SZINTVONALAK VEKTORIZÁLÁSA

SZAKDOLGOZAT

FÖLDTUDOMÁNYI ALAPSZAK

TÉRKÉPÉSZ SZAKIRÁNY

Készítette: Kocsis Balázs

Témavezető: Zentai László

Eötvös Loránd Tudományegyetem

Térképtudományi és Geoinformatikai Tanszék

Budapest, 2016

Tartalomjegyzék

1. Bevezetés

A XX. század második felében a számítástechnika ugrásszerű fejlődése a tudomány és a mindennapi élet minden területére rányomta bélyegét. A személyi számítógépek és az internet elterjedése következtében ugrásszerűen nőtt a kereslet a digitálisan elérhető tartalmak iránt. A „digitális forradalom" így természetesen a térképészet számára is alapvető változásokat hozott. A számítástechnika eredményei révén már az 1960-as években lehetővé váltak az első térinformatikai szoftverek (GIS) megjelenései.

A GIS (*Geographic/Geospatial Information System*) célja olyan adatok tárolása, kezelése, elemzése és megjelenítése, amik valamilyen térbeli vonatkozással rendelkeznek – tehát vannak koordinátái. Az első ilyen rendszert Roger Tomlinson alkotta meg a '60-as években (Geo-Information System of The Canada Land Inventory). A GIS szoftverek hasznát hamarosan felismerte az ipar is, és így a fejlesztés üteme meghatványozódott.

Magyarországon már az 1970-es évektől történtek szárnypróbálgatások a térinformatika területén, de igazán csak az 1980-as évek közepétől van jelen a digitális térképészet. Az Államigazgatási Számítógépes Szolgálat, a Magyar Honvédség Térképészeti Intézete (ekkor Tóth Ágoston Térképészeti Intézet), a Geometria (kisszövetkezet, majd kft.), és különböző állami kutatóintézetek berkeiben csupán néhány év eltéréssel jelennek meg digitális térképek. A magánszféra térképészeti vállalatainak számítógépes előretörése a rendszerváltás utáni időszakra tehető.

A szoftverek fejlesztésével együtt fokozatosan módosult a térképészek tevékenysége is: a térképrajzolásba egyre nagyobb mértékben vonták bele a komputereket. Napjaink térképei szinte kivétel nélkül számítógépen készülnek, sőt a térképek digitális megjelenítése is egyre nagyobb szerephez jut. A modern digitális térképek praktikusak (hiszen adott esetben kis helyen is elférnek), egyszerűen változtathatók, könnyen kezelhetők, és interaktívak – így szolgálva a ma emberének fokozott igényeit.

A világ szinte minden területén az ezredfordulóig lezajlott az elsődleges fontosságúnak ítélt topográfiai térképek digitalizálása, de máig akadnak olyan térképek, amiket valamilyen célból digitalizálni kell.

Ez a dolgozat a topográfiai térképek szintvonalainak vektorizálási lehetőségeivel foglalkozik, különös tekintettel az automatikus vektorizálás által kínált lehetőségekre és korlátozásokra.

2. Vektorizálás

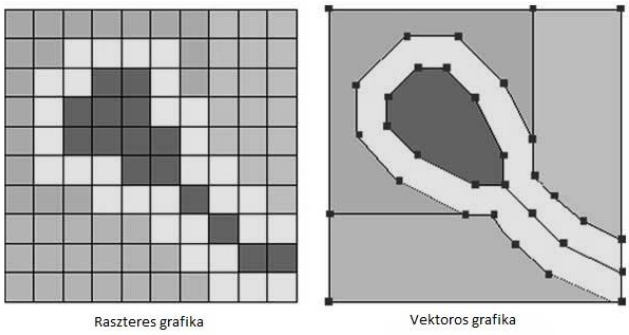
Pontosan mit értünk a térképek vektorizálásán? Többet jelent-e ez a szókapcsolat, mint a szavak értelmezése külön-külön? A vektorizálás egyik definíciója szerint olyan folyamat, melynek során raszteres képből vektorosat hozunk létre.

A raszteres képnek nincs belső szerkezete, csupán sorokba és oszlopokba rendezett pontok halmaza. Minden pontnak ismert a pozíciója, és valamilyen egyéb attribútuma. A számítógép minden egyes képpont (pixel) esetén tárolja annak elhelyezkedését: vízszintes és függőleges koordinátapár, általában a bal felső sarokból indítva, jobbra és lefelé növekvő értékekkel.

Mindezek mellett a számítógép megőriz még valamit: az adott képpont „színét”. Legegyszerűbb esetben ez csak egy bitnyi információ: 0 vagy 1 – üres, vagy kitöltött az adott pixel. Színes kép esetén ezen attribútum egy hosszabb lánc, amelyből a kódolás ismeretében kinyerhető a szín.

Egy ilyen képnek több előnye van:

- könnyen előállítható és megjeleníthető – megfelelő digitális kamerák és szkennerek kimeneti formátumának
- szín- és részletgazdag képek keletkezhetnek



1. ábra

Nagyításra ugyanakkor nem jól reagál – a sarkok és élek töredezni kezdenek, a kontúrok elmosódnak, az objektumokat nem lehet pontosan lehatárolni. Nagy méretű és felbontású kép esetén pedig nagyon megnőhet a fájl mérete is. Ezen problémák kiküszöbölésére jött létre a vektoros grafika.

2.1. Vektoros grafika

Vektoros kép esetén – a raszteressel ellentétben nem az összes, csak jóval kevesebb, matematikai algoritmusok által kijelölt pontnak őrződik meg a helye. A kép digitális kódja csupán e néhány pont pozíciójából, és a pontok közötti kapcsolatokból áll. A kapcsolatok (relációk) alapján osztályozhatjuk a képen megjelenő objektumokat:

- Lehetnek pontszerű objektumok – függetlenül álló pontok, melyeknél (a raszteres képhez hasonlóan) a számítógép rögzíti helyüket és attribútumaikat.
- Vonalas objektumok esetén ismert a kezdő- és végpont, a sarokpontok, illetve (programtól függően) az esetleges görbe meghatározott pontjai. A számítógép az attribútumokat nem pontról-pontra egyenként, hanem vonalanként tárolja
- Poligonról beszélünk, ha a rendszer egy önmagába záródó vonal elemeit, és ennek kitöltését rögzíti. Az attribútumok itt is a teljes objektumra vonatkoznak.

Programtól függően a felhasználó további helyet takaríthat meg akkor, ha a hasonló jellemzőkkel bíró objektumokat attribútumaik alapján is csoportokba rendezi, és így tárolja.

A vektoros ábrázolás egyik hátránya, hogy az automatikus átalakítás raszter és vektor között igen bonyolult, és sok lehetőséget hagy a hibára. A teljesen automatikus konverzió így sok esetben nem is lehetséges, vagy nem célravezető. Ez a helyzet a térképek átalakításával is.

3. Térképek vektorizálása

A térképek vektorizálása több, mint egyszerű vektorizálás. Itt nem egy szimpla konverzióról van szó – azt bármilyen szoftver több-kevesebb sikerrel végrehajtaná.

A térkép abban különbözik egy hétköznapi képtől, hogy azon minden egyes objektumnak meghatározott jelentősége van. Körök vagy pontok egymáshoz viszonyított helyzete (és esetleg száma) ugyanúgy jelentőséget hordoz, mint egy vonalban lévő szakadás. Ahhoz, hogy a térképet teljességében értelmezni tudjuk, és vektoros formában megjeleníthessük, szerkezetét mélységében kell felmérni, *interpretálni*. A térkép-interpretációt általánosan a *dokumentum-elemzés* tárgykörébe sorolják. Ennek több alterülete is kapcsolódik a térképek elemzéséhez, például a műszaki rajzok elemzése.

A műszaki rajz sok szempontból hasonlít egy térképre, hiszen többnyire van kerete és címe, illetve minden rajzi elemnek jelentősége van, amit fel kell tüntetni (jelmagyarázat). A jelek és a rajzolt elemek ugyanakkor szakterület-specifikusak, így egy műszaki rajzot interpretáló programot másra nem is lehetne használni. A szoftver alapelvei viszont megfelelnek az automatikus térkép-interpretáció igényeinek. Ennek része többek között:

- Optical Character Recognition – OCR: nyomtatott (és kézzel írt) karakterek felismerése és
- Form analysis: nyomtatványok feldolgozása, tágabb értelemben a vonal és szöveg szétválasztása.

Egy dokumentumot azonban nem csak automatikusan lehet vektorizálni – egy térképet pedig főleg nem. Maga a név – automatikus vektorizálás – is megtévesztő: bár a szoftver elméletileg önállóan képzi le a vektoros rajzot, futtatás előtt számos paramétert az operátornak kell beállítania, majd utána manuálisan helyesbítenie. A szintvonalakkal kapcsolatban felmerülő problémákkal a későbbiekben részletesen is foglalkozom.

Félautomatikus vektorizálásról akkor beszélünk, ha a szoftver a képernyőn automatikusan követi a térkép vonalait, és képezi azt le vektorokra, de minden kereszteződésnél megáll, és a felhasználóra hárul a feladat, hogy eldöntse, merre

tovább. Ez az eljárás főleg akkor előnyös, ha csak egy térképrészletet kell digitalizálni. A késztermék itt is rendszerint utólagos korrekcióra szorul.

3.1. Manuális vektorizálás

Manuális vektorizálás esetén a vektoros rajzot teljes egészében az operátor állítja elő. Az elérhető eszközök alapján itt két fő módszert különíthetünk el: történhet *digitalizáló tábla* segítségével, vagy úgynevezett *heads-up vektorizálással.*

- Digitalizáló tábla használata esetén az operátornak felváltva kell néznie a térképet, és rajzolnia azt a táblára, illetve ellenőriznie a felvitt adatokat a monitoron. A vektoros rajz pontossága csak a felhasználó kézügyességén múlik. Ez volt az első, szélesebb körben elterjedt digitalizációs forma – nyugaton már az 1960-as évektől kezdve elérhető áron kínálták a digitalizáló táblákat (pl. a RAND tabletet).
- A Heads-up vektorizálás egy máig népszerű módszer: a szkenner segítségével nyert raszteres képet a képernyőn manuálisan vektorizáljuk. Ez az eljárás egy lépésben megoldja a digitalizáló tábla hibáit; mivel a nyersanyagot és a vektoros rajzot már a kezdetektől egymásra vetítve látjuk, nem kell kettő (vagy három) dolog között megosztani a figyelmünket, és a pontosság is azonnal ellenőrizhető. Digitális térkép elkészítése heads-up vektorizálás használatával hozzávetőlegesen fele annyi időbe telik, mint digitalizáló táblával. Egyetlen hátránya a módszernek az, hogy nagyobb méretű dokumentumok szkenneléséhez a felszerelés nagyon költséges, így legtöbbször egy erre szakosodott cég bevonása szükséges.

3.2. Automatikus vektorizálás – a vektorizáló algoritmusok

Egy kész vektoros rajz kialakításának folyamatában az első lépés az elemi vonalak felismerése és vektorizálása. Az, hogy itt az adott szoftver milyen algo–ritmus szerint működik, életbevágóan fontos a későbbi lépések szempontjából. Annak érdekében, hogy a későbbi folyamatok megbízhatóan működjenek, elengedhetetlen a pontosságra való törekvés.

Bár az algoritmusok más-más módszerrel dolgoznak, „ütemtervük” általában hasonló:

1. Meghatározzák a vonal középtengelyét – egy pixelnyi szélességű „csontvázat” jelölnek ki. Több módszer létezik, amivel a vonalat „csupaszítják” – az egyszerűbb vékonyítástól kezdve, a rendkívül bonyolult egyenletrendszerekig.
2. Összekötik ezekek a pixeleket, elemi vektorok láncát hozva létre
3. Felismerik a töréspontokat, és ezek felhasználásával megalkotják a kész vektorokat.

A következőkben néhány ilyen eljárást részletesen is ismertetni fogok.

3.2.1. Vékonyításon alapuló módszerek

Ezek olyan módszerek, amik teljesen az említett algoritmust követve dolgoznak. Nagy előnyük, hogy minden esetben ragaszkodnak a vonal folytonosságához. Fő hátrányuk, hogy minden eredeti vonalvastagsági információ elveszik (a vékonyítás természetéből adódóan), és a kereszteződéseknél torzít (a vonalvastagság a kereszteződésben aszimmetrikusan változik, a vonalat követve a középvonalban is egy kis bevágódás lesz megfigyelhető a csatlakozó vonal irányában).

Első lépésként – a vékonyítás során – általában olyan megoldást alkalmaznak, amelyben a program a pixelek egyidejű, párhuzamos törlésével „hámozza” a vonal mindkét oldalát. A pixelnyi szélességű vonalvázon aztán már egyszerű felismerni a vég- és az elágazási pontokat. Amennyiben egy pontnak csak egy (értékkel bíró) szomszédja van, az végpont. Ha legalább három, akkor elágazás. Ez azonban még

nem elég a vektoros rajzhoz. A vonalak belső pontjainak (amiknek kettő szomszédjuk van) töréspontjait is meg kell találni, és ebben segít a *poligonizálás*:

- Az egyik legrégibb (1973) de máig legnépszerűbb (pl. ArcInfo, Autodesk Map) a Douglas-Peucker algoritmus. Ez első lépésként összeköti a görbe két végpontját, majd megkeresi az ettől az egyenestől legtávolabbi pontot a görbén. Majd az egyik végpontot és a legutóbb megtalált legtávolabbi pontot köti össze, majd addig ismételgeti az előző lépéseket, amíg az egyenestől legtávolabbi pont is egy adott határértéken belül nem lesz. Ezután lép tovább a következő szakaszra. Kiemelkedően fontos a helyes küszöbérték meghatározása – hozzávetőleg akkorának kell lennie, mint a legkisebb térképi vonalvastagság.
- K. Wall és P. E. Danielsson 1984-ben kitalált módszere ehhez nagyon hasonlóan dolgozik, de a távolság helyett az eredeti görbe és az újonnan megrajzolt egyenes közötti terület vonalegységre eső értékét figyeli. Az eljárás hátránya, hogy a görbe hirtelen változásait (pl. sarokpontok) csak bizonyos késéssel képes detektálni, így azok a kész rajzon eredeti helyükhöz képest elcsúszhatnak.
- Kumar S. Ray és Bimal Kumar Ray metódusa (1992) szerint a kész vektoros rajz majdani sarokpontjait az eredeti görbe pillanatnyi görbülete segítségével határozzák meg – egy előre beállított alsó és felső határérték által behatárolva.

Az első automatikus vektorizációs módszerek is vékonyításon alapultak, talán azért, mert ez az egyik legegyszerűbb metódus (ugyanakkor az egyik leglassabb is). Bizonyos modernebb kutatások néha ismét visszatérnek a vékonyításon alapuló módszerekhez. (#1)

[1] O. Hori és S. Tanigawa (1993): Raster-to-Vector Conversion by Line Fitting Based on Contours and Skeletons. Tsukuba, Japán

3.2.2. Kontúron alapuló módszerek

A vektorizálás kezdeti korszakában a kísérletezőbb kedvű programozók körében felmerült az ötlet, hogy az objektumot annak körvonalai alapján redukálják vektorokká. A kontúron alapuló módszerek első lépésként kijelölik az objektum körvonalait, majd ezek alapján egy középtengelyt számítanak ki. Nagy hátrányuk, hogy egymáshoz közeli objektumok esetén könnyen csak egyet látnak meg, illetve kereszteződéseknél az egyik vonalat figyelmen kívül hagyhatják (bár ez utóbbi jelentősége szintvonalak vektorizálása esetén kicsiny). Előnyük a vonalvastagság megőrzése, ami a későbbiek során hasznos lehet (alapszintvonalak és főszint-vonalak elkülönítésére).

3.2.3. Futam-gráf módszer

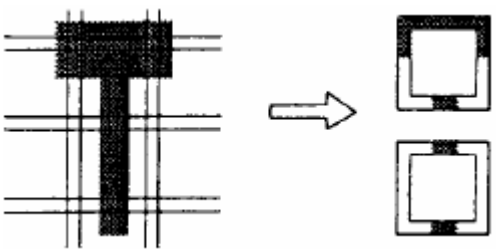
Az eljárást Di Zenzo & Morelli irodája dolgozta ki 1989 folyamán, majd L. Boatto és társai alkalmazták három évvel később az olasz kataszteri térképek feldolgozása során. A *futam* (run) a vonalnak egy bizonyos irányú metszete, ami négy koordinátával jellemezhető:

- A futam iránya (0 ha vízszintes, 1 ha függőleges)
- A futam koordinátája az irány tengelyére merőlegesen (a másik tengely)
- Kezdőpont koordinátája
- Objektum végének koordinátája

Tehát bármilyen kép leírható futamok halmazaként.

A módszer előnye, hogy megőrzi az eredeti vonalvastagságot, míg hátránya az, hogy irányváltásnál téves elágazást érzékelhet és emiatt elméletileg (Wenyin és Dori szerint) nem alkalmas görbe vonalak elemzésére.

3.2.4. Mesh pattern (mozaik) módszer



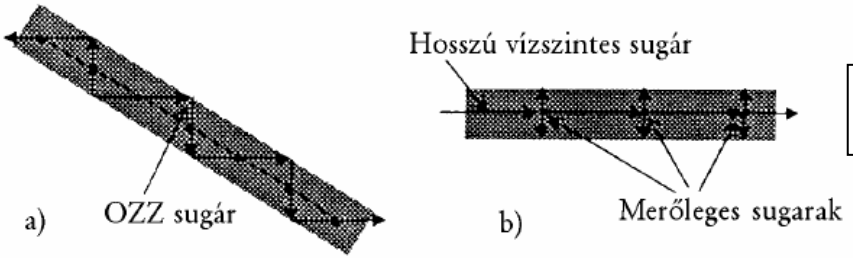
2. ábra – A mesh pattern-módszer szemléltetése

Lin és társai 1985-ben főként diagramok feldolgozására hozták létre ezt a módszert. Az objektumokat négyzetráccsal fedi le, és ez a négyzetek körvonalaira fókuszál. Egy mozaik adatbázissal összeveti a kapott hálót, és így azonosítja a jellemző objektumokat. Az eredeti adatbázis csupán 51 mintát tartalmazott, de ez természetesen szabadon bővíthető – alkalmazási területtől függően.

Legfontosabb paraméter a négyzetrács méretének meghatározása. Optimális esetben a rácsméret valamivel nagyobb, mint a maximális vonalvastagság, ugyanakkor kisebb, mint a minimális vonaltávolság. Gyakorlatban ez többnyire nem megvalósítható, így pl. pontozott vonalak könnyen megváltozhatnak és eltűnhetnek a rajzokon. Az eljárás előnye, hogy nagyon gyors, hiszen a négyzetrács miatt az összes képpontnak csak egy nagyon kis részét vizsgálja már az első lépéstől kezdve.

Műszaki rajzokra Vaxiviese és Tombre optimalizálták 1992-ben, főként úgy, hogy lecsökkentették az eredeti rácsméretet és ezzel párhuzamosan kibővítették az adatbázist. Szintvonalak vektorizálására ez az eddig ismertetett módszereknél kevésbé alkalmas.

3.2.5. Merőleges cikk-cakk módszer (OZZ)



3. ábra – Az OZZ vektorizálás alapelvei

Wenyin-Dori dolgozta ki 1996-ban. A módszer alapvetően különbözik minden egyéb nem–vékonyításon alapuló metódustól. Egy pixel vastagságú egyenes „fénysugarat” indít a program valahol az eredeti vonal belsejében és ez a sugár egyenesen megy az objektum faláig, ahol merőlegesen elkanyarodva mintegy visszaverődik, hogy másik irányban folytassa útját. Az így alkotott vízszintes és függőleges szakaszok felező pontjai lesznek a – végtermék vektor – sarokpontjai. A módszer megalkotói nem bíztak semmit a véletlenre: ha egy megszabott küszöbértéken túl sem talál szegélyt a fénysugár (tehát egy hosszú vízszintes, vagy függőleges vonal belsejében halad), automatikusan megszakad, és merőleges

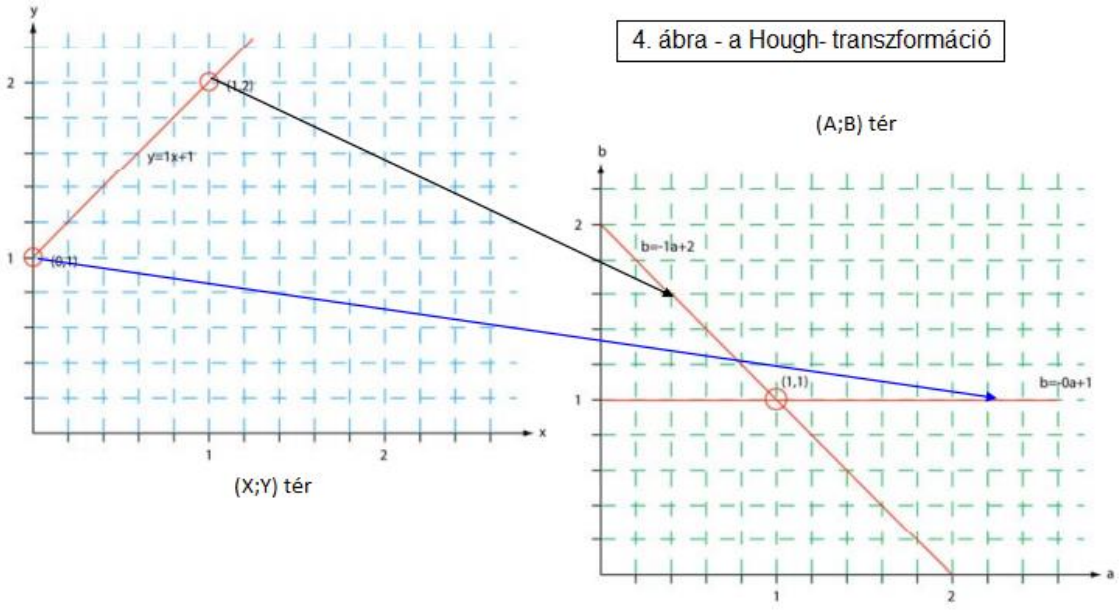
sugarak indulnak a helyén azért, hogy mindenképp legyenek belső pontjai egy egyenesnek. Előnyei:

- Időhatékony, mivel az összes pixelnek (a teljes képnek) csak egy kis részét vizsgálja: azt, amelyiken a fénysugár áthalad.
- A módszer emellett jól kezeli a kereszteződéseket és csomópontokat is (bár ennek jó esetben nincs jelentősége a szintvonalak vektorizálása során)
- Az íveket is helyesen, ívként detektálja.

Hátrányai:

- Bonyolult paraméterezés – 12 értéket kell manuálisan állítani.
- Egymáshoz közeli íveket hibásan ismer fel – szomszédos vonalak gyakran ütköznek, vagy összefolynak – ez főleg akkor probléma, ha eltérő vonalvastagságú elemeket kellene egy lépésben felismerni.

3.2.6. Hough-transzformáción alapuló módszer



4. ábra - a Hough- transzformáció

A módszer célja, hogy tökéletes mértani idomokat találjon és kijelöljön egy képen. Eredetileg (1962-ben) Paul Hough módszere még csak szakadozott egyenesek kiszűrésére volt képes, de tíz évre rá már köröket és ellipsziseket is felismert Hough követőinek munkája révén. A transzformáció alapja a Descartes-i koordináta rendszer alkotta normáltér és (modernebb formában már polár-koordinátákat tartalmazó) paramétertér kapcsolata.

Hough eredeti ötlete azon a tételen alapul, miszerint egy derékszögű koordinátarendszerben minden egyenes felírható az alábbi egyenlettel: Y=MX+B, ez pedig egy pont (M,B) a paramétertérben. Tehát a paramétertér egy egyenese felírható B=-XM+Y formában is. Eszerint tehát, bármelyik térben egy pont a másikban egy egyenessel egyezik meg. A normáltér két pontja két egymást metsző egyenest ad a paramétertérben. A két egyenes metszéspontja pedig megadja a normáltér eredeti két pontját összekötő egyenes paramétereit. Egy vonalban lévő pontok egyenesei a paramétertérben tehát egy pontban metszik egymást, csomópontot alkotnak.

A módszer a paraméterteret ezután egységnyi cellákra bontja és megszámolja, hogy hány egyenes metszéspontja esik az egyes cellákra – tehát a normáltér hány pontja esik az adott cella által paraméterezett egyenesbe. Ha ez a szám egy előre meghatározott határértéket meghalad (vagy lokális maximum), már meg is vannak egy, a normáltérben létező egyenes paraméterei.

A gyakorlatban a paramétertér koordinátáit inkább a normáltér egyeneseinek polárkoordinátájaként adjuk meg: X*cos(-)+Y*sin(-)=R, mert ez a módszer számítógépek számára jobban értelmezhető. (A függőleges egyeneseknek egyértelműen definiálhatóak a paraméterei.) Mivel a kapott egyenesek hosszáról semmilyen információval nem szolgál, a kapott egyenest újra össze kell vetni az eredeti képpel. A módszer előnyei:

- Viszonylag egyszerű koncepció és megvalósítás
- Nagyon jól kezeli a hiányos, szakadozott mértani formákat
- Több alak felismerésére átállítható, nem csak egyenesére (bár egyszerre csak egyfélét képes érzékelni)

Hátrányai:

- A több paraméterrel felírható formák esetén a kód nagyon bonyolulttá válik
- Gyakran „képzelődik” – téves vonalakat lát, ez beállítások függvénye
- Nem tartalmaz a vonal hosszára vonatkozó információt
- Azonos egyenesbe eső, de különböző vonalak nem szétválaszthatóak
- Pontatlan lehet jelentős lokális torzulásokkal, hibákkal

4. Az automatikus vektorizálás nehézségei

Tételezzük fel, hogy célunk egy olyan program létrehozása, amely képes bármely betáplált raszteres térképből szinte teljesen emberi beavatkozás nélkül létrehozni egy kész vektoros szintvonalas térképet. Ilyen (teljesen automatikus) program jelenleg nem létezik, de vajon mai ismereteinkkel lehetséges-e elkészítése?

Egyszerű válasz van erre a kérdésre: nem. A világ, sőt akár egy nemzet topográfiai térképei annyira sokszínűek, hogy ez nem lehetséges. A térképkészítő számára a sokféle méretarány, a jórészt univerzális szabályozás nélküli jelkulcs-rendszer (jelhasználat, és a térkép felhasználásának későbbi célja) bőven hagy annyi mozgásteret, hogy térképét egyedivé tehesse. Egy sablonokban és csoportokban „gondolkozó” számítógép számára ez áthidalhatatlan problémát jelent.

Redukáljuk tehát a programunkkal szemben támasztott követelményeket – egy adott méretarányban a generalizálás szintje hozzávetőleg állandó, így drasztikusan csökken az alaprajzszerű ábrázolások aránya – ez pedig csökkenti a feladat nehézségét. Ez a csökkentés azonban még mindig nem elég.

4.1. Kulturális különbségek

Földünk két különböző országában eltérő kultúrájú emberek ugyanazon dologra sok esetben más jelet használnak, nem is beszélve a térkép nyelvének és írásának (sőt számírásának) esetleges eltéréseiről. Ezenfelül szükséges a térkép tematikája / célközönsége szerint is osztályozni, mert bár a turista, katonai, kataszteri vagy esetleg tájfutó térképek jelkulcsa legalább országos szinten szabályozott, ezek egymás között általában nem kompatibilisek.

Ilyen szigorú szabályok közé szorítva már lehetséges egy program, programcsomag segítségével digitalizálni térképeket és ezt az információ-technológia fejlődésével a legtöbb ország meg is tette legalább a kataszteri térképek szintjén, noha eltérő fokban automatizálva. Egy-egy ilyen program kifejlesztése (és használata) sok munkaórát emészt fel és nagy befektetést igényel, ezért csak tőkeerős vállalatok engedhették meg maguknak. Kisebb cégek vagy kisebb

térképsorozatok számára célravezetőbb volt teljesen új térképet csinálni digitális formában, vagy manuálisan digitalizálni már meglevő térképeket.

A térképi jelek mellett még legalább egy dolog van, ami minden topográfiai térképen biztosan szerepel, és ez a szintvonal. Ha az előbbiekben felvázolt képzelt programunknak csak a szintvonalakat kellene automatikusan leolvasnia és felismernie, biztosan könnyebb dolga lenne – vagy mégsem? Milyen problémák adódhatnak szintvonalas térkép gépi értelmezésekor?

Bár definíciójuk szerint „a szintvonalak önmagukba visszatérő, egymást sohasem keresztező görbék”,[(#2)] ez a térkép készítése során nem mindig keresztülvihető. A szintvonalak megszakadhatnak akár egy megírás vagy térképi jel miatt, amiket nagyobb fontosságúnak ítéltek, egyes térképeken pedig magának a szintvonalnak a megírása szakíthatja meg, így okozva problémát a számítógépnek, ugyanis rendszertelen folytonossági problémák esetén kevés vonalfelismerő módszer használható. Egy gép számára akár egy eséstüske, vagy egy azonos színű érintkező objektum is rendkívül zavaró lehet, megnehezítvén az eredeti vonal felismerését. Nagy szintkülönbségű területeken (a szintvonalak szinte érintik egymást) is gondok lehetnek gépi beolvasásukkal. Egyéb természeti objektumok (például egy szurdok) térképi jelébe is belefutnak, majd megszakadnak a szintvonalak.

Természetesen a szintvonalak esetében is fennállnak a korábbiakban ismertetett eltérő kultúrákból adódó nehézségek. Bár Európában ilyet meglehetősen ritkán látni, de a Dél- és Délkelet-Ázsiában található volt gyarmati országok függetlenedésük után gyakran adtak ki kizárólag helyi nyelvű – tehát nem arab számírású és nem latin betűs térképsorozatokat. Fontos ezzel a látszólagos mellékvágánnyal is törődni, mert ezek a térképek többnyire közvetlen azt a kritikus időszakot megelőzően születtek, amikor fokozott igény mutatkozott szintvonalakat vektorizáló programokra.

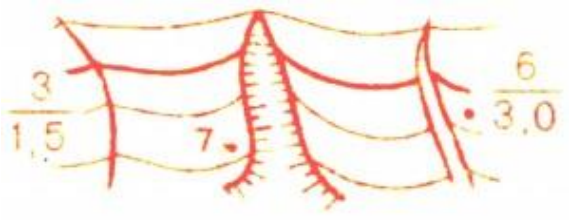
[2] Definíció Klinghammer István és Papp-Váry Árpád: Földünk tükre a térkép c. könyvéből

A II. világháború – vagy esetenként a függetlenségi harcok lezárultával megnőtt az igény új, a háborúk okozta változásokat dokumentáló, de egyben a nemzetek szuverenitását is hangsúlyozó (nemzeti nyelvű) térképek elkészítésére. A világ majdnem minden részén ekkor még manuálisan, minimális számítógépes segédlettel készültek papírtérképek. Az azóta eltelt 40–50 év során kevés helyen történt akkora változás, ami teljesen új térképsorozatok készítését tette volna szükségessé. A technológia fejlődésével egyre növekvő igény mutatkozik a digitális térképek készítésére, és amíg műholdas adatokból interpolált (kellő pontosságú) digitális térképekhez nincs mindenkinek elérhető áron hozzáférése, reális kereslet mutatkozik egy automatikusan digitalizáló szoftverre.

Nyilvánvaló az a tény, hogy ilyen térképek esetén az elemzés megkezdése előtt operátori beavatkozás szükséges a számítógép által észlelt betűkészlet átállításához. Ezt a folyamatot automatizálni egyszerűen nem éri meg.

Latin betűs és arab számos térképeken sem egyértelmű azonban, milyen mértékegység szerint történt a térkép számozása. Ma a világ túlnyomó részén ugyan a metrikus rendszer az irányadó, de többek között az (egyik legjelentősebb térképkibocsájtó) Amerikai Egyesült Államok is részben a brit mérték-egységrendszert (Imperial Unit System) használja. Több volt angol gyarmati ország (például Kanada) is csak a ’70-es években tért át az SI-re, azelőtt a topográfiai térképek szintvonalértékeit lábakban (feet) számolták. Mértékegységtől függetlenül természetesen bármely térkép digitális formában arányosan megjeleníthető és olvasható, ugyanakkor a mértékegységek ismerete nélkül a gyakorlatban használhatatlan, és másik térképpel össze nem vethető.

A gyakorlatban létező szinte minden program esetén a mértékegység beállítása is az operátor feladata.



5. ábra – Metsződés, horhos és vízmosás jelkulcsi ábrázolása

4.2. Térképi irregularitások

A térképen jelzett terepi jelenségek hasonlóképpen megzavarhatják a számítógépet, szintvonal ugyanis elméletben nem szakadhat meg (amíg a térképlap szélére nem ér, vagy önmagába vissza nem fordul), a gyakorlat azonban mást mutat. Hegységek és dombvidékek térképeit nézve gyakran találkozhatunk egyes domborzati tereptárgyakkal. Egy suvadás, horhos vagy egy tereplépcső jele önmagában is megszakítja vagy eltolja a szintvonalakat. Ez az emberi térképolvasó számára a térképet könnyebben értelmezhetővé teszi, az intuíciót támogatja, de egy szabott paraméterekkel üzemelő program számára nehezen áthidalható kihívást jelent.

Itt érdemel említést a nyergek, kúpok és hasonló részletidomok problémája is. Ha a szintvonalak digitális térképen való rögzítésének a fő célját nézzük, az mindenképpen egy domborzati modell megalkotása, így fontos volna, hogy a program ezeket automatikusan azonosítsa a digitalizálás során

Nyerget a térképen sokszor felező- (vagy negyedelő) szintvonalakkal ábrázolunk, és általában a nyeregtető magasságára ezek elhelyezkedése alapján lehet következtetni. Egy szurdok vagy horhos térképi jele többnyire ugyanolyan színű mint a szintvonal, így nehezítvén a számítógépnek az elkülönítést, emellett vonalába valódi szintvonalak csatlakoznak bele. Különbség általában csupán a vonal vastagságában (például tereplépcső esetén), vagy „díszítésében” (egy horhos „fogazása”) lelhető fel, és abban a tényben, hogy következetesen figyelmen kívül hagyja a szintvonalakra vonatkozó szabályokat. Ez az a pont, ahol a leginkább meg lehet értetni a különbözőséget egy számítógéppel: ha feltételezhetjük, hogy a program nem olvas be hibás adatokat, és az eredeti térkép is hibátlan, akkor feltehetjük, hogy a szoftver által hibaként érzékelt minden objektum egy domborzati idom. Sajnos általában nem ez a helyzet.

Egy átlagos topográfiai térkép szintvonalai azonban nem kizárólag alapszintvonalakból állnak.

Legtöbbször minden ötödik szintvonalat a többinél vastagabb vonal jelöl, ez a főszintvonal. Míg ez az emberi szem számára az olvashatóságot könnyíti, addig egy számítógépnek komoly gondot okozhat. Több vonalfelismerő technika

egyszerre csak egyféle vastagságú vonalat képes „látni” kellő pontossággal. Ettől eltérő vastagságú vonal felismerése a program más paraméterekkel történő ismételt futtatását igényli. A paraméterek növelésével párhuzamosan emelkedhet a hibák száma is.

Kis szintkülönbségű, de nagyobb kiterjedésű területeken – vagy ahol a térkép készítője pontosabban kívánja érzékeltetni a terep meredekségét – felező, sőt negyedelő szintvonalakat találunk. Ezek többnyire a „semmiből” indulnak, és dolguk végeztével egyszerűen eltűnnek – a teljes izohipszának csak egy szeletét mutatják. Mindig az alapszintvonalakkal megegyező színű, de szaggatott vonallal jelzik őket. Magának a felezőszintvonalnak a felismerése is nagy terhet ró a gépre, hiszen nem szokott rajta lenni megírás, de a vonal szaggatott mivolta még nagyobb kihívást jelent – az a tény, hogy egy folytonossági hiány után ugyanaz a vonal folytatódik, meghaladja a legtöbb általános célokra kifejlesztett vonalfelismerő algoritmus képességeit.

Lényegében ugyanez a probléma érvényesül, csak nagyobb léptékben, amikor a megírás szakítja meg a szintvonalat. Egy szaggatott vonal esetében van rá lehetőség, hogy a vonal vastagságánál kisebb méretű szakadáson egyszerűen átfut a vonalfelismerő algoritmus, egy vonalnak tüntetve fel az egészet. Operátor beállítás függvénye, hogy mekkora szakadást fogad el a rendszer, azonban a hézag növelésével nő a hiba valószínűsége is – valóságban különálló vonalakat egynek tekint. Ha pedig egy egyébként folytonos vonalban csak néhány szakadás van, és azok eloszlása is egyenlőtlen, még nehezebb feladat megfelelő módszert találni az automatikus vektorizáláshoz.

Elmondhatjuk tehát, hogy egy szintvonalakat többé-kevésbé automatikusan, csak kezdeti, minimális operátori beavatkozást igénylő program megalkotása igen körülményes, kompromisszumokkal teli folyamat. Első és legfontosabb ilyen az eredeti raszteres térkép „hibátlansága” – tehát az a feltételezés, hogy a térkép készítője a szintvonalak ábrázolásakor mindig arra törekedett, hogy (lehetőség szerint megszakítás nélkül) a teljes izohipsza ábrázolásra kerüljön.

Önmagában ez a kritérium is nagyon sok térképet kizár, hiszen kisebb méretarányú térképeken sok esetben a szintvonal csak mintegy kiegészítőként szerepel a térképen, utaknak, épületeknek alárendelten a lakott területeken lényegében

eltűnik. Gyakori probléma az is, hogy a szintvonalakat más objektumokkal megegyező színnel ábrázolják, így lehetetlenné téve elkülönítését, hiszen egy számítógép csak valamilyen látható kvalitás alapján különítheti el a térkép objektumait. Ez a minőség pedig nem lehet más, mint a szín; sok vonalas objektum található egy térképen, csupán a vonalvastagság nem feltétlenül elég az azonosításhoz. Az emberi agy csupán a kontextusból is képes kiválasztani a szintvonalakat, de egy géptől ez nem várható el (legalábbis nem elérhető áron).

5. A problémák lehetséges megoldásai

5.1. Mértékegység-különbségből, vagy eltérő számírásból adódó problémák elhárítása

Egy automatikus szintvonal-vektorizáló program a legegyszerűbben kezelő bevonásával kaphatná meg a mértékegységeket. Ez vélhetően a legegyszerűbb, és egyben legmegbízhatóbb módszer is. Egy átlagos felhasználó emellett viszonylag ritkán találkozik eltérő mértékegység-rendszerű térképekkel.

Noha a legtöbb program célja az, hogy minél több terhet vegyen le a felhasználó válláról, egy térkép vektorizálásakor felmerülnek olyan problémák, amiket operátori beavatkozás nélkül nem lehet megbízhatóan megoldani – többek között ilyen a georeferálás. Léteznek ugyan automatikus georeferáló szoftverek (az ArcGIS 10.2-nél frissebb verziókban is van ilyen), ezek csak műholdas képek használata esetén használhatóak. Egy eredetileg papíralapú, szkennelt térkép sarokkoordinátáinak beolvasására, és a vetületi rendszer meghatározására még mindig a felhasználó a legalkalmasabb. Ha pedig minden egyes térképlap beolvasása minden esetben ennyi emberi beavatkozással jár, egy lenyíló ablakból kiválasztani a megfelelő számírást és mértékegységrendszert igazán nem nagy ár a tökéletes pontosságért.

Nem lehetetlen a gépesítés sem, csupán nem éri meg. Bármelyik kereten kívüli koordinátát megíró számot egy karakterfelismerő-rendszeren lefuttatva megmondható a felhasznált számrendszer. A szintvonalak alapszintközeinek mértékegysége is kitalálható a gép által: a georeferálás után ismertté válik a térkép méretaránya (legalábbis hozzávetőlegesen). A méretarány és a szintvonalak

szintközeinek ismeretében pedig – egy viszonylag egyszerű táblázat segítségével – viszonylagos pontossággal lehet következtetni a mértékegységre, hiszen

1 méter = 3,28 láb

így egy adott méretarányú európai térkép alapszintköze körülbelül háromszorosa egy azonos méretarányú amerikainak. Természetesen nagyon sík, vagy szélsőségesen hegyvidéki térképek esetén a határok összemosódhatnak, így ez a módszer nem teljesen megbízható. Előfordulhat az is, hogy egy szelvényen belül változik az alapszintköz (mint a magyar 1:10000-es topográfiai térképeknél). Ekkor, ha nincs minden szintvonal megírva, csak kézzel lehetséges az adatok digitalizálása.

5.2. A vonal megszakadásából adódó problémák elhárítása

A rendszertelen folytonossági hiányok (megírás, térképi jel, stb.) felismerésére nem mindig használhatóak az egyszerű szaggatott vonalak lokalizálására szolgáló algoritmusok. Ezek ugyanis általában azon az elven működnek, hogy az áthidalt rések mérete állandó, csakúgy, mint a vonalszakaszoké. Mivel elméletileg minden szintvonal önmagába tér vissza, vagy a térkép keretén ér véget, feltételezhetjük, hogy bármely talált folytonossági hiány csak ideiglenes. Ha valamilyen domborzati objektum szakítja meg a szintvonalat, az sohasem a semmiben végződik, hanem például egy metsző vonalba torkollik, és úgy hal el. Abban az esetben, ha a program ilyet nem talál, megakad.

Több lehetőség is kínálja magát ekkor:

5.2.1. Manuális megoldás

A program hibát jelez, és a kezelőnek kell manuálisan berajzolni a folytatást a következő csomópontig. Ez a legegyszerűbb opció, és kétségkívül a legmegbízhatóbb is, ám ha célunk a minél nagyobb fokú automatizálás, elkerülendő, mert ez jár a legtöbb munkával. Ha rendelkezésre állnak a térkép domborzati rétegét tartalmazó fóliák, még a szkennelés előtt be lehet ezen jelölni a megszakadó szintvonalak folytatását.

5.2.2. A Hough-féle algoritmus

A Hough-algoritmus jó szakadozott vonalak felismerésére, de sok esetben körülményes a használata. A legtöbb esetben a szakadás egy egyenesben, vagy egyirányú íven van, irányváltás általában nincs. Ez előnyös, mert a modernebb Hough-féle vonalfelismerő rendszer már ívekkel is megbirkózik, de egyszerre csak egyféle vonallal. Így a potenciális sokszori szükséges futtatás miatt ez a módszer nagyon erőforrás-igényes. A Hough-módszer emellett semmilyen információt nem ad a vonal hosszára vonatkozóan, így csak más módszerekkel karöltve használható. Fokozottan fennáll az a veszély is, hogy rossz fonalat kezd el követni az algoritmus, és egy szomszédos szintvonalba torkollik bele. Ennek valószínűsége megfelelő programozással jelentősen csökkenthető ugyan, de teljesen nem lehet kiküszöbölni.

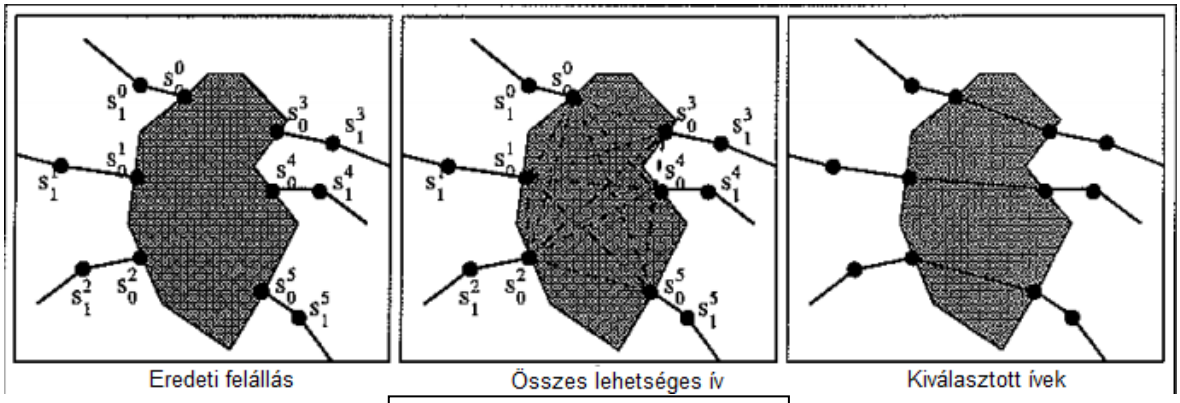
5.2.3. A szaggatott vonalat felismerő algoritmus

Hagyományos szaggatott vonal felismerésére szolgáló algoritmus tovább-fejlesztésével is lehet operálni, de ez sincs kockázatok nélkül. Minden ilyen program két kezdő adat bekérésével indít: a rés maximális mérete, illetve az a szögtartomány, amekkora eltérés lehet az eredeti vonaltól. Lényegében tehát egy körcikk adatait kérik, amiben megtalálható a következő vonalszakasz kezdete. A körcikk sugarának (a rés maximális méretének) adódik a kérdéses szintvonalnak legközelebbi szomszédjától való távolsága – hiszen el szeretnénk kerülni, hogy a szomszédos szintvonalba fusson bele. Ez önmagában is erősen csökkenti a módszer gyakorlati értékét, mert nagyobb szintkülönbségű területeken a szint-vonalak olyan közel futhatnak egymáshoz, hogy akár egy keresztező betű okozta szakadás mérete is meghaladja ezt a távolságot. A szög meghatározása még fogasabb kérdés. Nyilvánvalónak tűnik, hogy a szaggatott vonalak felismerésekor használt szélesebb intervallumok itt nem megengedhetők. Minél keskenyebb szögtartományt nézünk, annál jobban kitolható a keresési távolság. (Ezzel a megoldással szakirodalomban eddig nem találkoztam, de véleményem szerint egy olyan egyenlet birtokában, amivel a körcikk adatait ki lehet számolni, viszonylag megbízható módszerhez lehetne jutni.)

5.2.4. A Shimada-féle algoritmus

S. Shimada és munkatársai 1995-ben egy olyan algoritmust fejlesztettek ki, amiben a szintvonalak szomszédjaik segítségével lépnek tovább a szakadásokon, összeéréseken. A felhasználónak ki kell jelölnie egy szintvonalköteget, amiben megadott számú vonal fut. A program ezek után egymással párhuzamosan vektorizálja a köteg szintvonalait, és ha nehézségbe ütközik egy vonalon, szomszédjai vonalát használja támaszul. Egyes vélemények szerint ez bizonyos különleges esetekben nagyon hasznos módszer, de a leghétköznapibb szakadásokra nem nyújt jó megoldást.(#3)

5.2.5. A Dupont-Gondran-módszer



6. ábra – az első eljárás ábrán

Francois Dupont és Michel Gondran egy teljesen automatikus térkép-interpretáló módszert fejlesztett ki 1999-ben, a nagyobb hatékonyság érdekében kombinálva saját fejlesztéseiket a már meglevő módszerekkel. Az első eljárás akkor kerül alkalmazásra, ha valamely előtérbeli tárgy (például egy út, ami elfedi a szintvonalat) okozza a szakadást. A rendszer megkeresi az azonos objektum által megszakított végeket, és az összeset összeköti egymással. Az egymást keresztező vonalak kizárása után egy „energia” változó révén kiválasztja a helyes összekötéseket. A változó egy másodfokú függvény eredménye, ami számításba veszi a távolságot, csakúgy, mint a vektor két utolsó pontja alkotta egyenestől való szögeltérést. A részletes formulát nem ismertették. A következő módszer a kisebb szakadások eltüntetésére szolgál, a szintvonalak kezdeti vektorizálása során keletkező kettős csontvázat használva fel. A harmadik módszer egyfajta biztosítékként

[3] Dr. Katona Endre: Automatikus térkép-interpretáció – PhD értekezés, Szegedi Tudományegyetem, 2000

szolgál: ha a másik kettő nem járt megfelelő eredménnyel, de ezen felül is főként a hosszabb, görbe hiányosságokat képes pontosan közelítve pótolni. Shimada metódusát fölhasználva, a szomszédos szintvonalak segítségével következtet a hiányosságra, de csak a saját fejlesztések megtartásával – a kettős vonalváz alapján kutatva.

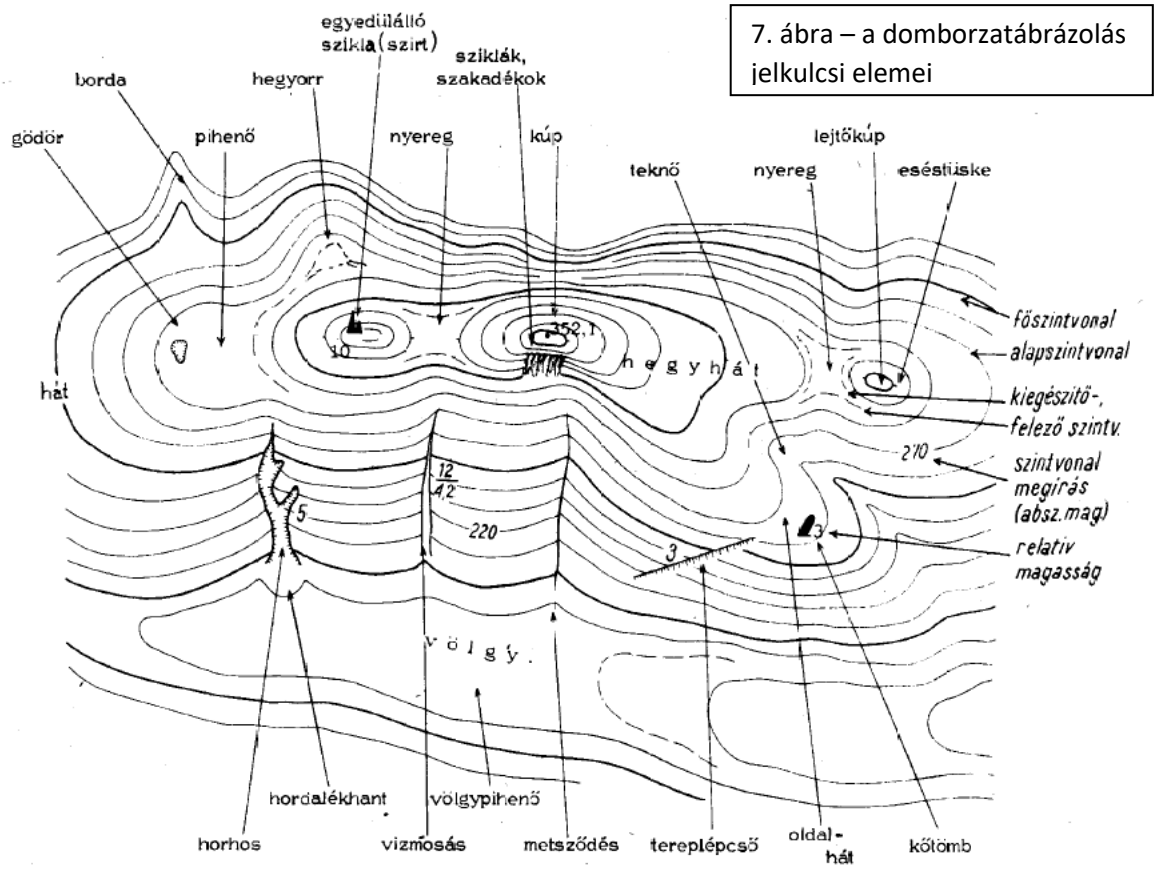
5.2.6. A Goodson-Lewis-módszer

K. J. Goodson és P. H. Lewis 1990-ben egy olyan módszert tettek közzé a szakaszok folytonossági hiányának kezelésére, ami minden addigitól eltérően adatbázis-építésen alapul. A rendszerben tárolt megannyi, már megoldott példa révén hivatott a program következtetni az aktuális probléma helyes megoldására. Szintvonalak estén ez lényegében minden nagyobb különálló térképcsalád vektorizálása előtt (vagy alatt) egy teljesen önálló adatbázis készítését teszi szükségessé. Ez kezdetben nagyon lelassítja a folyamatot, ám a későbbiek során a program egyre növekvő önállósággal és pontossággal lesz képes üzemelni. A módszert (legalábbis részleteiben) sok későbbi algoritmusban is megtalálhatjuk, a továbbiakban is bizonyítva annak életképességét.

5.3. Domborzati jelek okozta vonalszakadás / zavar elhárítása

Az eddigiek során abból a feltevésből kiindulva operáltam, hogy a felismert szintvonal-rétegen a program külön csoportba teszi, mintegy hibaként kezeli az összeérő, nem-végtelen vonalakat. Az így megjelölt hibák nagy részét pedig az előzőekben részletezett módszerek kijavítják; a szakadt szintvonalakat újra összekötik.

Szélsőséges esetben akár minden vonalat összekötnek. Ez nem a helyes eljárás, hiszen például egy metsződés esetén nem volna jó, ha a vonal a tulajdonképpeni objektum határain túl is folytatódna. Ez jelentős mértékben csökkentené a készülő digitális térkép (vagy domborzatmodell) pontosságát. Mindennél fontosabb tehát a szakadozott vonalakat összekötő algoritmus paramétereinek helyes megválasztása.



7. ábra – a domborzatábrázolás jelkulcsi elemei

A domborzatábrázolás jeleit szerencsére (általában) egyéb ismérveik alapján is el lehet különíteni a szintvonalaktól:

- Jellemző vonalvastagságuk nagyobb, mint az alapszintvonalaké, hiszen ki kell tűnniük környezetükből, hogy az emberi olvasó felismerje jelentőségüket. Önmagában erre még nem feltétlenül lehet építeni, mert a vonalvastagságuk még így sem feltétlenül haladja meg a főszintvonalakét.
- A vonalak végei jellemzően elkeskenyednek: a nagyobb vonalvastagság bár általában jellemző, a szakaszok végei jórészt hegyesebbek. Ez a tulajdonság bár nem feltétlenül előírás, mégis szinte minden kézzel készült topográfiai térképnek sajátja. A térképkészítő szinte öntudatlanul az objektum határának közeledtét érzékelteti ezzel. A gyakorlatban ez szemmel látható különbséget jelent a szintvonalakhoz képest, kizárólag erre építeni az elkülönítést azonban mégsem lehetséges. Akár a szkennelés, akár a papírtérkép előélete túl könnyedén okozhat ilyen hibát egy szintvonal beolvasása során, nem is beszélve a vektorizálás során fellépő estleges hibákról. Ha egy ilyen hiba véletlenül a szintvonal

megszakadásának környékére esne, nem volna helyes azt is külön domborzati jelként kezelni.

- Gyakran valamilyen extra jel alkalmazása teszi jobban elkülöníthetővé az objektumot – legyen az akár egy elágazás (vízmosás), vonal-összeérések (szakadék), vagy valamilyen fogazás (horhos, terep-lépcső). Ezen attribútumok többnyire egyediek, így ezek alapján valamelyest el lehet különíteni sok domborzati jelet, de nem mind-egyiket. A metsződés vonala például megszólalásig hasonlít egy szint-vonaléra, ha a végein az esetleges elkeskenyedéseket nem számítjuk (és nemigen számíthatjuk). Az eddig ismertetett jellemzők tehát önma-gukban, de együttesen sem feltétlenül elegek a domborzati jelek azonosításához, de metsződések talán legmarkánsabb megkülön-böztető vonásáról még nem esett szó.
- Shimada és társai abban alkottak maradandót, hogy a számítógépes szintvonal-felismerésbe bevezették a környező vonalak alapján történő hiánypótlást, de módszerük használható a nem odaillő objektumok – így az általunk keresett domborzati objektumok kiszűrésére is – csak a körülmények változtak meg. Egy metsződés (de a többi vonalszerű ob-jektum nagy része is) a szintvonalakat jórészt merőlegesen metszi. Amennyiben Shimada metódusát kissé módosítva arra is lehetne használni, hogy megmérjük a környező azonos szinten levő vonalaknak (tehát jórészt szintvonalaknak) a kérdéses vonalunkkal bezárt szögét (és esetleg metsződéseiket), nagy valószínűséggel rá lehetne mutatni a különleges domborzati elemekre.

Az eddig említett összes jellemző együttes vizsgálata révén véleményem szerint szét lehet választani egy térkép szintvonalait és egyéb domborzati jeleit. Ezek az objektumok attribútumaik alapján egyértelműen azonosíthatók még egy gép számára is, kérdéses azonban az, hogy a program az azonosításon felül mit csináljon velük. Amennyiben a digitalizálás célja „csak” egy digitális topográfiai térkép elkészítése, kézenfekvőnek tűnik az eredeti jel vonalait követő új objektum „rávetítése” az eredeti helyen, csupán attribútumként őrizve meg annak felismert nevét. A felhasználó e módon digitálisan hozzáfér az adatokhoz kereshető

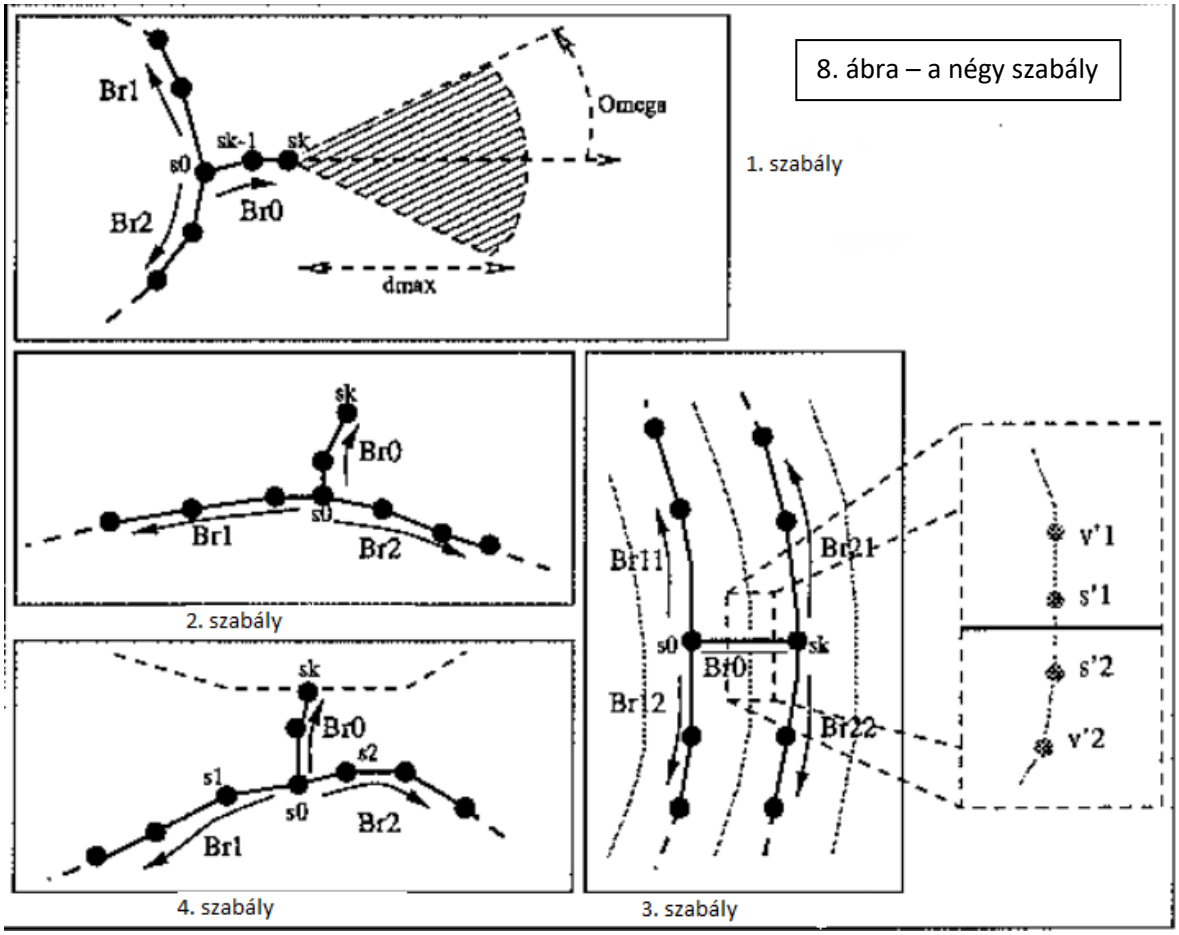
formában, bár az így átörökített vonalcsatlakozások nem feltétlenül őrzik meg kapcsolatukat az esetleges transzformációk után.

A biztosan jó megoldáshoz sajnos itt is a felhasználó beavatkozása szükséges – a legtöbb megvalósult módszer esetén ez is történik rögtön a vonalhiba felismerése után. Ha azonban a térkép digitalizálásának célja egy digitális terepmodell elkészítése, az objektum vonalainak egyszerű átmásolása nem lesz elegendő, bonyolultabb műveletek szükségesek. Lehet, hogy külön be kell táplálni a minden lehetséges domborzati jel vonalait kísérő függőleges irányváltásokat, és így építeni be a kész modellbe. A korábban elkövetett esetleges tévedések – téves beazonítások – hatása ekkor meghatványozódik, így különösen fontossá válik az utólagos ellenőrzés (vagy ha nagyon megbízhatatlan a program; a manuális digitalizálás).

5.4. Eséstüskék és szkennelési hibák / térképhibák okozta problémák elhárítása

Mivel a szintvonalak kezdeti kezelési módjának a színszűrés tűnik az egyetlen járható útnak, természetesen sok zavaró tényező kerül a kezdetben vektorizálásra váró még raszteres, de már színszűrt képre. Ilyen zavaró tényezők lehetnek többek között kisebb, azonos színű térképi objektumok, szkennelési hibák, de ide sorolhatók az eséstüskék is, hiszen a kész digitális térképen már nem lesz helyük. Előnyösebb az eséstüskéktől még a folyamatnak a kezdeti szakaszában megszabadulni, hiszen így a későbbiekben nem zavarják meg a tulajdonképpeni vektorizálás kényes szakaszát, illetve hasznos információval is szolgálhatnak. A lejtés irányával jobb minél hamarabb tisztába jönni, mert ez megkönnyítheti a későbbi kész vektoros rajz szintvonalainak paraméterezését.

A probléma Francois Dupont és Michel Gondran által kínált megoldása kellően jól kezeli a kérdést, de természetesen nem tévedésbiztos. Ők négy alapvető szabálynak feleltetik meg a kérdéses szabad végződéseket, így döntve el, hogy van-e maradásuk a később szintvonalakként kezelt vonalak között.

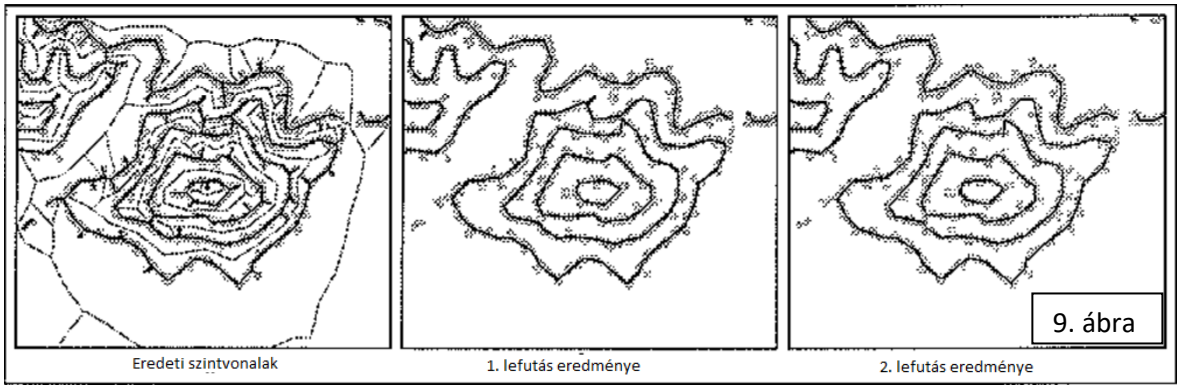


Minden szabály tartalmazza azt a kitételt, miszerint a levágandó vonalcsonk (minden esetben B_{r0}) maximum három sarokpontból (s_0 - s_k) állhat.

1. szabály: A valódi szintvonal (az ábrán két félre szakadva: B_{r1} és B_{r2}) mindkét irányban hosszabb, mint egy megszabott küszöbérték. A csonk utolsó két csomópontja által meghatározott meghosszabbított egyenes oldalirányban kiterjesztett (d_{max} távolság és Omega szög segítségével) palástjában nincs takaró elem (olyan, ami egyébként elfedi a szintvonalat). Ez a szabály valamennyire bizonyságot ad arról, hogy nem egy hosszabb, csak megszakadó vonalat törölnek ki.
2. szabály: E szerint kettő, egymástól eltérő küszöbértéknél is nagyobb a szétvágott szintvonalunk (B_{r1} és B_{r2}) csomópontjainak száma. Ez a szabály biztosítja azt, hogy a csonk hossza *jóval* kisebb, mint az eredeti szintvonalé.

3. szabály: Két egymással nagyjából párhuzamosan futó, de összekötött szakaszok esetén – ha szintvonalakról beszélünk – legalább egy vonal bizonyosan fölösleges. Ennek a vonalnak a kiszűrésére szolgál ez a szabály, megszabva a már ismert minimum hosszúságot (mind a négy vonalszakasz $\{B_{r11}; B_{r12}; B_{r21}; B_{r22}\}$ esetén, illetve egy maximum szöget definiálva (v'1;s'1 és v'2;s'2 által bezárt szög).
4. szabály: Az s1_s0_s2 szög abszolútértéke egy szabott határértéknél kisebb.

A szűrés ebben a sorrendben zajlik le az 1. lépéstől a 4-ig. Amennyiben az első szűrés után is túl sok „elvarratlan szál" marad, változtatott értékekkel újra lefuttatják a szűrést a 2. lépéstől kezdve. Kétszeri lefutás után szinte mindig kielégítő az eredmény. Dupont és Gondran publikációjukban sajnos nem közlik a számukra optimálisnak vélt változó értékeket, így csak az általuk megadott ábrákból lehet következtetni a rendszer hatékonyságára.



9. ábra

Bizonyos, hogy a program működik, hiszen sikerrel használták topográfiai térképek digitalizálására, ám semmiképpen sem tökéletes. Evidensnek tűnhet legfőbb gyengesége: amennyiben egy véletlen folytán háromnál több csomópont kötődik össze egy szintvonallal, a rendszer tehetetlen. Lehetséges, hogy ezt a számot a program üzemeltetői a második iteráció során felemelik, de nagyon óvatosnak kell lenni vele; könnyen eltűnhet az igazi szintvonal is. Ez a módszer a különleges domborzati jelek (mint a tereplépcső) fogazását is eltüntetné, így lehetetlenné téve helyes azonosításukat.

Az algoritmus készítői szerint is további fejlesztésre szorul, és ezt a véleményt osztja Dr. Katona Endre is, doktori disszertációja szerint. Én úgy gondolom, hogy tökéletlensége ellenére (nem is lehetne teljesen tökéletes) még mindig ez a jelenleg elérhető leginkább üzembiztos teljesen automata szintvonal-vektorizáló rendszer.

5.5. A szintvonalszám-leolvasás / rögzítés során felmerülő problémák megoldása

Ismereteim szerint jelenleg nem létezik olyan térkép-interpretáló szoftver, ami teljesen automatikusan rendelne magasságértéket az összes szintvonalhoz. Ez több oknak köszönhető: a szintvonalszámok viszonylag ritkák, az eséstüskék felismerése bizonytalan, és az alapszintköz sem minden térkép esetén egyértelmű. Több program tartalmazza a szintvonalszámok automatikus felismeréséhez szükséges kellékeket, és sikerrel is alkalmazza azokat. Ebben az esetben vélhetően a legegyszerűbb megoldás a célravezető. A program a betáplált szintvonalakkal megegyező színű réteget vékonyítással vektorizálja, és a keletkező „betűcsontvázakat" egy hagyományos optikai karakterfelismerő rendszeren (OCR) átvezetve 90% fölötti pontossággal megkapja a karaktert. Ha kizárólag számot kell felismernie, bizonyos, hogy a pontosság még jobb.

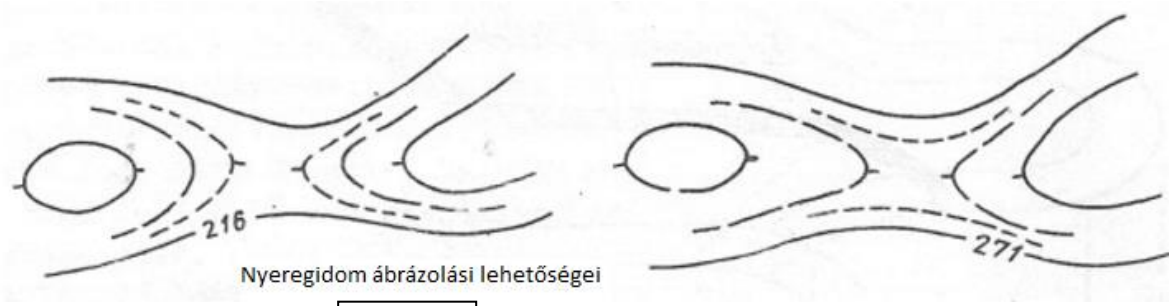
További problémát jelent, hogy a szintvonalszámok viszonylag ritkák – néha csak egy-egy főszintvonalon találni megírást. Ezeket tehát a rendszer elméletileg fel tudja paraméterezni, de a köztes területekre csak következtetni tud. Egyfajta megoldást kínál az, ha a térkép alapszintközét előre megadjuk, így könnyebb a rendszernek számolnia. Az egymás mellett sűrűn (vagy pont nagyon ritkán) sorakozó szintvonalak még így is megzavarhatják a számítógépet, főleg tekintetbe véve az eséstüskék felismerésének megbízhatatlanságát. Sok kanyarral, megszakadással tűzdelt szintvonalrengetegben a program tehát könnyen eltévedhet, és egy kis tévedés néhány szintvonallal odébb már nagyon nagy eltéréseket okozhat. Ez az oka annak, hogy a vektorizálás ezen szakasza még mindig interaktív segédletet igényel – de legalábbis nagyon szigorú utólagos ellenőrzést.

A megvalósult digitalizáló szoftverek túlnyomó többsége esetén a szintvonalak paraméterezése manuálisan történik, a gép esetleg javaslatot tesz a szomszédos (már meghatározott) szintvonal függvényében, vagy egy kijelölt szintvonalköteg elemeit címkézi fel (ha egyiküknek már van értéke). Nagyon ritka az a szoftver, ahol csak hibajelzés esetén, vagy a végső ellenőrzéskor kell az operátornak bekapcsolódnia.

5.6. Szintvonalak hierarchiájával kapcsolatban felmerülő problémák megoldása

A szoftvernek feltétlenül fel kell ismerni és elkülöníteni az alap-, a fő-, a felező-, és negyedelő szintvonalakat egymástól. A főszintvonalak elkülönítése a legegyszerűbb: a vektorizálás folyamán a futásirányra merőlegesen néha egy mérést indítunk, ami méri a vonal átlagos vastagságát pixelekben. Amennyiben ez az érték legalább egy küszöbértékkel meghalad egy korábban mért vastagságot, az illető vonal főszintvonal. Felismerésük azért fontos, mert több folyamatban segítségünkre lehet. A paraméterezés alatt az ellenőrzésben segíthet: a főszintvonal (még ha esetleg nincs is megírva) valamivel kerekebb szám, mint a többi. Egyszerű vektorizálási hibaellenőrzésnél is segíthet a tudat, hogy két főszintvonal között pontosan négy alapszintvonalnak kellene lennie.

A felező- és negyedelő szintvonalak felismerését már részleteztem, jelentőségük a domborzati formák hangsúlyozásában rejlik. Magassági értéket viszont csak manuálisan javasolt nekik kapni. Már megjelenésük oka is (néha) valamilyen szokatlan objektum, egy rendellenesség kiemelése, ezekkel pedig a számítógép hagyományosan nem jól boldogul. Egy nyereg térképi ábrázolásánál például még az emberi olvasó is tévedhet.



Nyeregidom ábrázolási lehetőségei

10. ábra

6. Összefoglalás

A szintvonalak vektorizálása tehát egy igen bonyolult, egymástól több, jórészt független részre osztható feladat. Minden részterületnek megvannak a maga szakértői, akik többnyire eltérő módon képzelik el a feladat megoldását. Egyértelműen jó vagy rossz módszer nincs. A ma létező egyik módszer sem tökéletes, mindnek vannak gyenge pontjai, és lehet is rajtuk javítani a pontosság és a magasabb fokú automatizálás érdekében.

A teljesen automatikus vektorizálás ma még elérhetetlennek tűnik; fejlettebb mesterséges intelligenciát igényel. A részfolyamatok közül legalacsonyabb fokú az automatizálás a szintvonalak paraméterezése terén, mert itt a legkisebb hiba sem megengedhető. Az egyéb domborzatábrázolási jelek is többnyire kézi erővel kerülnek föl a képernyőre, itt azonban elképzelhető a magasabb fokú gépesítés. A teljesen manuális digitalizálás már szinte csak egyetemi feladatként létezik, a szintvonalak nagy részének automatikus vektorizálása megfizethető áron elérhető.

A sok módszer közül az automatizálás folyamatában elől járnak azok, akik több más módszert szintetizálnak a saját eredményeikkel. Vélhetően ez a jövő útja.

7. Hivatkozások

7.1. A szöveg forrásai:

7.1.1. Nyomtatott források:

Zentai László (2000): Számítógépes térképészet. ELTE Eötvös Kiadó, Budapest

Dr. Katona Endre (2000): Automatikus térkép-interpretáció. Szegedi Tudományegyetem – PhD értekezés

Klinghammer István, Papp-Váry Árpád (1983): Földünk tükre a térkép. Gondolat Kiadó, Budapest

7.1.2. Digitális források:

History of GIS - *https://www.gislounge.com/history-of-gis/*

Raster vs. Vector graphics - *https://www.pixellogo.com/printing/raster-vs-vector-graphics*

OpenCV: Hough Line Transform - *http://docs.opencv.org/2.4/doc/tutorials/imgproc/imgtrans/hough_lines/hough_lines.html*

Anne Solberg: Hough transform - *http://www.uio.no/studier/emner/matnat/ifi/INF4300/h09/undervisningsmateriale/hough09.pdf*

Hough Transform - *http://homepages.inf.ed.ac.uk/rbf/HIPR2/hough.htm*

Dr. Siki Zoltán: Szintvonalak vektorizálása VPstudio-val - *http://www.agt.bme.hu/tantargyak/mernlet/r2v.pdf*

Karl Tombre, Salvatore Tabbone: Vectorization in graphics Recognition: To Thin or not to Thin - *https://members.loria.fr/KTombre/tombre-icpr00.pdf*

R2V Software description - *http://www.ablesw.com/r2v/contour.html*

Grass Tutorial: Digitizing Vector Maps - *https://grass.osgeo.org/gdp/grass5tutor/HTML_en/c922.html*

Több Wikipedia oldal: CAD data exchange, Pattern recognition, Topographic map, Terrain cartography, Geographic information system, Graphics tablet, Hough transform, Vector graphics, Image tracing - *https://en.wikipedia.org/wiki/CAD_data_exchange; https://en.wikipedia.org/wiki/Pattern_recognition; https://en.wikipedia.org/wiki/Topographic_map#United_States; https://en.wikipedia.org/wiki/Terrain_cartography https://en.wikipedia.org/wiki/Geographic_information_system https://en.wikipedia.org/wiki/Graphics_tablet https://en.wikipedia.org/wiki/Hough_transform https://en.wikipedia.org/wiki/Image_tracing https://en.wikipedia.org/wiki/Vector_graphics*

Több *books.google.com* könyvrészlet:

Thomas C. Henderson: Analysis of Engineering Drawings and Raster Map Images (1-30. oldal)

Sergey Ablameyko: An Introduction to Interpretation of Graphic Images (70-kb. 100. oldal)

Karl Tombre,Atul K. Chhabra: Graphics Recognition: Algorithms and Systems: Second International Workshop 2. kötet (190-206. oldal)

Line detection - *http://homepages.inf.ed.ac.uk/rbf/HIPR2/linedet.htm*

QGIS Tutorial: Digitizing Map Data - *http://www.qgistutorials.com/en/docs/digitizing_basics.html*

Dov Dori, Wenyin Liu: Sparse Pixel Vectorization: An Algorithm and Its Performance Evaluation - *http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.20.1546&rep=rep1&type=pdf*

Dov Dori, Wenyin Liu: Automated CAD Conversion with the Machine Drawing Understanding System – részletek - *http://esml.iem.technion.ac.il/wp-content/uploads/2011/02/10.1.1.10.737.pdf*

Topographic Map Contour Lines - *http://outdoorquest.blogspot.hu/2014/04/topographic-map-contour-lines_16.html*

Mélykúti Gábor: Topográfia 5., Domborzattan II. - *http://www.tankonyvtar.hu/hu/tartalom/tamop425/0027_TOP5/ch01s02.html*

Mélykúti Gábor: Topográfia 11., *Magyarországi térképezések története* - *http://www.tankonyvtar.hu/hu/tartalom/tamop425/0027_TOP11/ch01s02.html#id487955*

Szövegkijelölés, karakterfelismerés, vonalhasználat - *http://www.lti.hu/index.php/szovegkijeloles-karakterfelismeres-vonalhasznalat*

Dr. Sárközy Ferenc: Térinformatika – több részlet - *http://gisfigyelo.geocentrum.hu/sarkozy_terinfo/tbev.htm*

C.P.Lai, R. Kasuri: Detection of Dashed Lines in Engineering Drawings and Maps - *http://www.iapr-tc11.org/archive/icdar1991_proc/proceedings/507.pdf*

Bin Kong, Ishin T. Phillips stb.: A Benchmark: Performance Evaluation of Dashed-line Detection Algorithms – részletek – *http://haralick.org/conferences/benchmark_performance_dashed_lines.pdf*

ArcGis,ArcMap leírás:Georeferencing a raster automatically to another raster - *https://desktop.arcgis.com/en/arcmap/10.3/manage-data/raster-and-images/georeferencing-a-raster-automatically.htm*

Lab3: Contour mapping and topographic profile - *http://www.iupui.edu/~geogdept/g108/lab_3.htm*

Francois Dupont, Marc P. Deseilligny, Michel Gondran: Automatic interpretation of scanned maps: Reconstruction of contour lines - *http://recherche.ign.fr/labos/matis/pdf/articles_conf/1990s/dupont_mpd_GREYC_1997.pdf*

C. Helvaci, B. Bayram: Semi automatic digitizing of contours from 1:25000 scaled maps - *http://www.isprs.org/proceedings/XXXV/congress/comm3/papers/314.pdf*

7.2. A képek forrásai:

1. ábra: *http://www.fastprint.co.uk/Assets/User/17-1-vector-vs-raster.jpg*

2. és 3. ábrák: Dr. Katona Endre (2000): Automatikus térkép-interpretáció. Szegedi Tudományegyetem – 9. és 10. oldal

4. ábra: *http://www.uio.no/studier/emner/matnat/ifi/INF4300/h09/undervisningsmateriale/hough09.pdf*

5. ábra: *http://www.tankonyvtar.hu/hu/tartalom/tamop425/0027_TOP5/ch01s02.html*

6. ábra: Francois Dupont, Marc P. Deseilligny, Michel Gondran: Automatic interpretation of scanned maps: Reconstruction of contour lines - *http://recherche.ign.fr/labos/matis/pdf/articles_conf/1990s/dupont_mpd_GREYC_1997.pdf*

7. ábra: Dr. Katona Endre (2000): Automatikus térkép-interpretáció. Szegedi Tudományegyetem – 76. oldal

8. és 9. ábrák: Francois Dupont, Marc P. Deseilligny, Michel Gondran: Automatic interpretation of scanned maps: Reconstruction of contour lines - *http://recherche.ign.fr/labos/matis/pdf/articles_conf/1990s/dupont_mpd_GREYC_1997.pdf*

10. ábra: *http://www.tankonyvtar.hu/hu/tartalom/tamop425/0027_TOP5/ch01s02.html*

8. Köszönetnyilvánítás

Elsőként köszönetet mondok *Zentai László egyetemi tanárnak,* az Eötvös Loránd Tudományegyetem Térképészet és Geoinformatika tanszék vezetőjének – és téma-vezetőmnek – türelméért, tanácsaiért, és konstruktív kritikájáért; mindezek nélkül nem születhetett volna meg ez a dolgozat.

Emellett köszönettel tartozom a tanszék tanárainak, akik mélyebben megismertették, és megszerettették velem a szintvonalak világát.

Végső soron pedig köszönöm a családomnak, hogy kitartottak mellettem addig is, amíg ez a dolgozat meg nem született.