egyszer haladunk át. Konkrét feladatokban legtöbbször utakat keresünk, mert azok száma véges, és általában rövid úton szeretnénk egyik pontból a másikba jutni.

Vonalnak nevezzük a gráf éleinek olyan egymáshoz csatlakozó sorozatát, melyben minden él különböző, ám egyes pontokon többször is áthaladhatunk. Például AB, BD, DE, EF, FD, DC vonal, de nem út (A.11. ábra). Vonalról van szó, ha egy részgráfot a toll felemelése nélkül, minden élt csak egyszer érintve meg tudunk rajzolni.

Még általánosabb fogalom a séta. Sétának nevezzük gráfban az AB, BD, \dots egymáshoz csatlakozó élek sorozatát (A.11. ábra), ahol az élek és pontok nem feltétlenül különböznek. Ha két pont között van séta, akkor biztos, hogy út is van. Az AB, BD, DC, CA élsorozat a kiindulási pontba visszatérő vonal, melyben minden él és pont, a kezdő és végponttól eltekintve, egyszer szerepel. Az ilyen élsorozatot körnek nevezzük. Ha egyik pontból a másikba két út is vezet, akkor a gráfban biztosan van kör.

Legyen A egy gráf szomszédsági mátrixa, akkor a $K(k) = A^k$, azaz az A mátrix k -adik hatványaként kapott mátrix $K(k)_{ij}$ eleme, a gráf i pontjából a j pontjába vezető k hosszúságú séták számát adja. Itt a séta hosszán az i ponttól a j csúcspontig történő haladás közben bejárt élek számát kell érteni. Valóban, maga a szomszédsági mátrix is ezt mutatja: mely szögpontra melyik szögpontra lehet egyetlen élen haladva eljutni. Nyilvánvalóan abba a szögpontra, amelyikkel közvetlenül össze van kötve éllel.

Nehezebb átlátni a mátrix önmagával való többszöri beszorzása eredményeként kapott mátrix elemeinek értelmét. A mátrixok szorzása megértésének egy-



A.12. ábra. Vasúti példa [3]

A.1. táblázat.

	L_1	L_2	L_3
K_1	2	1	0
K_2	1	0	2
K_3	2	3	1
K_4	0	3	2

$$\mathbf{A} = \begin{pmatrix} 2 & 1 & 0 \\ 1 & 0 & 2 \\ 2 & 3 & 2 \\ 0 & 3 & 2 \end{pmatrix} \quad (\text{A.3})$$

szerűsítése céljából lássunk egy könnyen érthető példát, s az általánosítás érdekében tekintsünk el a mátrixok négyzetes voltától. Legyen K, L, M három ország, melyek egyes városait vasút köti össze. Legyenek K országban K_1, K_2, K_3, K_4 ; L országban L_1, L_2, L_3 és M országban M_1 és M_2 városok. Az egyes városok között közlekedő vonatok számát írjuk rá a városokat jelző pontokat összekötő szakaszokra. Mátrixok segítségével is ábrázolhatjuk az utazási lehetőségeket. Lássuk először a K -ból L -be közlekedő vonatokat (ha K_i és L_j között nem közlekedik vonat, akkor 0-t írunk a megfelelő helyre).

Ábrázoljuk a K -ból L -be közlekedő vonatokat az A.1. táblázatban, és \mathbf{A} -val jelölve mátrixként (A.3 mátrix). A mátrix i -edik sorának j -edik eleme azt mutatja, K_i -ből hány vonat megy L_j -be.

Tekintsük át az L -ből M -be menő vonatokat a A.2. táblázattal. A megfelelő mátrixot jelöljük \mathbf{B} -vel (A.4 mátrix). Ebben a mátrixban az i -edik sor j -edik eleme

A.2. táblázat.

	M_1	M_2
L_1	2	3
L_2	1	2
L_3	3	2

$$\mathbf{B} = \begin{pmatrix} 2 & 3 \\ 1 & 2 \\ 3 & 2 \end{pmatrix} \quad (\text{A.4})$$

A.3. táblázat.

	M_1	M_2
K_1	c_{11}	c_{12}
K_2	c_{21}	c_{22}
K_3	c_{31}	c_{32}
K_4	c_{41}	c_{42}

azt mutatja, hogy L_i városból hány vonat megy M_j városba.

Az A.12. ábra szerint K országból M országba csak valamelyik L -beli városon át juthatunk el. A kérdés az, hogy hány különböző módon juthatunk el K_1 -ből M_1 -be. Az A.12. ábra alapján K_1 -ből L_1 -be menni az kétféleképpen lehetséges. Innen ugyancsak kétféleképpen utazhatunk M_1 -be. Ez tehát 2×2 lehetőség. K_1 -ből L_2 -be egy vonatjártat van, L_2 -ből M_1 -be ugyancsak egy. Így ezen az úton csak 1×1 módon juthatunk M_1 -be. K_1 -ből nem vezet vonat L_3 -ba, így K_1 -ből L_3 -an át M_1 -be jutni 0 lehetőség. (Írhatnánk 0×3 -at is, mert L_3 -ból három lehetőség van M_1 -be jutni.) Összesen tehát $2 \times 2 + 1 \times 1 + 0 \times 3 = 5$ különböző lehetőségünk van arra, hogy K_1 -ből M_1 -be utazzunk.

A feladatot általánosabban megfogalmazva, határozzuk meg az A.3. táblázat elemeit, ahol c_{ij} az a szám, amely azt mutatja, hogy K_i -ből hány különböző módon juthatunk el M_j -be. Tudjuk, hogy $c_{11} = 5$. Határozzuk meg például a c_{22} -t!

Írjuk egymás mellé az \mathbf{A} , \mathbf{B} és a \mathbf{C} mátrixokat.

$$\mathbf{A} = \begin{pmatrix} 2 & 1 & 0 \\ 1 & 0 & 2 \\ 2 & 3 & 2 \\ 0 & 3 & 2 \end{pmatrix} \quad \mathbf{B} = \begin{pmatrix} 2 & 3 \\ 1 & 2 \\ 3 & 2 \end{pmatrix} \quad \mathbf{C} = \begin{pmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \\ c_{31} & c_{32} \\ c_{41} & c_{42} \end{pmatrix} \quad (\text{A.5})$$

Már ismerjük a c_{11} meghatározásának módját: $c_{11} = 2 \cdot 2 + 1 \cdot 1 + 0 \cdot 3 = 5 = a_{11} \cdot b_{11} + a_{12} \cdot b_{21} + a_{13} \cdot b_{31}$, és a feladat alapján hasonló módon számítjuk ki a c_{22} -t: $c_{22} = 1 \cdot 3 + 0 \cdot 2 + 2 \cdot 2 = 7$, vagyis $c_{22} = a_{21} \cdot b_{12} + a_{22} \cdot b_{22} + a_{23} \cdot b_{32}$. Vegyük észre,

hogy c_{11} -et az \mathbf{A} mátrix első sora elemeinek a \mathbf{B} mátrix első oszlopának megfelelő elemeivel való szorzása, és a kapott három kéttényezős szorzatot összeadása által kapjuk meg. Ugyanezt láthatjuk a c_{22} esetén is.

Két mátrixból így módon előállított új mátrixot az előbbiek szorzatának nevezük. Itt nem részletezve a mátrixok szorzásának szabályait (mivel a függelék későbbi részeiben erre kitérünk), definiáljuk két mátrix szorzatát:

Az \mathbf{A} $m \times n$ típusú és a \mathbf{B} $n \times p$ típusú mátrix \mathbf{AB} szorzatán azt a \mathbf{C} $m \times p$ típusú mátrixot értjük, melyben

$$c_{ij} = a_{i1}b_{1j} + a_{i2}b_{2j} + \dots + a_{in}b_{nj},$$

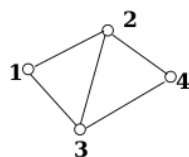
ahol $i = 1, 2, \dots, m$; $j = 1, 2, \dots, p$.

Mint látjuk, a szorzás definíciója nem bármely két mátrixra vonatkozik, hanem csak olyanokra, ahol az első tényezőnek annyi oszlopa van, ahány sora a másodiknak. A példában szereplő \mathbf{A} és \mathbf{B} mátrixok szorzata \mathbf{C} mátrix (A.6 formula).

$$\mathbf{A} = \begin{pmatrix} 2 & 1 & 0 \\ 1 & 0 & 2 \\ 2 & 3 & 2 \\ 0 & 3 & 2 \end{pmatrix} \quad \mathbf{B} = \begin{pmatrix} 2 & 3 \\ 1 & 2 \\ 3 & 2 \end{pmatrix} \quad \mathbf{C} = \begin{pmatrix} 5 & 8 \\ 8 & 7 \\ 10 & 14 \\ 9 & 10 \end{pmatrix} \quad (\text{A.6})$$

A mátrix szorzás fontos tulajdonsága, hogy nem kommutatív, vagyis $\mathbf{AB} \neq \mathbf{BA}$, ugyanakkor asszociatív, tehát $(\mathbf{AB})\mathbf{C} = \mathbf{A}(\mathbf{BC})$. A gráfokat leíró szomszédsági mátrixok hatványozása szempontjából ez fontos megállapítás. Könnyen belátható, hogy a négyzetes szomszédsági mátrixok hatványozása önmagával történő többszöri beszorzással egyszerűen megoldható.

Vegyünk egy négyponos egyszerű gráfot, melynek szomszédsági mátrixa \mathbf{A} (A.13. ábra). Jelöljük a csúcspontjait rendre arab számokkal, így ezek azonosíthatók a gráf mátrixának sor- és oszlopszámaival.



A.13. ábra. A négyponos \mathbf{A} gráf és szomszédsági mátrixa (A.7)

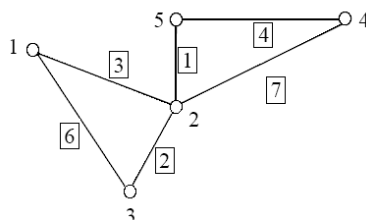
$$\mathbf{A} = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix} \quad (\text{A.7})$$

Számoljuk ki a $K(2) = \mathbf{A}^2$ és a $K(3) = \mathbf{A}^3$ mátrixokat (A.8. formula).

$$K(2) = \mathbf{A}^2 = \begin{pmatrix} 2 & 1 & 1 & 2 \\ 1 & 3 & 2 & 1 \\ 1 & 2 & 3 & 1 \\ 2 & 1 & 1 & 2 \end{pmatrix}; \quad K(3) = \mathbf{A}^3 = \begin{pmatrix} 2 & 5 & 5 & 2 \\ 5 & 4 & 5 & 5 \\ 5 & 5 & 4 & 5 \\ 2 & 5 & 5 & 2 \end{pmatrix} \quad (\text{A.8})$$

Ezen mátrixok elemei a sor és oszlopszámoknak megfelelő vertexek között való közlekedés közben két, illetve három élen történő áthaladások számát adják. A kapott mátrixok ugyanúgy, mint az irányítatlan gráfok szomszédsági mátrixai, szimmetrikusak a főátlóra. Vegyük észre, hogy itt nem a bejárható utak számát, hanem a séták számát kaptuk meg. Amennyiben az i -edik pontból a j -edikbe vezető $k = 1, 2, \dots, n - 1$ lépéses utak számát szeretnénk megtudni, rögzíteni kell a séta közben érintett pontokat, s csak azokat tartani meg, melyekben minden érintett vertex csak egyszer fordul elő.

Nézzünk egy térinformatikához közeli példát. Cél a legrövidebb utak meghatározása bizonyos csomópontok között. Ábrázoljuk a csomópontokat és az őket összekötő utakat súlyozott gráffal, ahol az élekhez számok vannak rendelve. A konkrét feladattól függően a gráf lehet irányítatlan vagy akár irányított. Többféle algoritmus létezik a probléma megoldására (Warshall algoritmus, Dijkstra algoritmus, stb.). Példaként ismerkedjünk meg a Warshall algoritmussal. Legyen adott egy egyszerű öt vertexű súlyozott gráf (A.14. ábra). Szomszédsági mátrixát jelöljük \mathbf{A} -val. A ∞ jelek az él nélküli pontpárokat jelölik.



A.14. ábra. G egy egyszerű, öt vertexű gráf, súlyozott élekkel, melynek szomszédsági mátrixa \mathbf{A} (A.9 formula)

$$\mathbf{A} = \begin{pmatrix} 0 & 3 & 6 & \infty & \infty \\ 3 & 0 & 2 & 7 & 1 \\ 6 & 2 & 0 & \infty & \infty \\ \infty & 7 & \infty & 0 & 4 \\ \infty & 1 & \infty & 4 & 0 \end{pmatrix} \quad (\text{A.9})$$

Vezessünk be egy új mátrix műveletet:

$$A[k]_{ij} = \min \{A[k-1]_{ij}, A[k-1]_{ik} + A[k-1]_{kj}\},$$

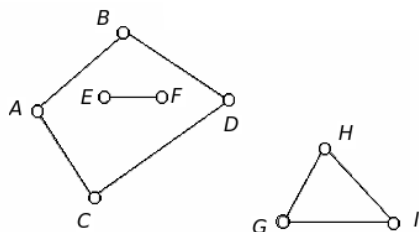
ahol a $\mathbf{A}[k]$ a mátrixok sorozata $k = 1, \dots, n$ -ig számítandó, ahol n az \mathbf{A} mátrix mérete ($n \times n$), és $\mathbf{A}[0] = \mathbf{A}$. Vagyis a súlymátrixból kiindulva (amikor $k = 0$) rendre előállítjuk az $\mathbf{A}[1], \mathbf{A}[2], \dots, \mathbf{A}[n]$ mátrixokat a fenti képlet szerint, vagyis az új – magasabb sorszámú mátrix – ij indexű elem értékéül az előző mátrix ij -edik eleme, valamint az ik és kj indexű elemek összegének összehasonlítása után a kisebbiket vesszük. Az $\mathbf{A}[n]$ -re kapott eredmény mátrix a legrövidebb utakat adja meg. Esetünkben $n = 5$, vagyis a súlymátrix alapján az ötödik menetben $k = 5$ -re a (A.10. számú) mátrixot kapjuk.

$$\mathbf{A} = \begin{pmatrix} 0 & 3 & 5 & 8 & 4 \\ 3 & 0 & 2 & 5 & 1 \\ 5 & 2 & 0 & 7 & 3 \\ 8 & 5 & 7 & 0 & 4 \\ 4 & 1 & 3 & 4 & 0 \end{pmatrix} \quad (\text{A.10})$$

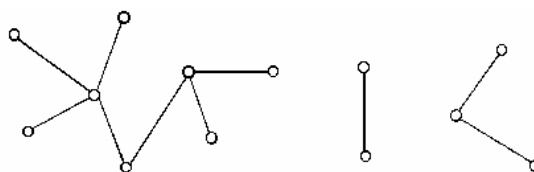
Ez tehát a keresett útmátrix. A harmadik sor negyedik oszlopának eleme $a[5]_{34} = 7$ például azt jelenti, hogy a gráf 3-as vertexéből a 4-esbe vezető legrövidebb út hossza 7 egységnyi (3-2-5-4 pontokon át vezet).

Összefüggő gráfok

Egy gráfot összefüggőnek mondunk, ha bármely pontjából bármely másik pontjába eljuthatunk úton (elég azt megállapítani, hogy egy tetszőlegesen kiválasztott pontjából az összes többihez van-e út). Az eddig vizsgált gráfok mind összefüggők voltak, míg az A.15. ábrán lévő gráf nem összefüggő. Ha egy gráf nem összefüggő, akkor van olyan pontja, melyből nem vezet út minden másik pontba. A gráf egy darabja az összes olyan pontok (és az őket összekötő élek) halmaza, melyek kölcsönösen elérhetők úton. Ezt a gráf egy összefüggő komponensének nevezzük.



A.15. ábra. Nem összefüggő gráfok



A.16. ábra. Fák és erdők

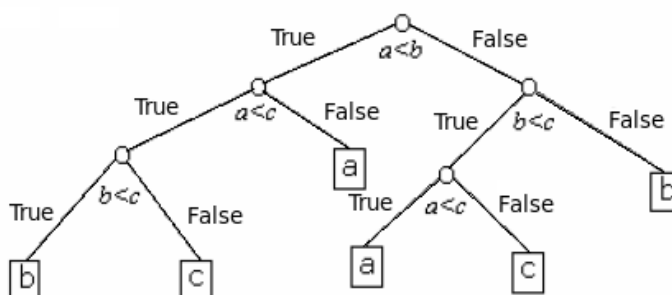
Fák, erdők

Az összefüggő, körmentes egyszerű gráfokat fának nevezzük. Az olyan nem összefüggő gráfot, melynek összefüggő komponensei fák, erdőnek nevezzük (A.16. ábra).

- Fában két pont között pontosan egy út létezik. Létezik, mert a gráf összefüggő, és pontosan egy, mert ha kettő lenne, kör is lenne.
- Ha fához hozzáteszünk egy élt, már nem lehet fa. Az (ij) él hozzávételével ugyanis kört kapnánk, hiszen eddig is volt a két csúcstől út, és most lett még egy.
- Fából egy élt elhagyva már nem lehet fa. Ha ugyanis fa lenne, akkor abba az iménti élt visszatéve megint csak fát kapnánk (az eredetit), ami az előző bekezdés szerint nem lehet.
- Úgy is fogalmazhatnánk, hogy adott számú vertexű fa a lehető legkevesebb élt tartalmazza, ami szükséges az összefüggőséghez.

Az A.16. ábrán látható fák mindegyikében van elsőfokú pont. Vajon minden esetben így van ez? Válasszuk a fa egy tetszőleges pontját, nevezzük A -nak (ez lehet elsőfokú vagy akármilyen más), és induljunk el valamely irányba, a ponthoz illeszkedő valamely élen (legalább egy ilyen él van, mivel a gráf összefüggő). Mivel nincs kör, így soha nem térhetünk vissza olyan pontba, ahol egyszer már jártunk, ezért a gráf végessége miatt az útnak egyszer vége lesz. Nem tudunk tovább menni, azaz ez a pont elsőfokú. Legyen ez a B pont. Most ebből a B elsőfokú pontból indulva, ugyanilyen módon, ismét lesz vége a körmentes gráfnak, kapunk egy újabb elsőfokú pontot, amely lehet az A (ha elsőfokú volt), vagy valamely más, például a C . Ebből következik, hogy egynél többpontú fában van legalább két elsőfokú pont.

Meg tudjuk-e mondani, hogy egy n -pontú fának, hány éle van? Az A.16. ábrán látható esetekben a 8-pontúnak 7 éle van; a 2-pontúnak 1 éle van; a 3-pontúnak 2 éle van. Általában az n -pontú fának $n - 1$ éle van. $n = 1$ -re igaz az állítás, az 1-pontú fának 0 éle van. Tegyük fel, hogy igaz k pontú fára, és igazoljuk, hogy akkor



A.17. ábra. Bináris fa

igaz $k + 1$ -pontúra is. Növeljük a k -pontú fát egy ággal. Mivel a fa körmentes, összefüggő, egyszerű gráf, így a fa bármely választott pontjából csak egy új, eddig nem létező pontba húzhatunk egy élt. Eggyel nőtt a fa pontjainak száma, és eggyel nőtt az élek száma is, így az állítás igaz.

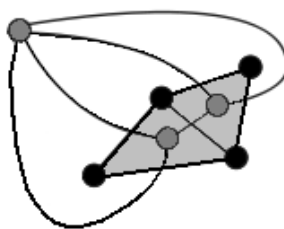
A fák alkalmasak arra, hogy számrendszereket szemléltessenek. Különösen nagy szerepük van az úgynevezett bináris fáknak, melyek a kettes számrendszert szemléltetik. Itt egy vertex egy kérdést jelent, melyre igaz (*i*) vagy hamis (*h*) a válasz, és e szerint megyünk tovább a bal vagy a jobb oldali élen. Példaként ábrázoljuk bináris fával azt az algoritmust, mely három szám közül kiválasztja a középsőt (A.17. ábra). Itt a kérdés a fa csúcspontja mellett feltüntetett számpárok viszonyának igaz vagy hamis volta. A fa valamely téglalappal jelzett elsőfokú pontja tartalmazza a választ (a konkrét számadatoktól függően).

Síkbarajzolható gráf

Egy gráfot akkor nevezünk síkbarajzolhatónak, ha lerajzolható úgy a síkban, hogy az élei nem metszik egymást. Az Euler szerint egy összefüggő, síkba rajzolt gráf csúcsainak és tartományainak száma (beleértve a külső, nem korlátos tartományt is) 2-vel nagyobb az éleinek számánál.

Duális gráf

Egy síkba rajzolható gráf duális gráfja alatt azt a gráfot értjük, aminek csúcsai az eredeti gráf tartományai, és azok a csúcsok vannak összekötve, amik megfelelői szomszédosak voltak (A.18. ábra). Síkbarajzolható gráf duálisa is síkbarajzolható gráf.



A.18. ábra. Egy gráf és duálisa

A.2. Mátrixszámítás

Determinánsok

Tekintsük a következő elsőfokú lineáris egyenletrendszert:

$$\begin{aligned} a_{11}x_{11} + a_{12}x_{12} &= b_1 \\ a_{21}x_{21} + a_{22}x_{22} &= b_2 \end{aligned}$$

Nevezük másodrendű determinánssnak az $a_{11}, a_{22}, a_{21}, a_{12}$ elemekből alkotott kifejezést, amely sorokból és oszlopokból áll,

$$\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}$$

és amelynek értéke $a_{11}a_{22} - a_{12}a_{21}$. A fenti egyenletrendszer megoldása a következőképpen írható fel determinánsokkal:

$$x_1 = \frac{\begin{vmatrix} b_1 & a_{12} \\ b_2 & a_{22} \end{vmatrix}}{\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}}$$

$$x_2 = \frac{\begin{vmatrix} a_{11} & b_1 \\ a_{21} & b_2 \end{vmatrix}}{\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}}$$

Az n -edrendű determinánsnak nevezük az $n \times n$ elemből álló determinánst.

$$\begin{vmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & & & \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{vmatrix}$$

Főátlónak nevezzük az $a_{11}, a_{22}, \dots, a_{nn}$, és mellékátlónak a $a_{1n}, a_{2n}, \dots, a_{n1}$ elemeket. Értékét a következőképpen határozhatjuk meg:

$$\begin{vmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & & & \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{vmatrix} = a_{11}A_{11} + a_{12}A_{12} + \dots + a_{1n}A_{1n} = \sum_{i=1}^n a_{1i}A_{1i}$$

ahol A_{1i} az a_{1i} elemhez tartozó $n - 1$ -edrendű aldeterminánst jelenti, amely az első sor és az i -edik oszlop elhagyásával keletkezett determináns.

Mátrixok

Mátrixnak nevezzünk egy $n \times m$ számú, táblázatosan a_{ik} elemeket tartalmazó elrendezést, amely a következő alakot ölti:

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & & & \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{pmatrix}$$

A fenti mátrix $n \times m$ típusú, mivel n sorból és m oszlopból áll, továbbá a_{ik} a mátrix i -edik sorában és a j -edik oszlopban található eleme. A mátrix elemei lehetnek valós vagy komplex számok, vektorok, függvények, akár mátrixok is. Az áttekinthetőség kedvéért a mátrixokat a bonyolult táblázatos megjelenítés helyett gyakran nagybetűs, félkövér szimbólummal jelöljük. A fenti mátrixot tehát jelölhetjük \mathbf{A} -val, vagy $\mathbf{A} = [a_{ik}]$ -val, vagy ha kiterjedését is láttatni szeretnénk, akkor $\mathbf{A} = [a_{ik}]_{n,m}$ -val.

Ha egy mátrix sor- és oszlopszáma egyenlő, akkor az egy négyzetes mátrix. A négyzetes mátrix sorainak (oszlopainak) száma a mátrix rendje.

Transzponált mátrixot kapunk, ha a sorokat és az oszlopokat felcseréljük. Jelölése \mathbf{A}^* . Ily módon tehát egy $n \times m$ -es mátrix transzponáltja egy $m \times n$ -es mátrix.

$$\mathbf{A}^* = \begin{pmatrix} a_{11} & a_{21} & \cdots & a_{n1} \\ a_{12} & a_{22} & \cdots & a_{n2} \\ \vdots & & & \\ a_{1m} & a_{2m} & \cdots & a_{nm} \end{pmatrix}$$

Vektorok leírására gyakran használunk speciális mátrixokat. Ha egy mátrix egyetlen oszlopból (sorból) áll, akkor arra azt mondjuk, hogy oszlopvektor (sorvektor). Jelölése

$$\mathbf{a} = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix}$$

Oszlop mátrix transzponáltja sormátrix, vagyis $\mathbf{a}^* = (a_1, a_2, \dots, a_n)$.

Zérusmátrix

A zérusmátrix minden elem 0.

$$\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

Diagonálmátrix

Diagonális mátrixnak nevezzük az olyan négyzetes mátrixot, amelynek főátlójában van csak nullától különböző elem.

$$\begin{pmatrix} a_{11} & 0 & \dots & 0 \\ 0 & a_{22} & \dots & 0 \\ \vdots & & & \\ 0 & 0 & \dots & a_{nn} \end{pmatrix}$$

A diagonális mátrixot a főátlójának elemeivel is szoktuk rövidítve jelölni: $\langle a_{11}, a_{22}, \dots, a_{nn} \rangle$

Egységmátrix

Az egységmátrix egy olyan diagonális mátrix, amely főátlójának minden eleme 1.

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Egységvektorok

Az olyan sor vagy oszlopvektorokat, amelynek egyik eleme 1, a többi 0, egységvektoroknak nevezzük.

$$\mathbf{e}_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \quad \mathbf{e}_2 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \quad \mathbf{e}_3 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

vagy sorvektor alakban $\mathbf{e}_1 = [1, 0, 0]$, $\mathbf{e}_2 = [0, 1, 0]$, $\mathbf{e}_3 = [0, 0, 1]$

Szimmetrikus mátrix

Az a négyzetes mátrix, amelynek elemei a főátlóra szimmetrikusak, szimmetrikus mátrix, vagyis elemeire igaz, hogy $a_{ij} = a_{ji}$.

$$\begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}$$

Antiszimmetrikus mátrix

Az a négyzetes mátrix, amelynek elemei a főátlóra antiszimmetrikusak, vagyis a főátlóra szimmetrikus elemei egymás ellentettjei, antiszimmetrikus mátrix, vagyis elemeire igaz, hogy $a_{ij} = -a_{ji}$.

$$\begin{pmatrix} 0 & 2 & 1 \\ -2 & 0 & 0 \\ -1 & 0 & 0 \end{pmatrix}$$

Mátrixműveletek**Mátrixok egyenlősége**

A és **B** mátrix akkor, és csak akkor egyenlő, ha azonos típusúak, vagyis ha ugyanannyi sort és ugyanannyi oszlopot tartalmaznak, továbbá a megfelelő elemeik azonosak ($a_{ik} = b_{ik}$ minden ik -ra).

Összeadás

Az összeadás (kivonás) csak azonos típusú mátrixokra értelmezett, vagyis ha **A** és **B** $n \times m$ -es mátrixok, $a_{ij} \in \mathbf{A}$ és $b_{ij} \in \mathbf{B}$ akkor

$$\mathbf{A} + \mathbf{B} = \begin{pmatrix} a_{11} + b_{11} & a_{12} + b_{12} & \dots & a_{1m} + b_{1m} \\ a_{21} + b_{21} & a_{22} + b_{22} & \dots & a_{2m} + b_{2m} \\ \vdots & & & \\ a_{n1} + b_{n1} & a_{n2} + b_{n2} & \dots & a_{nm} + b_{nm} \end{pmatrix}$$

Az összeadás (kivonás) kommutatív, azaz $\mathbf{A} + \mathbf{B} = \mathbf{B} + \mathbf{A}$, továbbá kommutatív, azaz $(\mathbf{A} + \mathbf{B}) + \mathbf{C} = \mathbf{A} + (\mathbf{B} + \mathbf{C})$.

Tranzponáltakra fennáll a következő összefüggés: $(\mathbf{A} + \mathbf{B})^* = \mathbf{A}^* + \mathbf{B}^*$

Mátrix szorzása skalárral

Mátrixot úgy szorzunk skalárral, hogy minden elemét megszorozzuk a skalárral

$$k\mathbf{A} = \begin{pmatrix} ka_{11} & ka_{12} & \dots & ka_{1m} \\ ka_{21} & ka_{22} & \dots & ka_{2m} \\ \vdots & & & \\ ka_{n1} & ka_{n2} & \dots & ka_{nm} \end{pmatrix}$$

A skalárral való szorzás kommutatív, asszociatív és disztributív, azaz $k\mathbf{A} = \mathbf{A}k$, $(k_1k_2)\mathbf{A} = k_1(k_2\mathbf{A})$, $k(\mathbf{A} + \mathbf{B}) = k\mathbf{A} + k\mathbf{B}$, valamint $(k_1 + k_2)\mathbf{A} = k_1\mathbf{A} + k_2\mathbf{A}$

Mátrix szorzása mátrixszal

Az \mathbf{A} és \mathbf{B} mátrix csak abban az esetben szorozható össze, azaz az \mathbf{AB} szorzat akkor értelmezett, ha \mathbf{A} -nak ugyanannyi oszlopa van, mint ahány sora \mathbf{B} -nek. Legyen tehát az \mathbf{A} mátrix egy $n \times m$ -es, míg \mathbf{B} egy $m \times p$ -s mátrix. Ekkor az \mathbf{AB} szorzat egy olyan $n \times p$ típusú \mathbf{C} mátrix lesz, amelynek elemei $c_{ik} = \sum_{j=1}^m a_{ij}b_{jk}$ módon számíthatók ki. Az érthetőség kedvéért fejtsük ki részletesebben a műveletet. Legyen tehát

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1m} \\ a_{21} & a_{22} & \dots & a_{2m} \\ \vdots & & & \\ a_{n1} & a_{n2} & \dots & a_{nm} \end{pmatrix} \quad \text{és} \quad \mathbf{B} = \begin{pmatrix} b_{11} & b_{12} & \dots & b_{1p} \\ b_{21} & b_{22} & \dots & b_{2p} \\ \vdots & & & \\ b_{m1} & b_{m2} & \dots & b_{mp} \end{pmatrix}$$

Vegyük a \mathbf{B} mátrix első oszlopát és fektessük rá az \mathbf{A} mátrix első sorára. Szorozzuk össze az azonos indexű elemeket, majd adjuk össze. Az összeg a szorzat mátrix első sorának első eleme lesz. Ezután vegyük a \mathbf{B} mátrix második oszlopát, és fektessük az \mathbf{A} mátrix első sorára. Szorozzuk össze az megfelelő elemeket, és a szorzatokat adjuk össze, amely a szorzat mátrix első sorának második eleme lesz, és így tovább.

Amint látható a mátrixszorzás nem kommutatív, azaz $\mathbf{AB} \neq \mathbf{BA}$, ellenben disztributív, azaz $\mathbf{A}(\mathbf{B} + \mathbf{C}) = \mathbf{AB} + \mathbf{AC}$. Mivel a mátrixszorzás nem kommutatív, ezért beszélhetünk jobboldali és baloldali szorzatról. Kivételesen, diagonális mátrixok esetében a szorzás kommutatív.

Az egységmátrixszal való szorzás (jelöljük \mathbf{AE} -vel), akár balról, akár jobbról szoroztunk, a mátrixot érintetlenül hagyja, azaz $\mathbf{EA} = \mathbf{AE} = \mathbf{A}$. Zérusmátrixszal való szorzás zérusmátrixot eredményez, azaz $\mathbf{0A} = \mathbf{A0} = \mathbf{0}$.

Skaláris szorzat, diádikus szorzat

Legyen \mathbf{a} és \mathbf{b} n elemű oszlopvektorok. Végezzük el az $\mathbf{a}^*\mathbf{b}$ szorzást (\mathbf{a}^* az \mathbf{a} transzponáltja):

$$\mathbf{a}^*\mathbf{b} = \mathbf{b}^*\mathbf{a} = (a_1, a_2, \dots, a_n) \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix} = \sum_{i=1}^n a_i b_i$$

A szorzás eredménye egy szám, a neve skaláris szorzat.

Végezzük el a fenti oszlopvektorokra az \mathbf{ab}^* szorzást:

$$\mathbf{ab}^* = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix} (b_1, b_2, \dots, b_n) = \begin{pmatrix} a_1 b_1 & a_1 b_2 & \dots & a_1 b_n \\ a_2 b_1 & a_2 b_2 & \dots & a_2 b_n \\ \vdots & \vdots & \dots & \vdots \\ a_n b_1 & a_n b_2 & \dots & a_n b_n \end{pmatrix}$$

Belátható, hogy a \mathbf{ba}^* szorzás is ugyanerre az eredményre vezet. Az \mathbf{ab}^* és \mathbf{ba}^* szorzatok neve diádikus szorzat.

A mátrix nyoma

Az \mathbf{A} mátrix főátlójában lévő elemek összegét a mátrix nyomának nevezzük, $\mathbf{Sp}(\mathbf{A})$ -val jelöljük, és a következőképpen számítjuk ki:

$$\mathbf{Sp}(\mathbf{A}) = \sum_{i=1}^n a_{ii}$$

Négyzetes mátrix hatványa

Az \mathbf{A} mátrix n -edik hatványának nevezzük a következő n tényezős mátrix szorzatot:

$$\mathbf{A} \cdot \mathbf{A} \cdots \mathbf{A} = \mathbf{A}^n,$$

ahol n természetes szám. Megállapodás szerint $\mathbf{A}^0 = \mathbf{E}$. Ezekből következően egységmátrix bármely hatványa egységmátrix, valamint zérusmátrix bármely hatványa zérusmátrix. Diagonális mátrix n -edik hatványa is diagonális lesz, de a főátlóban a megfelelő elemek n -edik hatványai lesznek, vagyis

$$\langle a_1, a_2, \dots, a_m \rangle^n = \langle a_1^n, a_2^n, \dots, a_m^n \rangle$$

Négyzetes mátrix determinánsa

Legyen \mathbf{A} egy $n \times n$ -es négyzetes mátrix:

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \dots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix}$$

Az \mathbf{A} mátrix determinánsának az elemeiből alkotott determinánst nevezzük, és $\det \mathbf{A}$ -val vagy $|\mathbf{A}|$ -val jelöljük:

$$|\mathbf{A}| = \begin{vmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & & & \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{vmatrix}$$

Ha $|\mathbf{A}| \neq 0$, akkor az \mathbf{A} mátrix reguláris, ha $|\mathbf{A}| = 0$, akkor szinguláris.

Négyzetes mátrix adjungáltja

Az \mathbf{A} négyzetes mátrix adjungáltja

$$\text{adj}\mathbf{A} = \text{adj} \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & & & \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix} = \begin{pmatrix} A_{11} & A_{21} & \dots & A_{n1} \\ A_{12} & A_{22} & \dots & A_{n2} \\ \vdots & & & \\ A_{1n} & A_{2n} & \dots & A_{nn} \end{pmatrix}$$

ahol A_{ij} az \mathbf{A} mátrix determinánása a_{ij} -edik eleméhez tartozó aldetermináns. Vegyük észre, hogy a sor- és oszlopindexek felcserélődtek, vagyis $\text{adj}\mathbf{A}$ az

$$\begin{pmatrix} A_{11} & A_{12} & \dots & A_{1n} \\ A_{21} & A_{22} & \dots & A_{2n} \\ \vdots & & & \\ A_{n1} & A_{n2} & \dots & A_{nn} \end{pmatrix}$$

mátrix transzponáltja.

Mátrix inverze

Legyen \mathbf{A} egy olyan mátrix, melynek determinánása nem nulla ($|\mathbf{A}| \neq 0$). Keresünk az \mathbf{A}^{-1} mátrixot, melyre igaz, hogy $\mathbf{A}\mathbf{A}^{-1} = \mathbf{E}$ és $\mathbf{A}^{-1}\mathbf{A} = \mathbf{E}$. \mathbf{A}^{-1} mátrixot az \mathbf{A} mátrix inverzének nevezzük, és a következőképpen számítjuk ki:

$$\mathbf{A}^{-1} = \frac{1}{|\mathbf{A}|} \text{adj}\mathbf{A}$$

Mátrix sajátértékei és sajátvektorai

Keressük azokat az összetartozó skalárokat és vektorokat, amelyek kielégítik a következő mátrixegyenletet:

$$\mathbf{A}\mathbf{x} = \lambda\mathbf{x}$$

ahol λ az \mathbf{A} mátrix sajátértékeit, \mathbf{x} a sajátvektorait jelenti. Amint látható, a vektor λ -val való szorzása ugyanazt eredményezi, mint a vektor \mathbf{A} -val való szorzása. Alakítsuk át az egyenletet a következő módon:

$$(\lambda \mathbf{E} - \mathbf{A})\mathbf{x} = \mathbf{0}$$

Írjuk át a fenti kifejezést homogén lineáris egyenletrendszer alakba:

$$\begin{aligned} (\lambda - a_{11})x_1 - a_{12}x_2 - \cdots - a_{1n}x_n &= 0 \\ a_{21}x_1 - (\lambda - a_{22})x_2 - \cdots - a_{2n}x_n &= 0 \\ \vdots & \\ a_{n1}x_1 - a_{n2}x_2 - \cdots - (\lambda - a_{nn})x_n &= 0 \end{aligned}$$

Ennek az egyenletrendszernek akkor van megoldása, ha a determinánsa nulla, vagyis

$$\begin{vmatrix} \lambda - a_{11} & -a_{12} & \cdots & -a_{1n} \\ -a_{21} & \lambda - a_{22} & \cdots & -a_{2n} \\ \vdots & & & \\ -a_{n1} & -a_{n2} & \cdots & \lambda - a_{nn} \end{vmatrix} = 0$$

A.3. Vektorterek

Kérdezzünk meg egy mérnököt, hogy mi a vektor. A válasz valami olyasmi lesz, hogy „irányított mennyiség”. Ez a megközelítés a vektor fogalom geometriai szemléletét tükrözi. Van azonban a vektoroknak más megközelítése is, amely szakít az „irányított mennyiség” szemlélettel. A vektorfogalom megközelíthető úgy is, hogy vektor az a halmaz, amely kielégít bizonyos algebrai kritériumokat, és egyben kielégíti a geometriai vektor fogalomból eredő kívánalmakat is. A következőkben a vektor fogalom ezen általánosabb megközelítését fogjuk részletezni.

Vektorok definíciója

Nevezzük vektortérnek a $V = \{v_i\}$ halmazt és egy skalár mezőt $A = \{a_i\}$, amely következő műveletekkel és tulajdonságokkal rendelkezik:

1. Két vektor összege egy harmadik vektor
2. Egy vektor skalárral szorozva egy másik vektort eredményez

Tekintsük át a vektorok összeadásának és a skalárral való szorzásának tulajdonságait. Legyen $v_i \in V$ és $a_i \in A$.

1. $v_1 + v_2 = v_2 + v_1$
2. $(v_1 + v_2) + v_3 = v_1 + (v_2 + v_3)$
3. $a_1(v_1 + v_2) = a_1v_1 + a_2v_2$
4. $(a_1 + a_2)v_i = a_1v_i + a_2v_i$

5. $(a_1 a_2) v_i = a_1 (a_2 v_i)$
6. $1 \cdot v_i = v_i$
7. Egyetlen olyan vektor létezik, amelyre igaz, hogy $0 \cdot v_i = v_0$. Ez a null vektor (jele v_0).

A skalár lehet valós szám, de lehet komplex szám is. A fenti definíció definiálja a vektortér fogalmát. A vektortér tehát egy halmaz két művelettel, vagyis egyrészt $V \times V \rightarrow V$, amely művelet a vektorok összeadása, és a $A \times V \rightarrow V$, amely a skalárszorzást definiálja ($a \times a$ keresztszorzatot jelöli). Bármely halmaz, amely rendelkezik ezzel a két művelettel, és kielégíti a fenti hét tulajdonságot, vektortér.

Ez a vektorfogalom vajon alkalmazható valós vektorokra („irányított mennyiségekre”)? Így még nem. Nem látszik ugyanis, hogy ezen vektorok milyen irányúak, és mekkorák. Legyen v_1, v_2, \dots a hagyományos értelemben vett vektorok halmaza. Legyenek a_1, a_2, \dots skalárok (például valós számok). Úgy adjuk össze őket, ahogy 1. mutatja. A vektorok összeadása (2.) szerint asszociatív. Ha végig nézzük 1.-7.-ig a műveleteket, látható, hogy a hagyományos, geometriai jelentést hordozó vektorokra is igazak. Csakhogy ezek a műveletek nemcsak ezekre a vektorokra igazak, hanem például 3×2 -es mátrixokra is, amikre sosem mondjuk, hogy vektor.

Metrika

Az A.3-ben megadott definíció semmit nem mond a vektorok hosszáról. Vezessük be a távolság fogalmát következő módon: Legyen A metrikus tér (X, d) egy halmaza X -nek, amely rendelkezik a következő művelettel: $dX \times X \rightarrow R$. Minden a, b elemére X -nek $d(a, b) \geq 0$, amelyre

1. $d(a, b) = 0$ akkor és csak akkor, ha $a = b$
2. $d(a, b) = d(b, a)$ (szimmetrikus)
3. $d(a, b) \leq d(a, c) + d(c, b)$ (háromszög egyenlőtlenség)

A $d(a, b)$ számot az a és b vektorok távolságának nevezzük, a d -t pedig metrikának.

Ez alapján számos módon megadható a távolság fogalma. Vegyünk például két 3×3 -as oszlopvektort. Legyen ezek egyik lehetséges távolsága a következő:

$$d_2(x_1, x_2) = d_2 \left(\begin{bmatrix} a_1 \\ b_1 \\ c_1 \end{bmatrix}, \begin{bmatrix} a_2 \\ b_2 \\ c_2 \end{bmatrix} \right) = \sqrt{|a_1 - a_2|^2 + |b_1 - b_2|^2 + |c_1 - c_2|^2}$$

Az előbbihez hasonlóan megfelelő távolság definíció lehet a következő:

$$d_1(x_1, x_2) = |a_1 - a_2| + |b_1 - b_2| + |c_1 - c_2|$$

Norma

A metrikus tér X, d összetartozó párok, ahol X egy halmaz, és d a távolság képzési művelet (távolság fogalmat bármely halmazra alkalmazhatunk, nemcsak vektortérre).

Legyen $V = \{v_i\}$ vektortér, és a v vektor hossza $\|v\|$, amelyet nevezzünk normának. A leképezés $\|\cdot\| : X \rightarrow \mathcal{R}$ kielégíti a következőket:

1. $\|v\| = 0$ akkor és csak akkor, ha $v = v_0$ (null vektor)
2. $\|av\| = |a| \cdot \|v\|$ minden $v \in V$ -re és minden $a \in \mathcal{A}$
3. $\|v_1 + v_2\| \leq \|v_1\| + \|v_2\|$ minden $v_1, v_2 \in V$ -re

A norma fogalma tehát a vektor hosszát méri. Lássunk néhány norma definíciót.

$$\|v\|_\infty = \max_{a \leq t \leq b} |v(t)|$$

$$\|v\|_1 = \int_a^b |v(t)| dt$$

$$\|v\|_2 = \sqrt{\int_a^b |v(t)|^2 dt}$$

Belső (skaláris) szorzat

A következő vektortereket leíró fogalom a belső vagy skaláris szorzat. Jelölje a műveletet $\langle v_1 | v_2 \rangle$, amely v_1 és v_2 vektorpárokra vonatkozik. A művelet a következő négy tulajdonságnak kell eleget tegyen minden $a_i \in \mathcal{C}$ és minden $v_i \in V$ -re:

1. $\langle v_1 | v_2 \rangle = \langle v_2 | v_1 \rangle^*$ (* a szám konjugáltja)
2. $\langle v_1 + v_2 | v_3 \rangle = \langle v_1 | v_3 \rangle + \langle v_2 | v_3 \rangle$
3. $\langle av_1 | v_2 \rangle = a^* \langle v_1 | v_2 \rangle$ (a^* a szám konjugáltja)
4. $\langle av_1 | v_1 \rangle \geq 0$ és $\langle av_1 | v_1 \rangle = 0$ akkor és csak akkor, ha $v_1 = v_0$ (null vektor)

Ha a $V = \{v_i\}$ vektortér kielégíti ezeket a tulajdonságokat, akkor azt mondjuk, hogy a skalártér, ebből következően normált tér is, ami viszont maga után vonja, hogy metrikus tér is. Ebben az esetben a norma:

$$\|v_1\| = \sqrt{\langle v_1 | v_1 \rangle}$$

A vektorok közötti szöveget a következőképpen definiáljuk:

$$\cos \theta = \frac{\langle v_1 | v_2 \rangle}{\|v_1\| \cdot \|v_2\|}$$

ahol v_1, v_2 a $V = \mathcal{C}^n$ a komplex számsík feletti térben értelmezett két vektor.

Ortogonalitás

Két vektor ortogonális, ha a belső szorzatuk nulla, azaz $\langle v_1 | v_2 \rangle$ (geometriai vektorokra ez akkor áll elő, ha a bezárt szögük 90°).

Cauchy-Buniakovsky-Schwartz egyenlőtlenség. Ha V vektortér, vagyis van geometria, akkor igaz a következő egyenlőtlenség bármely két vektorra ($u, v \in V$)

$$|\langle u | v \rangle|^2 \leq \langle u | u \rangle \langle v | v \rangle$$

az egyenlőség akkor és csak akkor áll fenn, ha $u = kv$.

Lineáris függetlenség

Legyen adva egy v_1, v_2, \dots, v_n vektorokból álló V vektortér és egy A skalár mező. A vektorok lineárisan függetlenek, ha a

$$\sum_{i=1}^n a_i v_i = 0$$

összeg akkor és csak akkor nulla, ha valamennyi a_i skalár nulla. Akkor lineárisan összefüggők a vektorok, ha nem függetlenek.

Bázis

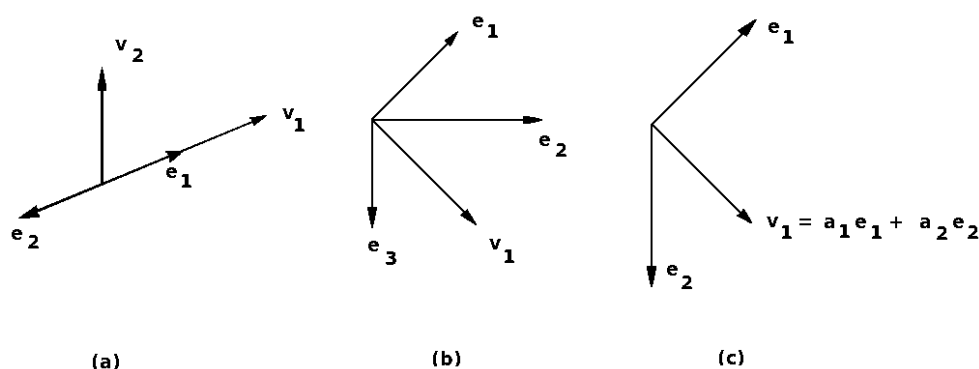
Bázisnak nevezzük az olyan vektorok halmazát, amelyekre igaz, hogy

1. minden vektor előállítható a bázisvektorok lineáris kombinációjaként
2. ez az előállítás egyértelmű (unikális)

Legyen adva egy V vektortér, amelynek v_1, v_2, \dots, v_n részhalmaza bázist alkot, ha a részalmaz vektorai lineárisan függetlenek, és bármely vektor hozzáadása ezen részalmazhoz lineárisan függővé teszi a részalmazt.

Az A.19. ábrán a bázisvektorok különböző problémái láthatók. Az (a) esetben v_2 nem állítható elő e_1, e_2 bázisvektorokból, vagyis nincs elég bázisvektor, e_1, e_2 nem függetlenek. A (b) esetben e_1, e_2, e_3 három vektor ugyan elegendő, de a vektorok előállítása többféleképpen is lehetséges, vagyis a bázisnak vélt vektorok nem függetlenek. A (c) esetben pedig e_1, e_2 bázisvektorok, mert a síkban bármely vektor előállítható lineáris kombinációjukból.

Ha a bázisvektorok egységvektorok, akkor a bázis normált bázis. Ha a bázisok páronként ortogonálisak, akkor a bázis ortogonális. Ha mindkét előbbi állítás fennáll, akkor a bázis ortonormált.



A.19. ábra. Túl kevés, túl sok, és megfelelő számú bázisvektor

Egy vektor dimenziószámát többnyire úgy határozzuk meg, hogy megszámloljuk a koordinátáinak a számát. A korrekt válasz az, hogy egy vektortér dimenzióját a bázisvektorok száma határozza meg.

Legyenek e_1, e_2, \dots, e_n egységvektorok egy n dimenziós vektortérben. Jelöljük őket sorvektorokként $e_1 = (1, 0, 0, \dots, 0)^*$; $e_2 = (0, 1, \dots, 0)^*$; $e_n = (0, 0, \dots, 1)^*$, amiknek transzponáltjait egymás mellé írva egy n dimenziós egységmátrixot kapunk:

$$\mathbf{E} = (e_1, e_2, \dots, e_n) = \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & & & \\ 0 & 0 & \dots & 1 \end{pmatrix}$$

Írjuk fel bázisvektorok segítségével az \mathbf{x} vektort:

$$\mathbf{x} = \sum_{i=1}^n x_i e_i$$

vagyis rövidítve: $\mathbf{x} = \mathbf{E}\mathbf{x}_E$ alakban írható, ahol \mathbf{x}_E koordinátái \mathbf{E} bázisra nézve.

Ugyanennek a vektortérnek egy másik bázisa z_1, z_2, \dots, z_n . \mathbf{x} vektor ezzel a bázissal felírva:

$$\mathbf{x} = \sum_{i=1}^n a_i z_i$$

vagyis rövidítve: $\mathbf{x} = \mathbf{Z}\mathbf{x}_Z$ alakban írható, ahol \mathbf{x}_Z koordinátái \mathbf{Z} bázisra nézve. E két összefüggésből adódik, hogy

$$\mathbf{x}_E = \mathbf{Z}\mathbf{x}_Z$$

illetve

$$\mathbf{x}_Z = \mathbf{Z}^{-1} \mathbf{x}_E$$

Általában kimondhatjuk, hogy ha az \mathbf{x} vektor \mathbf{Z} bázisra vonatkozó koordinátái \mathbf{x}_Z , \mathbf{W} bázisra vonatkozó koordinátái pedig \mathbf{x}_W , akkor a két bázisbeli koordináták közti összefüggés:

$$\mathbf{x}_W = \mathbf{W}^{-1} \mathbf{Z} \mathbf{x}_Z = \mathbf{B} \mathbf{x}_Z$$

ahol \mathbf{B} mátrix az u.n. báziscsere mátrixa

Lineáris transzformációk

Legye $\mathbf{x} = \{x_1, x_2 \dots x_n\}$ és $\mathbf{y} = \{y_1, y_2 \dots y_n\}$ az \mathbf{E}_n bázis által meghatározott n -dimenziós vektortér egy-egy vektora. Tegyük fel, hogy fennáll a vektorok között a következő összefüggés:

$$\mathbf{y} = \mathbf{A} \mathbf{x}$$

ahol $\mathbf{A} = \{a_{ij}\}$, amely transzformáció \mathbf{x} vektort \mathbf{y} vektorba viszi át (képezi le). A transzformáció megadható lineáris kombinációk együtthatóiból alkotott mátrixszal.

Legyen megadva egy lineáris transzformáció az \mathbf{A} mátrixszal, ami \mathbf{x} vektort \mathbf{y} vektorba képezi le, valamint egy \mathbf{B} lineáris transzformáció, amely \mathbf{y} vektort \mathbf{z} vektorba képezi le, továbbá \mathbf{C} lineáris transzformáció, amely \mathbf{z} vektort \mathbf{v} vektorba képezi le. Ekkor

$$(\mathbf{B}\mathbf{A})\mathbf{x} = \mathbf{z}$$

illetve

$$(\mathbf{C}\mathbf{B}\mathbf{A})\mathbf{x} = \mathbf{v}.$$

Elmondható tehát, hogy több lineáris transzformáció egymás után történő alkalmazása egyes transzformációk mátrixainak szorzatával állítható elő, vagyis ezen mátrixok szorzata egyetlen mátrixszal való szorzással helyettesíthető. Ez a mátrix úgy állítható elő, hogy az első transzformációs mátrixot balról szorozzuk a soron következő transzformációs mátrixszal, és így tovább.

Ha az \mathbf{A} mátrix nem szinguláris (azaz $\det \mathbf{A} \neq \mathbf{0}$), akkor létezik az inverz transzformáció is:

$$\mathbf{x} = \mathbf{A}^{-1} \mathbf{y}$$

amely az \mathbf{y} képvektorhoz az eredeti \mathbf{x} vektort rendeli. Az eredeti és leképezett vektor közötti kapcsolat kölcsönösen egyértelmű.

Vegyük észre, hogy a transzformációra felírt összefüggések nemcsak úgy foghatók fel, mint két vektornak azonos bázisra vonatkozó összefüggése, hanem úgy is, mint ugyanazon vektornak két különböző bázisra vonatkozó előállítás. Ez a báziscsere, amelyet gyakran nevezünk koordináta rendszer transzformációnak is. (A térinformatikában ezt a műveletet hajtjuk végre georeferáláskor.) Ennek fényében vizsgáljuk meg a következő gondolatmenetet:

Legyen az \mathbf{x}_Z vektornak \mathbf{Z} bázisra vonatkozó \mathbf{A} mátrixszal megadott lineáris transzformáció utáni képe \mathbf{y}_Z , azaz $\mathbf{y}_Z = \mathbf{A}\mathbf{x}_Z$. A báziscsere utáni vektorokat $(\mathbf{x}_Z, \mathbf{y}_Z)$ transzformáljuk \mathbf{W} bázisra vonatkoztatva $(\mathbf{x}_W, \mathbf{y}_W)$. Ekkor létezik egy olyan \mathbf{B} mátrix ($\mathbf{B} = \mathbf{W}^{-1}\mathbf{Z}$), amelyre igaz, hogy $\mathbf{x}_W = \mathbf{B}\mathbf{x}_Z$ és $\mathbf{y}_W = \mathbf{B}\mathbf{y}_Z$, valamint $\mathbf{x}_Z = \mathbf{B}^{-1}\mathbf{x}_W$ és $\mathbf{y}_Z = \mathbf{B}^{-1}\mathbf{y}_W$, amely alapján

$$\mathbf{y}_W = \mathbf{B}\mathbf{y}_Z = \mathbf{B}\mathbf{A}\mathbf{x}_Z = \mathbf{B}\mathbf{A}\mathbf{B}^{-1}\mathbf{x}_W = \mathbf{C}\mathbf{x}_W$$

Így tehát az új rendszerben $\mathbf{y}_Z = \mathbf{A}\mathbf{x}_Z$ transzformációnak az $\mathbf{y}_W = \mathbf{C}\mathbf{x}_W$ felel meg, ahol $\mathbf{C} = \mathbf{B}\mathbf{A}\mathbf{B}^{-1}$

A.4. Komplex számok

A következőkben röviden áttekintjük a számfogalom fejlődését, a természetes számoktól a komplex számokig.

A számfogalom fejlődése

A szám fogalma absztrakció útján jött létre, a világ megfigyelése révén. Először a természetes számok keletkeztek a dolgok megszámlálása által. Ezek az 1,2,3 stb. vég nélkül folytatható sorozatot alkotnak. Egyből kiindulva számlálással egyre újabb természetes számokat kapunk ad infinitum. Tekintsük át a természetes számokkal való műveleteket.

Természetes számok (\mathbb{N})

- **Összeadás:** Legyen a és b természetes szám. a -ból kiindulva számláljunk tovább b lépésben. Azt a számot, amelyet a számlálás eredményeként kapunk, nevezzük a és b természetes szám összegének, és jelöljük $a + b$ -vel. Az összeadás kommutatív (azaz felcserélhető), ugyanis $a + b = b + a$.

Legyen a , b és c természetes számok. Összegük: $a + b + c$. Az összeadás asszociatív (csoportosítható), ugyanis $(a + b) + c = a + (b + c)$

- **Szorzás:** Ha az összeg minden tagja ugyanaz a természetes szám, akkor a művelet neve szorzás, jelölése: $a + a + a = 3 \cdot a$. Az $a \cdot b$ olyan összeg, amelynek a darab tagja van, és azok mindegyike b . Egytagú összeg esetén $1 \cdot b = b$. A szorzás is asszociatív és kommutatív, vagyis

$$a \cdot b = b \cdot a, \quad (a \cdot b) \cdot c = a \cdot (b \cdot c)$$

A szorzás az összeadásra nézve disztributív (széttagolható), vagyis

$$(a + b)c = (ac) + (bc)$$

- **Hatványozás:** Ahogy az összeadásból kaptuk a szorzást, hasonlóan leszámaztatható a hatványozás a szorzásból: ha egy b tényező szorzat mindegyik tényezője az a természetes szám, akkor ez a művelet hatványozás, melyet a^b jelölünk. Ha egytényezős a szorzat, akkor $a_1 = a$. A hatványozás nem kommutatív, vagyis $a^b \neq b^a$ (kivéve speciális eseteket). Ugyancsak nem érvényes a disztributivitás sem.
- **Kivonás:** A kivonás az összeadás inverz művelete, amelynek során ismerjük az összeget és az egyik tagot, keressük a másikat, vagyis adott a és b természetes számok, amelyhez keressük azt az x természetes számot, amelyre teljesül, hogy $a + x = b$, jele: $x = b - a$. A természetes számok körében a kivonás nem mindig végezhető el, mert ha $b < a$, akkor az eredmény nem természetes szám, vagyis a kivonás nevű inverz művelet kivezet a természetes számok köréből.
- **Osztás:** Az osztás a szorzás inverz művelete. Adott a és b természetes számokhoz keressük azt az x természetes számot, amelyre teljesül, hogy $a \cdot x = b$. A művelet neve osztás, jele: $x = b/a$. Az osztás legtöbbször nem végezhető el a természetes számok körében, vagyis az osztás kivezet a természetes számok köréből.
- **Gyökvonás, logaritmus:** A hatványozás, mint tudjuk nem kommutatív, így kétféle inverz művelete is van. Ha ismert a hatvány, a hatványkitevő, és keressük az alapot, akkor a művelet neve gyökvonás. Ha ismert a hatvány, az alap, és keressük a kitevőt, akkor logaritmusról beszélünk. A természetes számok körében általában egyik inverz művelet sem végezhető el.

Egész számok (\mathbb{Z}), racionális számok (\mathbb{Q}), valós számok (\mathbb{R})

Az inverz műveletek kivezettek minket a természetes számok köréből, ezért szükségessé vált egy absztraktabb szám bevezetése, a számkör bővítése, amelyre érvényben maradnak a természetes számokra vonatkozó műveletek, és az új számfogalomban benne vannak a természetes számok. Az egész számok bevezetése által létrejöttek a kivonás inverz művelete minden természetes számra elvégezhetővé vált, Az új számok tehát olyanok legyenek, hogy a természetes számok az új számok között legyenek; az új számok körében elvégezhető legyen a kivonás; az új számok körében végzett műveleteket természetes számokra alkalmazva ugyanazt kell kapni, mintha a természetes számokra eredetileg vonatkozó

műveleteket alkalmaztuk volna; a természetes számokra vonatkozó azonosságok lehetőleg maradjanak érvényben; műveletek permanenciájának elve: a természetes számok körében az $a - a$ kivonás nem végezhető el, ezért az új számok körében megjelenik a nulla. Az új számok körében tehát egy adott a számhoz létezik egy $-a$ -val jelölt szám, amelyre igaz, hogy $a + (-a) = 0$. Az így kapott számokat egész számoknak nevezzük. A rájuk vonatkozó műveletek összeadás, kivonás, szorzás, osztás, hatványozás.

A kivonás, elvégezhető az egész számok körében, de az osztás, mint inverz művelet többnyire nem, vagyis, szükségessé válik a racionális számok bevezetése, amelyek két egész szám hányadosaként írhatók fel. A racionális számok tehát a/b alakú kifejezések, ahol a és b egész számok, és $b \neq 0$. A racionális számokkal való műveletek kommutatívok, asszociatívok, és a szorzás az összeadásra nézve disztributív.

Vannak olyan számok, amelyek nem írhatók fel két egész szám hányadosaként. Ezek az irracionális számok. A racionális és az irracionális számokat együttesen valós számoknak nevezzük. A valós számok műveletei a következők:

- **Összeadás, szorzás:** az összeadás és a szorzás a valós számok körében korlátlanul és egyértelműen elvégezhető. Mindkét művelet kommutatív és asszociatív, valamint a szorzás az összeadásra nézve disztributív. Pozitív egész n hatványkitevőre a hatványozás mint n -szer ismételt szorzás elvégezhető.
- **Kivonás:** a kivonás korlátlanul és egyértelműen elvégezhető művelet.
- **Osztás:** az osztás is korlátlanul és egyértelműen elvégezhető művelet, kivéve a 0-val való osztást.
- **Hatványozás:** negatív szám páros kitevőjű hatványa pozitív, páratlan kitevőjű hatványa negatív szám.
- **Gyökvonás:** adott hatvány és adott kitevő esetén az alapot keressük. Eddig a hatványkitevő mindig természetes szám volt, vagyis $a^n = b$. Tetszőleges n természetes szám és a valós szám esetén a gyök: $a = {}^n\sqrt{b}$. A gyökvonást nem mindig lehet elvégezni, és ha el lehet, akkor sem mindig egyértelmű. Pozitív alap esetén egy és csak egy olyan pozitív b valós szám létezik, amelyre igaz, hogy $b^n = a$. Mivel páros n esetében $b^n = (-b)^n$, így ebben az esetben a gyökvonás egy kétértékű művelet. AZért, hogy ezt elkerüljük, megállapodunk abban, hogy eredményül csak a pozitív b értéket vesszük figyelembe. Mivel negatív szám páros kitevőjű hatványa pozitív, ezért negatív számból páros kitevőjű gyököt nem lehet vonni (de páratlan kitevőjű gyököt lehet).
- **Logaritmus:** Mint láttuk, a hatványozás nem kommutatív, vagyis két inverz művelet létezik. Az egyik a gyökvonás, a másik a logaritmus, amikor az alap

és hatvány ismeretében keressük a kitevőt. Az $b = a^x$ művelet inverze (logaritmus) az $x = {}^a \log b$. A logaritmusnak csak pozitív alap és logaritmálandó szám esetén van értelme.

Komplex számok

A valós számok körében nem lehet minden másodfokú egyenletet megoldani. Például nincs gyöke az $x^2 + 1 = 0$ egyenletnek, hiszen nincs olyan valós szám, amelynek a négyzete -1 lenne. Próbáljuk meg kiterjeszteni a valós számok körét egy olyan számkörre, ahol a fenti egyenletnek is van megoldása. Jelöljük ezt a számot i -vel ($i = \sqrt{-1}$). A kiterjesztést úgy kell végrehajtanunk – mint eddig bármikor, amikor valamely számkört bővítettük –, hogy az új számok összege és szorzata is szám legyen, vagyis $a + bi$.

Alkalmazzuk a valós számok műveleteit ezekre az új számokra:

$$(a + bi) + (c + di) = a + c + bi + di = (a + c) + (b + d)i$$

$$(a + bi)(c + di) = ac + adi + bic + bdi^2 = ac - bd + (ad + bc)i$$

Az új számokat tehát két valós számmal lehet jellemezni, és a műveletek a fentiek szerint kell végrehajtani. Ezen új számok neve komplex számok. Nézzük meg a tulajdonságaikat. Legyen $(a; b)$ valós számokból álló számpár

$$(a; b) = (c; d) \text{ akkor és csak akkor, ha } a = c \text{ és } b = d$$

$$(a; b) + (c; d) = (a + c; b + d) \text{ és } (a; b)(c; d) = (ac - bd; ad + bc)$$

A fenti összeadás és szorzás kommutatív, asszociatív és disztributív, vagyis a komplex számok a valós számoknál bővebb számkört alkotnak. Továbbá mindig elvégezhető a kivonás: $(a; b) - (c; d) = (a - c; b - d)$, valamint az osztás.

Vizsgáljuk meg, hogy a komplex számokkal hogyan kaphatók vissza a valós számok. Az $(a; 0)$ alakú komplex számok nyilvánvalóan leírják a valós számokat, mert

$$(a; 0) + (b; 0) = (a + b; 0)$$

$$(a; 0)(b; 0) = (ab - 0; 0 + 0) = (ab; 0)$$

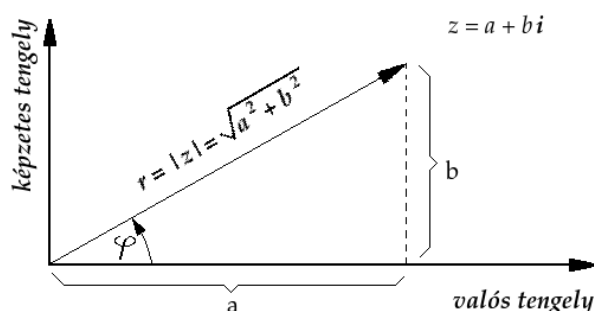
Figyelembe véve, hogy $i = 0 + 1i$ írható, hogy $(a; 0) + (b; 0)(0; 1) = a + bi$.

A $z = a + bi$ komplex szám konjugáltjának nevezzük a $\bar{z} = a - bi$ komplex számot.

Összefoglalva tehát elmondható, hogy a számfogalmat úgy bővítettük, hogy lehetőleg minden igaz maradjon az új számokra is. Ez az óhaj általában nem lehetséges.

A komplex számok geometria jelentése

A komplex számok számpárokkal történő jellemzése adja a geometriai jelentést. A sík pontjait is számpárokkal jellemezhetjük, ezért kézenfekvően adódik a sík



A.20. ábra. A z komplex szám abszolút értéke, $|z| = \sqrt{a^2 + b^2}$, a φ pedig az arcusa ($\text{arc}(z)$)

pontjainak komplex számokkal történő leírása. Vegyünk fel egy koordináta rendszer, amelynek abszcisszája legyen az ún. valós tengely, az ordinátája pedig az ún. képzetes tengely. Feleltessük meg az $(a; b)$ koordinátájú pontot egy komplex számnak (A.20. ábra).

A komplex szám trigonometrikus alakja $z = r(\cos\varphi + i\sin\varphi)$. Vizsgáljuk meg, hogy mi lesz a komplex konjugált, vagyis lehet-e geometriai jelentést tulajdonsítani a $z = r(\cos\varphi - i\sin\varphi)$ komplex számnak. A A.20. ábra segítségével belátható, hogy mivel $\cos(-\varphi) = \cos\varphi$ és $\sin(-\varphi) = -\sin(\varphi)$, így a z komplex konjugált a komplex szám valós tengelyre vett tükörképe.

Komplex számok arcusa úgy viselkedik, mintha hatványkitevő lenne, ezért z felírható $z = re^{i\varphi}$ alakban is. Ennek belátásához egy pillanatra forduljunk el a komplex számoktól a sorfejtések felé.

Taylor-sor

A matematika alkalmazásának egy igen fontos momentuma a függvények sorbafejtése. A továbbiakban arra törekszünk, hogy a \sin , \cos , stb. függvényeket a leg-egyszerűbb racionális egész függvények sorával állítsuk elő, amelyek $a + bx$ vagy $a + bx + cx^2 + dx^3 + \dots$ alakúak. Ha általában $f(x)$ valamely függvénye az x -nek és ilyen formában akarjuk előállítani, akkor nem várhatjuk, hogy másod-, harmad- vagy negyedfokú stb. racionális egészfüggvényre jussunk, mert akkor $f(x)$ éppen racionális egészfüggvény volna. Ilyenkor tehát azt próbáljuk meg, hogy $f(x)$ -et végtelen sor segítségével állítsuk elő a következő formában:

$$f(x) = A_0 + A_1x + A_2x^2 + A_3x^3 + \dots$$

A feladat $A_0, A_1, A_2 \dots$ együtthatók meghatározása. Ez a következőképpen történhet: helyettesítsünk x helyére 0-t, vagyis A_0 a függvény 0 helyen felvett értéke:

$$f(0) = A_0$$

Állítsuk elő $f(x)$ sorfejtés mindkét oldalának deriváltját:

$$f'(x) = A_1 + 2 \cdot A_2x + 3 \cdot A_3x^2 + 4 \cdot A_4x^3 +$$

Helyettesítsünk x helyébe 0-t. Így tehát $A_1 = f'(0)$. Folytassuk tovább a deriválást. A második derivált

$$f''(x) = 2 \cdot A_2 + 3 \cdot 2 \cdot A_3x + 4 \cdot 3 \cdot A_4x^2 + 5 \cdot 4 \cdot A_5x^3 + \dots$$

Az $x = 0$ helyettesítés révén $A_2 = f''(0)/2$

Az eljárást folytatva kapjuk a magasabb deriváltakra és együtthatókra, hogy

$$A_4 = \frac{f^{IV}(0)}{2 \cdot 3 \cdot 4}, \quad A_5 = \frac{f^V(0)}{2 \cdot 3 \cdot 4 \cdot 5}$$

és így tovább. Ha tehát ismerjük a szóban forgó függvény differenciál hányadosait az $x = 0$ helyen, akkor $f(x)$ függvény előállítható ilyen végtelen sor alakban, melynek neve Taylor-sor. Ha ez a végtelen sor konvergens, akkor fölhasználhatjuk az $f(x)$ függvény értékeinek a kiszámításához.

Ezek után próbáljuk meg előállítani a \sin és a \cos függvényt Taylor-sor segítségével.

sin x

A $\sin x$ függvény Taylor-sora a következő:

$$\begin{array}{ll} (\sin x)' = \cos x & (\sin x)^v = \cos x \\ (\sin x)'' = -\sin x & (\sin x)^{vi} = -\sin x \\ (\sin x)''' = -\cos x & (\sin x)^{vii} = -\cos x \\ (\sin x)^{iv} = \sin x & (\sin x)^{viii} = \sin x \end{array}$$

és így tovább. Látható, hogy a negyedik deriváltól kezdve ugyanaz az alak ismétlődik. Helyettesítsük be az $x = 0$ értéket. Ekkor

$$\begin{array}{ll} (\sin x)_{x=0} = 0 & \\ (\sin x)'_{x=0} = 1 & (\sin x)^v_{x=0} = 1 \\ (\sin x)''_{x=0} = 0 & (\sin x)^{vi}_{x=0} = 0 \\ (\sin x)'''_{x=0} = -0 & (\sin x)^{vii}_{x=0} = -1 \\ (\sin x)^{iv}_{x=0} = 0 & (\sin x)^{viii}_{x=0} = 0 \end{array}$$

Így tehát

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots$$

cos x

A $\cos x$ függvény Taylor-sora a következő:

Mivel tudjuk, hogy $\cos x$ a $\sin x$ deriváltja, ezért deriváljuk a $\sin x$ Taylor-sorát:

$$\cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \dots$$

Az e^x függvény Taylor-sora

A felsőbb matematika egyik legfontosabb függvénye az Euler-féle $\varphi(x)$ függvény, amely a következő végtelen sorként állítható elő:

$$\varphi(x) = 1 + \frac{x}{1} + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \frac{x^5}{5!} + \frac{x^6}{6!} + \dots$$

Állítsuk elő $\varphi(x)$ deriváltját:

$$\varphi'(x) = 1 + \frac{x}{1} + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \frac{x^5}{5!} + \dots$$

vagyis $\varphi'(x) = \varphi(x)$. Ebből az látható, hogy $\varphi(x)$ függvény minden deriváltja önmaga.

Vizsgáljuk meg $\varphi(x)^k$ -t, azaz a k -adik hatványának viselkedését a deriváltra vonatkozóan. Az első derivált $k\varphi(x)^{k-1}\varphi'(x)$. Mivel azonban $\varphi'(x) = \varphi(x)$, ezért a derivált $k\varphi^k$. Minden egyes deriválás a kitevővel való újabb szorzást jelent. A $\varphi(x)^k$ Taylor-sora tehát:

$$\varphi(x)^k = 1 + \frac{kx}{1} + \frac{(kx)^2}{2} + \frac{(kx)^3}{3} + \frac{(kx)^4}{4} + \dots$$

vagyis a $\varphi(x)$ k -adik hatványát úgy kaptuk meg $\varphi(x)$ Taylor sorából, hogy x helyébe kx -et helyettesítettünk.

Ha most az $x = 1$ helyettesítést alkalmazzuk, akkor

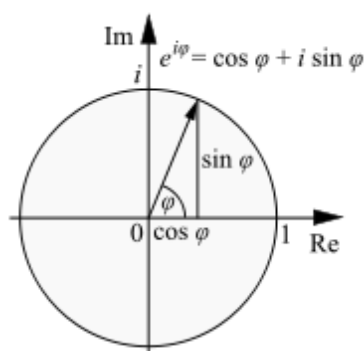
$$\varphi(1)^k = 1 + \frac{k}{1} + \frac{k^2}{2!} + \frac{k^3}{3!} + \frac{k^4}{4!} + \frac{k^5}{5!} + \frac{k^6}{6!} + \dots$$

valamint k helyére x -et téve

$$\varphi(1)^x = 1 + \frac{x}{1} + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \frac{x^5}{5!} + \frac{x^6}{6!} + \dots$$

Ebből pedig következik, hogy

$$\varphi(1)^x = \varphi(x)$$



A.21. ábra. Az Euler-összefüggés geometriai jelentése. *Re* a valós tengely, és *Im* a képzetes tengely

$\varphi(1)$ nem más, mint a

$$\varphi(1) = 1 + \frac{1}{1} + \frac{1}{2!} + \frac{1}{3!} + \frac{1}{4!} + \frac{1}{5!} + \frac{1}{6!} + \dots$$

végtelen sor, amelynek összegét e -vel jelöljük, közelítő értéke 2,718281828.... Az előbbi egyenlet tehát felírható a következő módon is:

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \frac{x^5}{5!} + \frac{x^6}{6!} + \dots$$

Az Euler-formula

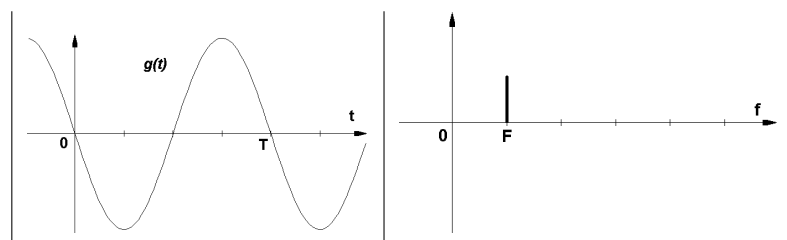
Végezetül térjünk vissza a komplex számokhoz, és hasonlítsuk össze a $\sin x$ és a $\cos x$ függvények Taylor-sorát az e^x Taylor-sorával. Belátható, hogy

$$e^{ix} = \cos x + i \cdot \sin x \quad e^{-ix} = \cos x - i \cdot \sin x$$

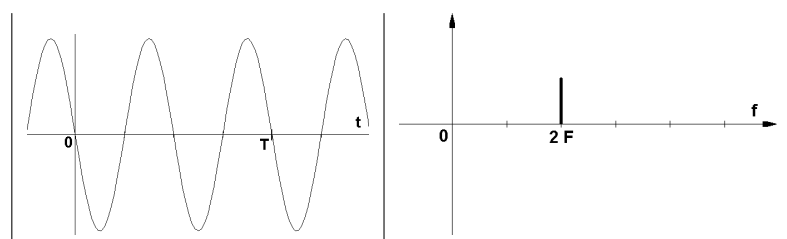
illetve

$$\cos x = \frac{e^{ix} + e^{-ix}}{2} \quad \sin x = \frac{e^{ix} - e^{-ix}}{2}$$

Az $e^{ix} = \cos x + i \cdot \sin x$ összefüggést Euler-formulának nevezzük. Geometria jelentését mutatja az A.21. ábra.



A.22. ábra. A T periódusidejű szinuszfüggvény az időtartományban és a frekvenciatartományban



A.23. ábra. A $2F$ frekvenciájú szinuszfüggvény az időtartományban és a frekvenciatartományban

A.5. A digitális szűrések matematikai alapjai

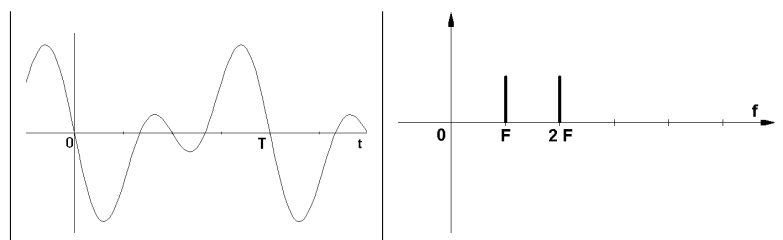
Időtartomány, frekvenciatartomány

Jelölje $g(t)$ az idő T periódusidejű szinuszfüggvényét. Ábrázoljuk $g(t)$ -t, mint az idő függvényét (A.22. ábra bal oldala). A T periódusidő reciproka a frekvencia: $F = 1/T$. Ábrázoljuk a szinuszfüggvényt, mint a frekvencia függvényét (A.22. ábra jobb oldala). Hasonlítsuk össze a A.22. ábra két oldalát. Látható, hogy az időtartománybeli szinuszfüggvény „képe” a frekvencia tartományban egyetlen érték, a szinuszfüggvény F frekvenciája. Az érthetőség kedvéért végezzük el ezt az összehasonlítást néhány másik periódikus függvényre is (A.23. ábra).

Most adjuk össze az előbbi két szinuszfüggvényt, és vizsgáljuk meg az összeg viselkedését az időtartományban és a frekvencia tartományban (A.24. ábra). A A.24. ábra jobb oldalán látható függvényt a bal oldali spektrumának nevezzük. A spektrumból leolvasható, hogy az F és $2F$ frekvenciájú harmonikus összetevőkből áll a bal oldali periódikus függvényünk.

Fourier-sorfejtés

Legyen $g(t)$ egy T periódusidejű periódikus függvény, azaz $g(t) = g(t + kT)$. Fourier kimutatta, hogy a minden periódikus függvény felírható a következő sorfejtés segítségével:



A.24. ábra. A $\sin(t)$ és a $\sin(2t)$ függvény összege az időtartományban és a frekvenciatartományban

$$g(t) = a_0 + \sum_{k=1}^{\infty} 2a_k \cos\left(\frac{2\pi}{T}kt\right) + \sum_{k=1}^{\infty} 2b_k \sin\left(\frac{2\pi}{T}kt\right)$$

ahol $\cos(2\pi/T)t$ és $\sin(2\pi/T)t$ az ú.n. alapharmonikusok, míg a sorfejtés további függvényei a k -adik felharmonikusok, az a_1, \dots, a_k együtthatók pedig az ú.n. Fourier együtthatók. Az együtthatók meghatározhatók a következő módon:

$$a_k = \frac{1}{T} \int_a^{a+T} g(t) \cos\left(\frac{2\pi}{T}kt\right) dt, \quad b_k = \frac{1}{T} \int_a^{a+T} g(t) \sin\left(\frac{2\pi}{T}kt\right) dt$$

Másképpen fogalmazva a Fourier-sor bármely tagjához tartozó a_0, a_k, b_k együtthatók arányosak a sorbafejtendő függvény és a sorfejtésben szereplő periódikus függvény szorzatának a T periódushosszra vett átlagával (integráljával).

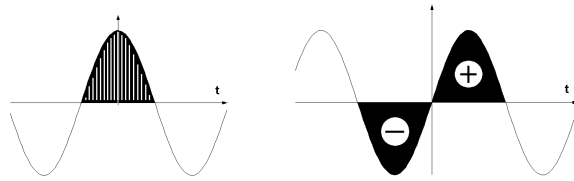
Nézzünk meg egy periódikus függvényt, amit jelöljünk $g(t)$ -vel, és ábrázoljuk az együtthatókat, vagyis a függvény spektrumát. A Fourier-sorfejtést az Euler-formulára való tekintettel komplex alakban is felírhatjuk:

$$g(t) = \sum_{n=-\infty}^{\infty} C_n e^{i2\pi nt/T},$$

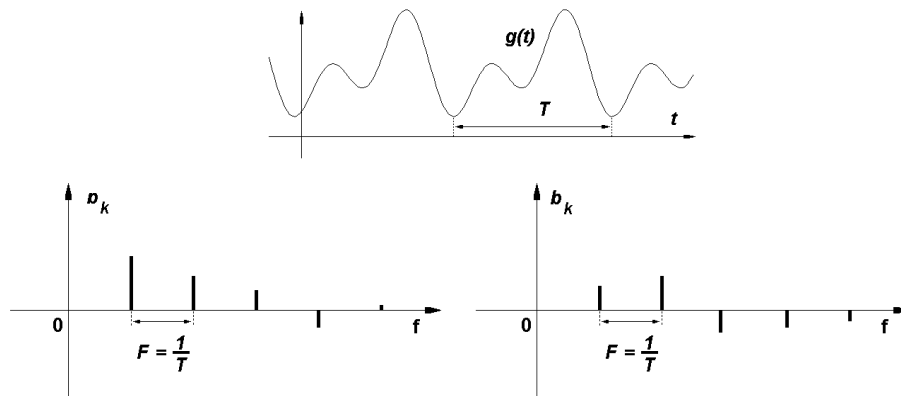
ahol C_n a komplex Fourier együtthatók.

Páros és páratlan függvények

Nevezzük párosnak azokat a függvényeket, amelyekre az $s(t) = s(-t)$, és páratlanoknak, amelyekre az $s(t) = -s(-t)$ összefüggés igaz (A.25. ábra). Fontos összefüggés, hogy a páratlan függvények origóra szimmetrikus intervallumon képzett integrálja 0, mivel a + és - jelű területek kiejtik egymást az összegzés során, míg a páros függvényekre ugyanez az origótól jobbra eső tartományra vett integrál kétszerese. Ebből következik, hogy páros függvények Fourier-sora csak koszinusz függvényt tartalmaz, míg páratlan függvényeké csak szinuszos tagokat tartalmaz.



A.25. ábra. Páros és páratlan függvények

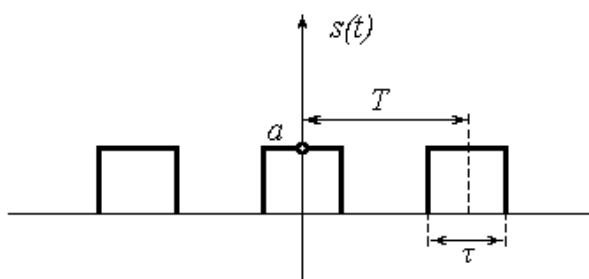
A.26. ábra. Egy periódikus függvény, az amplitúdóspektruma (a_k) és a fázisspektruma (b_k)

Periodikus függvények spektruma vonalas, vagyis az F frekvencia egészszámú többszöröseinek megfelelő frekvenciák fordulhatnak csak elő a sorfejtésben (A.26. ábra).

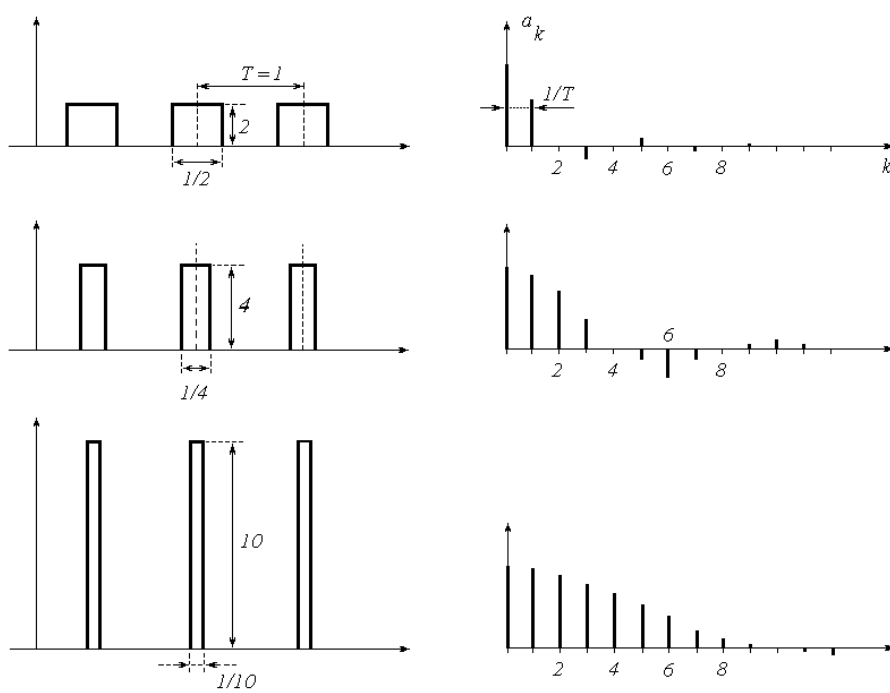
Vizsgáljuk meg a négyszög impulzus sorozat (A.27. ábra) Fourier-sorát. A T periódus idejű, τ hosszúságú és a magasságú négyszög impulzus Fourier együtthatóinak kiszámításakor használjuk ki, hogy a négyszög impulzus függvény páros, vagyis a Fourier együtthatók csak koszinuszos tagokat tartalmaznak, vagyis a $b_k = 0$ minden k -ra.

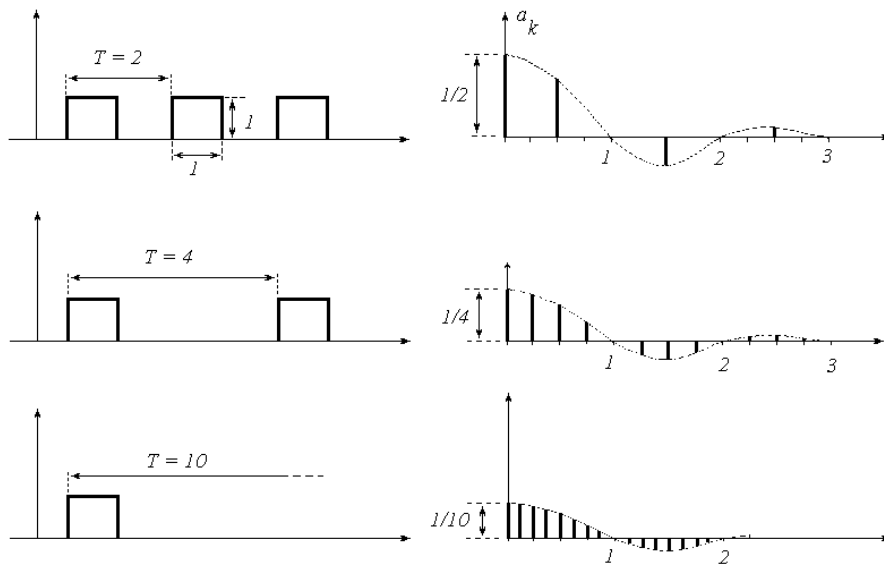
Időzzünk el kicsit a kapott eredménynél. Vizsgáljuk meg a formula viselkedését T , τ és a változtatásával. Állítsuk be úgy τ és a értékét, hogy az impulzusok területe egységnyi legyen, T -t pedig az egyszerűség kedvéért válasszuk egynek. Ekkor a függvény és spektruma a következőképpen alakul: ahogy keskenyedik a négyszög impulzus sorozat, úgy szélesedik a spektruma. A spektrum szélesedés egyben azt is jelenti, hogy minél keskenyebb a jelsorozat, annál több nagyfrekvenciás összetevői vannak a jelnek.

Most vizsgáljuk meg azt az esetet, amikor a T periódusidő változásának hatására vagyunk kíváncsiak, ellenben a jelek hosszát állandóan tarjuk. Az egységnyi területű impulzus sorozat spektruma egyre sűrűbbé válik az impulzusok



A.27. ábra. A négyzög impulzus sorozat

A.28. ábra. A négyzög impulzus sorozat jelszélessége (τ) változásának hatása spektrumra



A.29. ábra. A négyzet impulzus sorozat jelhossza (T) változásának hatása spektrumra

távolságának növelésével, ami azt jelenti, hogy egyre több nagyfrekvenciás komponens jelenik meg. Minden határon túl növelve a jel hosszát (vagyis $T \rightarrow \infty$), a spektrum vonalak végtelen sűrűn fognak elhelyezkedni, vagyis a spektrum folytonossá válik. A $T \rightarrow \infty$ egyben azt is jelenti, hogy a periódikus függvény nem periódikussá, azaz transzienssé vált. Kimondható tehát, hogy nem periódikus függvények spektruma folytonos.

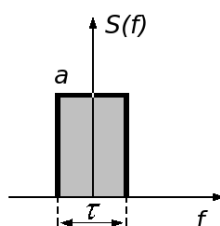
Fourier-transzformáció

A Fourier-transzformációt részletesen tárgyalja számos, az irodalomjegyzékben jelzett könyv, tehát csak érintőlegesen mutatjuk be a legfontosabb összefüggéseket. Legyen $s(t)$ az idő nem periódikus függvénye. Írjuk fel a következő kifejezést:

$$S(f) = \int_{-\infty}^{\infty} s(t)e^{-2\pi ift} dt,$$

ahol $S(f)$ az $s(t)$ függvény Fourier-transzformáltja, vagyis a frekvenciatartománybeli „képe”, f a frekvencia és $i = \sqrt{-1}$. $S(f)$ -t gyakran nevezik $s(t)$ spektrumának is, az eljárást pedig Fourier-transzformációnak. Írjuk fel a következő kifejezést, amelyet inverz Fourier-transzformációnak is neveznek:

$$s(t) = \int_{-\infty}^{\infty} S(f)e^{2\pi ift} df,$$



A.30. ábra. A négyszögimpulzus az időtartományban

A Fourier transzformált létezésének feltétele, hogy $s(t)$ függvény szakaszonként sima legyen (vagyis deriváltja véges számú hely kivételével folytonos), és az $\int_{-\infty}^{\infty} |s(t)| dt$ kifejezés konvergens legyen. Ebben az esetben az $S(f)$ Fourier-transzformált meghatározható.

Az $s(f)$ függvény spektrumából, $S(f)$ -ből számítsuk ki az ún. amplitúdóspektrumot és a fázisspektrumot a következő módon. Az amplitúdóspektrum $S(f)$ abszolútértéke lesz, míg a fázisspektrum, $S(f)$ -nek f -től függő irányszöge lesz. Jelölje a spektrum valós részét S_v , a képzetes részét S_k . Ekkor az amplitúdóspektrum a következő lesz:

$$|S(f)| = \sqrt{S_v^2(f) + S_k^2(f)},$$

míg a fázisspektrum

$$\arctan \frac{S_k(f)}{S_v(f)}.$$

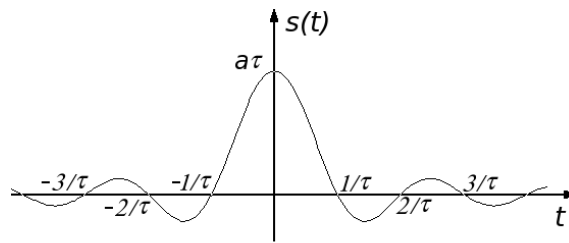
Néhány speciális függvény Fourier-transzformáltja

A szűrési eljárások bemutatásakor szükségünk lesz néhány speciális függvény Fourier-transzformáltjára. Az egyik ilyen kiemelten fontos függvény a négyszög függvény.

Négyszög impulzus. Az úgynevezett tranzienst függvények az érdekesek a számunkra, vagyis azok, amelyek nem periodikusak, hanem véges hosszúságú intervallumon különböznek a nullától (ilyenek például a digitális képek, domborzat modellek). Vizsgáljuk meg néhány tranzienst függvény Fourier-transzformáltját. Az egyik fontos függvény a négyszög impulzus (A.30. ábra), amely a következő:

$$s(t) = a, \quad \text{ha } |t| < \tau/2 \\ = 0, \quad \text{egyébként.}$$

Az egyszerűség kedvéért legyen $a = 1/\tau$, így a négyszög impulzus terület egységnyi lesz.



A.31. ábra. Az időtartománybeli négyzög impulzus Fourier-transzformáltja a *sinc* függvény a frekvenciatartományban

Végezzük el a függvény Fourier-transzformációját:

$$S(f) = \int_{-\infty}^{\infty} s(t)e^{-2\pi ift} dt = a \int_{-\tau/2}^{\tau/2} e^{-2\pi ift} dt = \frac{\sin \pi f \tau}{\pi f \tau}$$

Ezt a függvényt szinusz kardinális függvénynek is nevezik és *sinc*-vel jelölik (A.31. ábra).

Dirac- δ . Paul Dirac vezette be a róla elnevezett δ -t pontszerű objektumok leírására. Használata megkönnyíti a digitális adatrendszerek leírását számos vonatkozásban. Definíció szerint legyen δ egy olyan függvény (klasszikus értelemben nem az, hanem úgynevezett. disztribúció), amely

$$\delta(t) = 0, \quad \text{ha } t \neq 0$$

$$\text{és} \quad \int_{-\infty}^{\infty} \delta(t) dt = 1$$

A Dirac- δ szemléletes jelentése egy végtelenül keskeny, és végtelenül magas amplitúdójú impulzus, amelynek „görbe alatti területe” egységnyi. Számunkra azért fontos a δ -val való foglalkozás, mert számos olyan tulajdonsága van, amely a jelfeldolgozás szempontjából lényeges. A $t = t_0$ esetre alkalmazva a δ -t, azt kapjuk, hogy $\delta(t - t_0) = 0$, ha $t \neq t_0$. Lássuk egyik fontos tulajdonságát, amelyet kiválasztó tulajdonságnak nevezhetünk. Szorozzunk meg az $f(t)$ függvényt a $\delta(t - t_0)$ -val. Ekkor

$$\delta(t - t_0)f(t) = \delta(t - t_0)f(t_0)$$

mivel

$$\int_{-\infty}^{\infty} \delta(t - t_0)f(t) dt = f(t_0)$$

Ez a kiválasztási képesség, vagyis a $\delta(t_0)$ az $f(t)$ függvényből kiválasztotta a t_0 -hoz tartozó értéket.

A következőkben vizsgáljuk meg egy Dirac impulzus és egy Dirac impulzus sorozat Fourier-transzformáltját. Jelölje \mathcal{F} a Fourier-transzformációt, amely a definíciója alapján:

$$\mathcal{F}\{\delta(t)\} = \int_{-\infty}^{\infty} \delta(t)e^{-2\pi ift} dt = 1$$

Nem az origóban lévő Dirac impulzus Fourier-transzformáltja pedig:

$$\mathcal{F}\{\delta(t-a)\} = \int_{-\infty}^{\infty} \delta(t-a)e^{-2\pi ift} dt = e^{-2\pi ifa}$$

amiről Euler óta tudjuk, hogy

$$e^{-2\pi ifa} = \cos 2\pi fa - i \sin 2\pi fa$$

Dirac impulzus sorozat (az időtartományban) Fourier-transzformáltja:

$$\mathcal{F}\left\{\sum_{k=-\infty}^{\infty} \delta(t-k\tau)\right\} = \frac{1}{\tau} \sum_{k=-\infty}^{\infty} \delta(f-k/\tau)$$

ahol \mathcal{F} -fel jelöltük a Fourier-transzformációt. Ennek az összefüggésnek még fontos szerepe lesz a mintavételezés tárgyalásakor.

Konvolúció

Legyenek f_1 és f_2 folytonos függvények. Jelölje konvolúciójukat $h = f_1 * f_2$, melyet a következő kifejezés definiál:

$$h(t) = f_1(t) * f_2(t) = \int_{-\infty}^{\infty} f_1(\tau)f_2(t-\tau)d\tau$$

Digitális jelekre alkalmazva az összefüggést:

$$h(t) = f_1(t) * f_2(t) = \sum_{\tau=-\infty}^{\infty} f_1(\tau)f_2(t-\tau)$$

Az eddigiekben csak egydimenziós függvényekkel foglalkoztunk. Könnyen általánosítható a konvolúció fogalma kétdimenziós függvényekre is, mint amilyen a digitális kép.

A konvolúció tulajdonságai. A következőkben röviden összefoglaljuk a konvolúció főbb tulajdonságait (bizonyítás nélkül). Egyes azonosságoknak nagy jelentősége lesz a digitális szűrések elméleti hététerét tekintve.

- A konvolúció kommutatív: $f(t) * g(t) = g(t) * f(t)$

- A konvolúció asszociatív: $f(t) * [g(t) * h(t)] = [f(t) * g(t)] * h(t)$
- A konvolúció disztributív: $f(t) * [g(t) + h(t)] = f(t) * g(t) + f(t) * h(t)$
- Szorzat Fourier-transzformáltja a spektrumok konvolúciója. Jelölje \mathcal{F} a Fourier-transzformációt. Ekkor $\mathcal{F}\{g(t)h(t)\} = G(f) * H(f)$, ahol $G(f)$ és $H(f)$ a $g(t)$ és $h(t)$ függvények spektrumai.
- Függvények konvolúciójának Fourier-transzformáltja a spektrumok szorzata: $\mathcal{F}\{g(t) * h(t)\} = G(f)H(f)$, ahol $G(f)$ és $H(f)$ az $g(t)$ és $h(t)$ függvények spektrumai
- Két függvény konvolúciójának deriváltja: $(u * v)' = u' * v$
- Konvolúció Dirac- δ -val. Konvolváljuk az $f(t)$ függvényt egy Dirac impulzussal: $f(t) * \delta(t - t_0) = f(t - t_0)$ A Dirac impulzus eltolja a függvényt a saját argumentumában szereplő helyre.
- Konvolúció végtelen Dirac- δ sorozattal

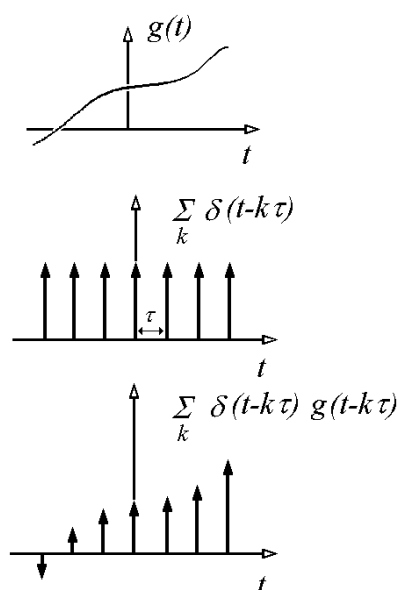
$$g(t) \sum_{k=-\infty}^{\infty} \delta(t - k\tau) = \sum_{k=-\infty}^{\infty} g(k\tau)\delta(t - k\tau)$$

Mintavételezés

A szaktudományokban alapvető jelentőségű az adatnyerés folyamata, amely voltaképpen mintavételezés, azaz analóg-digitál (AD) konverzió. Mintavételezéskor analóg jelekből állítunk elő digitális jel sorozatokat, máskor eleve diszkrét mintavételi pontokban történik az adatnyerés. A mintavételezés folyamatának megértése nagy jelentőségű, mivel az adatokból levonható következtetések függhetnek a mintavételezés módjától.

Nevezzük mintavételi távolságnak (τ) két mintavétel közötti intervallumot, amely, ha idősorokról van szó, akkor idő dimenziójú, de lehet távolság dimenziójú is, ha két mintavételi pont között térbeli távolságról van szó. Itt és most csak a szabályos mintavételt tárgyaljuk.

A következőkben vázlatosan áttekintjük a mintavételezés legfontosabb törvényszerűségeit, egyenletes mintavételezést feltételező esetekben. Az egyszerűség kedvéért idősorokkal, vagyis egy dimenziós problémákkal foglalkozunk, amik teljes mértékben általánosíthatók több dimenziós esetekre is, mint amilyen a háromdimenziós terepmodell.



A.32. ábra. A mintavételezés eredménye egy olyan impulzus sorozat, melynek tagjai az eredeti függvénynek a mintavételi helyen felvett értékei

Mintavételi tétel

Legyen τ az úgynevezett mintavételi távolság. Ábrázoljuk a $g(t)$ időfüggvényt és a mintavételezés eszközt, a Dirac-impulzusok sorozatát, majd a mintavételezés eredményét, a digitalizált időfüggvényt (A.32. ábra). A mintavételezés tehát nem más, mint a $g(t)$ időfüggvény és a Dirac impulzus sorozat szorzata:

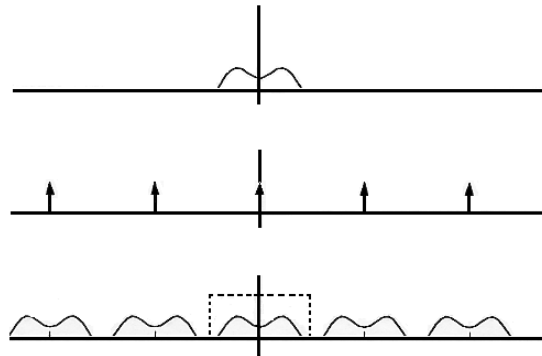
$$g(t) \sum_{k=-\infty}^{\infty} \delta(t - k\tau) = \sum_{k=-\infty}^{\infty} g(k\tau) \delta(t - k\tau)$$

Vizsgáljuk meg az analóg függvény és a mintavételezett spektrumát. Jelöljük $G(f)$ -fel az eredeti, és $G_d(f)$ -fel a digitalizált függvény spektrumát. A Dirac- δ Fourier-transzformáltjának és a konvolúció tételek felhasználásával felírható a mintavételezett függvény spektruma:

$$G_d(f) = G(f) * \frac{1}{\tau} \sum_{k=-\infty}^{\infty} \delta\left(f - \frac{k}{\tau}\right)$$

vagyis

$$G_d(f) = \frac{1}{\tau} \sum_{k=-\infty}^{\infty} G\left(f - \frac{k}{\tau}\right)$$



A.33. ábra. A mintavételezés a spektrumot periódikussá teszi. Az ábra felső részén az analóg jel spektruma látható, míg középen a Dirac- δ impulzus sorozat Fourier-transzformáltja, alul pedig a digitalizált jel periódikussá vált spektruma, amelyről egy jól megválasztott négyszögfüggvénnyel (szaggatott görbe) levághatjuk a spektrum ismétlődést, az aliaszokat

Értelmezzük a kapott eredményt. A kifejezés jobb oldala szerint a digitalizált jel spektruma periodikus, ami azért érdekes, mert aperiodikus függvények (vagy más néven tranziens függvények) spektruma nem periodikus függvény, vagyis a mintavételezés az eredetileg nem periodikus spektrumot periodikussá teszi (A.33. ábra).

A spektrumnak a $-1/2\tau$ és $1/2\tau$ eső részét a spektrum fő részének, az $f_N = 1/2\tau$ értéket Nyquist-frekvenciának nevezzük. A spektrum többi részén a fő rész f_N periódussal ismétlődik. A fenti formulákból világosan kiolvasható, hogy az analóg és a digitalizált függvény spektruma jelentősen eltérhet egymástól, ha az analóg függvény tetszőleges frekvenciájú jeleket is tartalmazhat. Ha azonban létezik egy olyan felső határfrekvencia (f_f), amelynél nagyobb frekvenciájú jel nem fordulhat elő (például valamely érzékelő eszköz átviteli képességei miatt), akkor belátható, hogy a felső határfrekvencia és a τ mintavételi távolsággal még átvihető legnagyobb frekvencia között igaz a következő összefüggés:

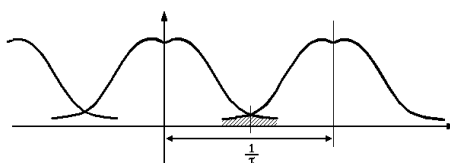
$$f_f \leq f_N$$

vagyis

$$\tau \leq \frac{1}{2f_f}$$

Ez az összefüggés a mintavételi tétel. Jelentése, hogy mintavételezéskor a még átvihető legnagyobb frekvenciához, f_f -hez úgy kell megválasztanunk a mintavételi távolságot (τ), hogy teljesüljön a mintavételi tétel.

Ha ennél kisebbre választjuk a mintavételi távolságot, akkor feleslegesen sűrűn mintavételezett adatrendszerrel kapunk, ha pedig ennél nagyobbra, akkor nem fog



A.34. ábra. A mintavételi távolság helytelen megválasztása következtében torzult a spektrum. A megismételt részek átnyúlnak a fő részbe, és azt módosítják

teljesülni az f_f felső határfrekvencia szerinti jelátvitel. Túlzottan ritka mintavételezéssel tehát felülvágást fogunk végrehajtani az adatrendszer spektrumán. Torzul továbbá a spektrum fő része is (A.34. ábra), mivel a megismételt részek „átnyúlnak” a fő részbe.

Az analóg jel visszaállítása

Láthattuk, hogy bizonyos esetekben a digitalizálás (mintavételezés) adatvesztéssel járhat. Tekintsük át, hogy mikor nem veszünk adatot. A megfelelően mintavételezett digitális adatrendszerből az eredeti analóg jel pontosan visszaállítható. (Akkor mondjuk megfelelően mintavételezettnek az adatrendszert, ha teljesült a mintavételi tétel.)

Ha pontosan vissza kívánjuk állítani az eredeti analóg jelet a mintavételezett jel spektrumából ($G_d(f)$), akkor el kell tüntetnünk a spektrum fő részein kívüli részeit, mivel a mintavétel periódikussá tette a tranziens függvény spektrumát, vagyis meg kell szoroznunk $G_d(f)$ -t egy olyan négyzetű függvénnyel (A.33. ábra), amelynek magassága τ , szélessége $1/\tau$. Így egyszerűen kiküszöböljük a mintavételezéssel belevitt periodicitást, vagyis visszakapjuk az analóg jel spektrumát. Az ismertetett gondolatmenet akkor adja vissza a jelet az idő tartományban, ha visszakapott spektrumot inverz Fourier-transzformáljuk.

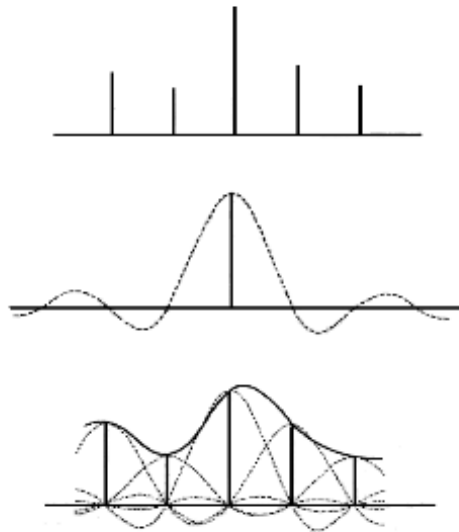
Idézzük fel a konvolúció tulajdonságait és a négyzetű-függvény (inverz) Fourier-transzformáltját. Belátható, hogy a τ magasságú és $1/\tau$ szélességű négyzetűfüggvénnyel való szorzás a frekvencia tartományban a *sinc* függvénnyel való konvolúciót jelent az időtartományban.

Jelölje $s(t)$ a négyzetűfüggvényt. Ekkor a spektrumisméltődést megszüntető levágásra írható, hogy

$$G_d(f)\tau \cdot s(f\tau) = G(f)$$

Így elértük, hogy az analóg és a digitális adatrendszer spektruma azonos lesz, amely egyben azt is jelenti, hogy a digitális adatrendszerből adatvesztés nélkül visszaállítható az analóg adatrendszer.

A mintavételezés azonban az időtartományban történik, ezért meg kell értenünk azt is, hogy az amúgy meglehetősen egyszerű művelet a frekvencia tartományban – mármint a spektrum fő részén kívüli tartomány levágása – mit je-



A.35. ábra. Az analóg jel helyreállítása a digitalizált jelből

lent az időtartományban. A digitalizált adatrendszer levágott spektrumának inverz Fourier-transzformációjával

$$\mathcal{F}^{-1}\{G_d(f)\} = \sum_{k=-\infty}^{\infty} g(k\tau)\delta(t - k\tau)$$

valamint annak figyelembe vételével, hogy

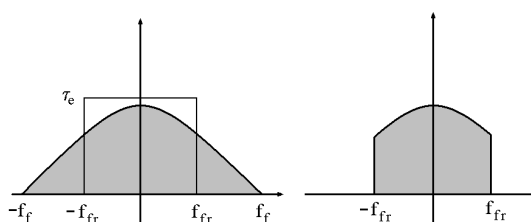
$$\mathcal{F}^{-1}\{\tau \cdot s(f\tau)\} = \text{sinc}(t/\tau - k)$$

az analóg jel visszaállítása az időtartományban a következő módon lehetséges:

$$\left(\sum_{k=-\infty}^{\infty} g(k\tau)\delta(t - k\tau) \right) \text{sinc}(t/\tau) = \sum_{k=-\infty}^{\infty} g(k\tau)\text{sinc}(t/\tau - k)$$

vagyis a visszaállított értékek a minták és a *sinc* függvény megfelelő argumentummal vett értékeinek szorzata. Ez azt jelenti, hogy az így visszakapott értékek pontosan azonosak az egyes minták értékeivel, valamint az analóg függvény tetszőleges helyén felvett értékek előállíthatók a mintáknak és a megfelelő argumentummal megadott *sinc* függvényértékek szorzatának összegzésével (A.35. ábra). Természetesen csak akkor igazak ezek a megállapítások, ha teljesült a mintavételi tétel.

Túl nagyra választott mintavételi távolsággal, azaz túl keskenyre választott csonkító függvénnyel (amivel a spektrum ismétlődést levágtuk) az analóg jel spektruma nem állítható vissza pontosan, mivel a túl ritka mintavétel felülvágást hajtott



A.36. ábra. A túl nagyra választott mintavételi távolság (azaz a túl kicsi felső határfrekvencia f_{fr}) hatására a spektrum torzul, és az eredetileg helyes felső határfrekvencia (f_f), és az új mintavételhez tartozó (f_{fr}) felső határfrekvenciája közötti spektrum tartomány eltűnik a jelből

vége a spektrumban (A.36. ábra). Előfordulhat, hogy ez a spektrum torzulás nem zavaró, mivel például ha digitális képre alkalmazzuk a ritkítást, akkor a megjelenítő eszköz felbontóképessége eleve megszab egy felső határfrekvenciát, amelynél sűrűbben mintavételezett adatot nem tud megjeleníteni.

Irodalomjegyzék

- [1] R. Allen – D. Mills: „Signal Analysis, Time, Frequency, Scale, and Structure”, Wiley, IEEE, 2004
- [2] D. Arctur – M. Zeiler: „Designing Geodatabases”, ESRI, 2004
- [3] Balogh Sándor: „Logikai elemek és kapcsolások; Gráfeméleti alapfogalmak”, 2004, <http://mek.oszk.hu/02900/02901/>
- [4] Bronstejn – Szemengyajev – Musiol – Mühlig: „Matematikai kézikönyv”, TypoTEX Kiadó, 2002
- [5] R. Burke: „Gettin to know ArcObjects”, ESRI Press, 2003
- [6] T. Cormen – C. Leiserson – R. Rivest – C. Stein: „Új algoritmusok”, Scolar kiadó, 2003
- [7] Czímber K.: Geoinformatika, Elektronikus jegyzet, 2001, <http://www.geo.u-szeged.hu/joe/fotogrammetria/GeoInfo/index.htm>
- [8] Elek I.: „Bevezetés a geoinformatikába”, ELTE Eötvös kiadó, 2006
- [9] Elek I.: „Térinformatikai gyakorlatok”, ELTE Eötvös kiadó, 2007
- [10] I. Elek: „Automatic Generalization of Maps by Digital Filtering”, Geomatics, 2007
- [11] Elek I. – Dezső B. – Máriás Zs.: „IKKK kutatási jelentés”, 2006
- [12] I. Elek: „Data Mining Methods in Vectorbased GIS”, Nova Acta Leopoldina, Neue Folge, Band 94, Nummer 349, Thematische Kartierungen in den Geowissenschaften, 2007
- [13] ESRI Shapefile Technical Description – ESRI White Paper, July 1998
- [14] D. Arctur – M. Zeiler: „Designing Geodatabases”, ESRI, 2004
- [15] Gonzalez – Woods: „Image processing”, Prentice Hall, 2006
- [16] Hajnal P.: „Gráfelmélet”, Poligon kiadó, 2003

- [17] Ivanyi A. (szerk.): „Informatikai algoritmusok 1-2”, ELTE Eötvös Kiadó, 2005
- [18] Jánossy L. – Tasnádi P.: „Vektorszámítás I., Vektor- és tenzoralkgebra”, Nemzeti Tankönyvkiadó, 2002
- [19] Klinghammer – Papp-Váry: „Földünk tükre a térkép”, Gondolat, 1983
- [20] Kuros: „Felsőbb algebra”, Tankönyvkiadó, 1975
- [21] Lovász L. – Pelikán J. – Vesztergombi K.: „Diszkrét matematika”, TypoTEX kiadó, 2006
- [22] MacEachren – Taylor: „Visualization in Modern Cartography”, Elsevier, 2005
- [23] A. Mesko: „Digital Filtering”, John Wiley & Sons, 1984
- [24] A. Mitchell: „The ESRI Guide to GIS Analysis”, ESRI, 1999
- [25] D. Mix – K. Olejniczak: „Elements of Wavelets for Engineers and Scientists”, Wiley, 2003
- [26] Móri T. – Székely G.: „Többváltozós statisztikai analízis”, Műszaki Könyvkiadó, 1986
- [27] M.E. Mortenson: „Mathematics for Computer Graphics Applications”, Industrial Press, 1999
- [28] Mucsi L.: „Műholdas távérzékelés”, Libellus, 2004
- [29] M. Petrou – P.G. Sevilla: „Dealing with Texture”, Wiley, 2006
- [30] J. F. Richards: „Remote sensing Digital image analysis”, Springer-Verlag, 1986
- [31] P. Rigaux – M. Scholl – A. Voisard: „Spatial Databases with Application to GIS”, Morgan Kaufmann Publishers, 2002
- [32] S. Smith: „Digital Signal Processing”, Elsevier Science, 2003
- [33] E. Stein: „Fourier analysis: an introduction”, Wiley, 2003
- [34] E. Stein – R. Shakarchi „Complex analysis”, Princeton University Press, 2003
- [35] Szirmay-Kalos L. – Antal Gy. – Csonka F.: „Háromdimenziós grafika, animációs és játékfejlesztés”, Computerbooks, 2003
- [36] Scharnitzky V.: „Mátrixszámítás”, Műszaki könyvkiadó, 1970

- [37] G. Thomas – M. Weir – J. Hass – F. Giordano: „Thomas-féle Kalkulus 1-2-3”
TypoTEX kiadó, 2006-2007
- [38] R. Tomlinson: „Thinking About GIS”, ESRI Press, 2007
- [39] Laurini – Thomson: „Fundamentals of Spatial Information Systems”, Academic Press, 1992