Faro Focus 120: http://www.laserscanning-europe.com/de/system/files/redakteur_images/TechSheet_Laser%20Scanner_Focus3D.pdf
NBS National BIM Survey 2016: National BIM Report 2016
Varga Tímea 2016. BIM modellezés lézerszkennelés támogatásával. Diplomamunka

Summary

Supporting BIM by Terrestrial Laser Scanning

Changes in building rules and regulations can broaden the application field of BIM procedures (NBS National Survey, 2016). First, new constructions should by supported by BIM, but the technology can be used effectively in case of some existing buildings, too. Terrestrial laser scanning is a powerful tool to acquire detailed data from existing buildings or from those under construction. The laser scanned point cloud provides accurate, full coverage representation of the current state, and ensures access to data that are not present on the blueprints. The model based on laser scanned point cloud enables integrating objects surveyed with other data acquisition technologies. Surveying the finished building, even with furniture and with all equipment inside supports not only quality control but provides useful information for building operation purposes.

Somogyi Árpád
doktorandusz

BME Fotogrammetria és Térinformatika Tanszék
e-mail: somogyi.arpad@epito.bme.hu

Dr. Lovas Tamás
docens

BME Fotogrammetria és Térinformatika Tanszék
e-mail: lovas.tamas@epito.bme.hu

Vonalak automatizált generalizálása az elméletben és a gyakorlatban – Vonalegyszerűsítő és -simító eljárások

Ungvári Zsuzsanna

1. Bevezetés

A térképre a földfelszínt három elemtípussal képezzük le: pontokkal, vonalakkal és felületekkel. Ebben a cikkben a vonalas elemek automatizált generalizálási lehetőségeivel foglalkozom. Mivel a felületek kontúrjának, vagyis körvonalának generalizálása visszavezethető vonalas elemekre, ezért ezekre is használhatók az itt bemutatott módszerek, némi kiegészítéssel. A vonalas elemek lehetnek például a szint- és mélységvonalak, vonalas vízrajzi elemek (folyók, patakok), utak, vasutak, határvonalak és partvonalak; a felületek pl. felületi vízrajzi elemek (tavak, tározók), igazgatási egységek, felszínborítottság, épületek stb.

A magyar szakirodalomban hét generalizálási szabályt, vagyis elemi folyamatot különítünk el (Klinghammer–Papp-Váry 1983), de a jelen cikkben csak az elemek egyszerűsítésével foglalkozom. Az angolszász szakirodalom általában több elemi folyamatra bontja a generalizálást, de ezek tulajdonképpen „lefedik" a hét szabályt. Az eltérés akkor szembetűnő, ha a folyamatot automatizáltan valósítjuk meg, vagyis egy algoritmus végzi el helyettünk a vonalak generalizálását. Ekkor vonalegyszerűsítő és -simító algoritmusokat is meg kell különböztetnünk.

Az 1960-as évektől kezdve dolgoztak ki olyan számítógépes eljárásokat, amelyekkel megpróbálták kiváltani a munkaigényes és monoton folyamatokat, például így megszülettek az első vonalgeneralizálási algoritmusok (Ramer 1972, Douglas–Peucker 1973). Az algoritmus matematikai és logikai műveletek véges sorozata, amely egy folyamatot reprezentál (Rogers 1987). A generalizálás automatizálása algoritmusok kidolgozásával lehetséges. Lévén a generalizálás szubjektív feladat, ezért nem egyszerű a matematikai alapokra helyezése, máig nagy kihívás elé állítja a kutatókat. A célom az volt, hogy olyan eljárásokat alkossak, amelyekkel a generalizált térképi tartalom minőségében a legjobban hasonlít a hagyományos úton készült térképekhez. Az automatizált generalizáláshoz vonalegyszerűsítő és -simító algoritmusokat használtam, ezért a következő részben röviden ismertetem a rendelkezésünkre álló eljárásokat. A cikk célja, hogy részletes áttekintést nyújtson az algoritmusok működéséről. A matematikai háttérről részletesebben az eredeti cikkekben olvashatnak.

2. Vonalegyszerűsítés és simítás

A hagyományos térképek rajzolásánál akár manuálisan, akár térképrajzoló szoftverben a térképszerkesztő „fejében" zajlik a generalizálás: csökkenti a kanyarulatok számát, simítja a vonalat. Ha ezt a folyamatot automatizálva szeretnénk végrehajtani, matematikai alapokra kell helyezni az egyes lépéseket. Alapvetően két csoportba soroljuk az e célra alkalmas algoritmusokat: egyszerűsítő és simító algoritmusokra (Slocum 2005). Egyszerűsítés során a vonal töréspontjainak száma csökken, a vonal szerkezete egyszerűsödik.

Simítás során a csúcsokat, szögleteket távolítjuk el, ezáltal kerekebbé, „simábbá" válik a vonal: új pontokat számítunk ki. Li (2007) ezzel szemben egy újabb csoportot is bevezetett, ezek a méretarányfüggő generalizáló algoritmusok. Ezekben a méretarányszám paraméterként megadható. Véleményem szerint ugyan mindkét csoportosítás helytálló, de a Li által méretarányfüggő generalizáló algoritmusok besorolhatók a vonalegyszerűsítők közé, annak egy alcsoportját képezik.

Az egyszerűsítő algoritmusok áttekintése

Ezeket a rutinokat többféleképpen is csoportosították, magyarul is megjelent a Térinformatikai alapismeretek c. könyvben fordításként (NCGIA Core Curriculum 1994). Li ezt használta fel és fejlesztette tovább 2007-ben. Az algoritmusok alapos ismerete alapján a következő csoportosítást javaslom: méretarány-független és méretarányfüggő eljárások. A méretarány-független algoritmusoknál nem adható meg sem a kiindulási méretarányszám, sem a célméretarányszám paraméterként (természetesen közvetett módon a különböző toleranciaértékhez rendelhető méretarány pl. Ungvári 2015): ide sorolom a hagyományos vonalegyszerűsítő rutinokat, az alább felsorolt csoportokban, amely már McMaster és Shea kategorizálásán alapul. Ezeknek az algoritmusoknak a futtatásához legalább két bementi paraméter szükséges: a vonalláncok töréspontjainak koordinátái, és legalább egy toleranciaérték, ez lehet távolság, terület, szög, számosság. A teljesség igénye nélkül, a kartográfiai gyakorlatban eredményesen használható, vagy a térinformatikai szoftverekbe épített algoritmusokat tekintem át, kiemelve legfontosabb jellemzőiket. Egy korábbi munkámban részletesen kitérek a szoftveres lehetőségekre és javaslatot teszek az automatizált generalizálás oktatására is (Ungvári 2016).

Méretarány-független eljárások

A méretarány-független eljárások esetében a vonalak meglévő csomópontjainak száma csökken, általában megtartják az eredeti csomópontokat, de némely esetben előfordulhat, hogy újakat számítanak ki.

Független pontok módszere. Egyszerű algoritmusok, nem veszik figyelembe a vertexek közötti kapcsolatokat. *N-edik pont módszer:* megtart minden n-edik pontot a vonal töréspontjai közül. N természetes szám. Túl sok töréspont esetén érdemes először ezzel ritkítani a csomópontok számát, de önmagában nem alkalmas generalizálásra.

Lokális módszerű (szomszédospont-vizsgálati) eljárások. Szomszédos pontokat vizsgál, például köztük lévő távolság, általuk bezárt szög, vagy ezek kombinációja alapján. A pontok közötti távolság és az általuk bezárt szöget felhasználó algoritmust *Jenks* után nevezték el. A merőleges távolság esetén a vizsgált, vagy más néven a kritikus pont előtti és utáni csomópontot köti össze egy képzeletbeli egyenessel, és a kritikus pont egyenestől való távolságát vizsgálja meg.

Korlátozottan kiterjesztett lokális eljárások. A pontok szélesebb szomszédságát nézik meg: például a Lang-, a Deveau-, és a Visvalingam–Whyatt-algoritmusok. A *Lang-algoritmusnak* két, a felhasználó által megadható paramétere van: az egy lefutásban maximálisan vizsgálandó pontok száma, és egy merőleges távolság az aktuálisan vizsgált első és utolsó pontot összekötő egyenestől (Lang 1969). A *Deveau-algoritmus* a felesleges, az objektum alakját nem meghatározó pontokat törli, helyenként újakat határoz meg. Két bemeneti paramétert adhatunk meg, ezek a simasági faktor és a maximális élességi szög (Deveau 1985). A *Visvalingam–Whyatt-rutinnál* az egyetlen választható paraméter az éppen vizsgált három pont által bezárt háromszög területe, vagyis a hatékony terület (Visvalingam–Whyatt 1993).

Nem korlátozott, kiterjesztett lokális eljárások. A *Reumann–Witkam-algoritmus* esetén a vonal morfológiája szabja meg a vizsgált pontok számát: a kritikus pontot a következő ponttal összekötő egyenes irányában húzott sávon belül elhelyezkedő pontokat törli, majd áthelyezi a kritikus pontot az első, sávon kívül eső pontba. A választható toleranciaérték a sáv teljes szélességének felét jelenti. Eredményeként hosszú egyenes szakaszok jönnek létre (Reumann–Witkam 1974). Ebbe a csoportba sorolom a *Wang(–Müller)-féle algoritmust* is, amely a vonalak kanyarulatait egyszerűsíti, és amelyben egy paramétert adhatunk meg, ez a kanyar mérete (a polyline kanyart leíró része, és a kanyar alapvonala által bezárt terület). A kanyar a vonal azon szakasza, ahol az egymást követő csomópontok irányultsága vagy negatív vagy pozitív, vagyis amíg nem következik be lokális irányváltás, azaz inflexió a vonal futásában (Wang–Müller 1998).

A globális eljárások a teljes vonallánc ot, vagy a vonal egy hosszabb szegmensét veszik figyelembe; iteratív módszerrel választják ki a kritikus pontokat. A legszélesebb körben alkalmazott algoritmus a *(Ramer–) Douglas–Peucker-féle egyszerűsítő rutin* (Ramer 1972, Douglas–Peucker 1973). A rekurzív eljárás merőleges távolságokkal dolgozik, ez a változtatható paramétere. Első lefutásban megvizsgálja, hogy melyik pont esik legtávolabb a kezdő- és a végpontot összekötő egyenestől. Ha ez a távolság kisebb, mint a toleranciaérték, ez lesz az új szakasz (vagy vonal); ha nagyobb; akkor rekurzívan újra hívja önmagát az eljárás, mindaddig, míg be nem fejezi a vonal vizsgálatát. A kezdő- és végpontot mindig megtartja.

Méretarányfüggő eljárások

A *méretarányfüggő rutinok* közé sorolom a Li által méretarányfüggő generalizálásként megadott algoritmusokat, amelyekben közvetlenül megadható a kiindulási és célméretarányszám is (itt ez lesz a toleranciaérték). Perkal ötlete az automatizálási gyakorlatba nem került át. Ennek lényege, hogy a vonal mentén mindkét oldalon köröket helyezzünk el oda, ahol a vonalnak inflexiós pontja van. A kör vonallánc ot érintő szakaszai lesznek a vonal új szakaszai. A belső és külső elhelyezésű körök különbségéből jön létre a generalizálási határzóna. Erre Perkal nem javasolt gyakorlati megoldást, de az örvény-algoritmus (angolul: whirlpool) működése ehhez hasonló (Dougenik 1980). Méretarány szerint változó sugarú köröket helyezünk el az egyes csomópontokban. Ahol a nem szomszédos körök átfedik egymást,

ott a méretarányhoz képest túl hegyes csúcs van, amit egyszerűsíteni kell.

A *Li–Openshaw-féle eljárások* figyelembe veszik az objektumok méretét. Ha az adott elem a célméretarányban kisebb, mint a minimális méret, akkor elhagyják. Bármilyen komplex felépítésű is az elem, ha nagyobb a minimális méretnél, egyszerűsíthető. A méretarány szerint definiálni kell egy raszteres rácshálót, amely az egyes celláiban elhelyezkedő objektumrészlet töréspontjaiból egy új pontot átlagol, majd ezeket köti össze (Li 2007).

Az *objektumok területét megőrző kartográfiai vonalgeneralizálási algoritmus* egy összetett metódus, amely két lépésben működik: először egyszerűsít, majd az ezután megmaradt, túl éles csúcsokat elsimítja. Az egyszerűsítés során csökken a vonal komplexitása: négy pontból álló csoportokból képez ponthármasokat úgy, hogy a négyszög területe megegyezik a háromszög területével, ezáltal nem keletkeznek bezárult felületek, mint pl. a Douglas–Peucker-algoritmusnál. A simítást csak ott hajtja végre, ahol túl éles szög keletkezett az egyszerűsítésnél is alkalmazott a területmegőrző módszer ellentétjével. Bemeneti paraméterként csak a célméretarány nevezőjét kell megadni (QGIS-es verzió). Az algoritmus egy tapasztalati úton meghatározott változót is tartalmaz (vonalvastagság), amelyről a szerző kijelenti, hogy interaktívan egyelőre nem állítható, csak a forráskódban, de a helyes eredményhez szükség lehet az eredeti rajz felülbírálására (Tutic 2009). Ezáltal az algoritmus a méretarányfüggő, egyszerűsítő eljárások közé tartozik, bár simítást is végez, az csak részleges, kivitelezésében nem tartozik a simító algoritmusok közé.

A vonalsimító eljárások áttekintése

A simítás megszabadítja a vonalláncokat az éles szögektől, csúcsoktól, az ún. zajoktól, és részletektől. McMaster és Shea három csoportba sorolja a simító algoritmusokat (Slocum 2005), míg Li már négy kategóriát állított fel (Li 2007). A kétféle csoportosítás alapján kidolgoztam egy újabbat, amely egyesíti a kettőt. A magyar nyelvű szakirodalomban eddig még nem foglalkoztak vonalsimító eljárások csoportosításával, elnevezésével. Vannak olyan csoportok, amely mindkét szerzőnél megegyeznek.

Súlyozott átlagok

A vonallánc csomópontjainak a szomszédos töréspontok átlaga alapján egy új pozíciót számít ki. Ide tartozik pl. *a simítás a McMaster-féle súlyozott átlaggal, és simítás a McMaster-féle csúsztatott átlaggal.* Mindkettőnél két bemeneti paraméter van, az egyik az átlagolásban részt vevő pontok száma (célszerű páratlan számú pontot átlagolni), a másik szám az új pont eredetihez való közelebb „csúsztatásának" mértékét fejezi ki [0–1] között, ahol 0 az eredetivel megegyező, 1 pedig az átlagolásban bevont pontokból számított új helyzetet jelenti. A csúsztatott átlag esetén az új töréspont eltolása lineáris, súlyozott átlag esetén a távolsággal fordítottan arányosan súlyozott. *Boyle „előretekintő" algoritmusa* (forward looking) esetén a megadható paraméter egy szám (x), amelytől függ az új pont helyzete. Az eredeti algoritmust kissé módosítva ültették át a gyakorlatba: a megadott szám a kritikus pont új helyzetének kiszámításához szükséges. A következő x darab pont 1/x súllyal vesz részt a kritikus pont új helyzetének számításában (Boyle 1970). A szerző többek között a mélységvonalak simításához ajánlja, ezért az algoritmust szintvonalakon is megvizsgáltam.

Görbeillesztés

Görbeillesztés során a vonalláncot görbékkel helyettesítjük. Mivel a térinformatikai szoftverek vagy fájlformátumok többségében nem támogatják a görbék tárolását, ezért ezeket polylineokkal ábrázoljuk úgy, hogy megfelelő sűrűséggel megadjuk a görbe pontjait, így a célméretarányban görbének látszanak. A görbéknek matematikai szempontból két fajtája lehet: interpolációs (a görbe átmegy a vezérlőponton) és approximációs görbék (követi a vezérlőpontot, de általában nem megy át rajta). A görbeillesztés során a gyakorlatban általában harmadfokú görbékkel helyettesítjük az eredeti „törött" vonalat. A szoftverekben használt görbéket harmadfokú egyenletekkel, vagy polinomokkal írhatjuk le a legegyszerűbben. A Bernstein-polinommal állítható elő a *Bézier-görbe* (Kovács 2011). Egy harmadfokú Bézier-görbe ívének futásvonala a két végpont és a két vezérlő v. más néven kontrollpont koordinátájának ismeretében írható le. Ha egy „törött" vonalat helyettesítek görbével, akkor az egyes görbeívek érintői a végpontokban párhuzamosak és folytonosak, így biztosítható a görbe vonal „sima" futása. (Két görbe folytonos illeszkedésű, ha az egyik görbe második deriváltjai a végponton megegyeznek a másik görbe második deriváltjaival a kezdőponton. Ezeket a folytonos, összetett görbéket szplájnoknak nevezzük.). Az egyes görbeívek futásvonalának számításához a két végpontra, és egy íveltségi együtthatóra van szükség: ezekből interpolálható a két kontrollpont, amivel már megadható görbe (Szirmay-Kalos 2003, AGG 2007).

A Bézier-görbéhez „küllemre" igen hasonló megoldást kapunk, ha Hermite-görbéket használunk fel. Ezek bár interpolációs görbék, de ugyanúgy megadhatók polinomokkal is: két pont és két érintő vektor lesz az input adat a görbeívek számításakor (Kovács 2011).

A görbék ezen kívül lehetnek approximációs *B-szplájnok* is. Ennek két altípusa van: a *NUBS* (nem uniform B-szplájn: az egymást követő görbeszegmenseknek nem egységnyi intervallum felel meg) és a *NURBS* (nem uniform racionális B-szplájn: az egymást követő görbeszegmenseknek nem egységnyi intervallum felel meg, és a súlyfüggvények két polinom hányadosai is lehetnek). Előnyük, hogy rugalmasan alakíthatók, ezért inkább a mérnöki tervezésben használják őket, a térinformatikai szoftverekben kevésbé (Szirmay-Kalos 2003).

A *Chaikin-algoritmussal* a vonallánc szögletességét csökkentjük: veszünk három, egymást követő csomópontot (P1, P2, P3), amely a vonallánc két szakaszát (P1–P2 és P2–P3) alkotja. Mindkét szakaszra, a P2 közelében beszúrunk két új pontot, és a P2-t töröljük. Ugyanezt végrehajtjuk a teljes vonalláncra. Ha ezt a folyamatot egymás után legalább háromszor ismételjük, általában kellően sima lesz az új vonallánc, megszűnik a vonallánc

„sarkossága" (Chaikin 1974, Riesenfeld 1975). Az így kapott görbék másodfokú B-szplájnok.

Bár a *polinomiális approximáció exponenciális kernellel* (PAEK) eredményeként kapott vonalak a Chaikin-algoritmushoz hasonlítanak leginkább, ezért ebbe a csoportba soroltam be, de a felhasznált matematikai módszer alapján mindhárom kategóriába illene. A görbe egyes szakaszain a csomópontok helye alapján átlagolt pontot számít ki egy, a Gauss-szűrőhöz hasonló, de azzal nem azonos kernellel (konvolúció). Ehhez másodfokú polinomokat is felhasznál (Bodansky et al. 2002).

Zajszűrésalapú simítás

A zajszűrés során a vonalláncot digitális jelnek tekintjük. A kisebb formákat reprezentáló ívek, görbületek a zajok, ezeket kiszűrjük, a nagyobb „trendeket", görbületeket megtartjuk. A *Fourier-transzformációt* például a digitális jelfeldolgozásban is használják. A szűrés időbeli folyamatokról szól, ez a szintvonalak esetén távolságot jelent, a frekvencia pedig térfrekvenciát. Mivel a szintvonalak általában nem ábrázolhatók függvényként (többértékű függvény nem fejthető Fourier-sorba), ezért Stegena a következőképpen járt el: egy síktartományt rendel minden szintvonalhoz (egy lépcső jön létre a szintvonal két oldala között, melynél a szintvonal egyik felén lévő magasságérték 1, a másikon 0), ezen szimmetrikus, kétváltozós szűrőt futtat. A 0,5 magasságú pontokat összeköti, ez lesz a szűrt vonallánc. A szűrők közül a felülvágó szűrőt ajánlja (Stegena 1970).

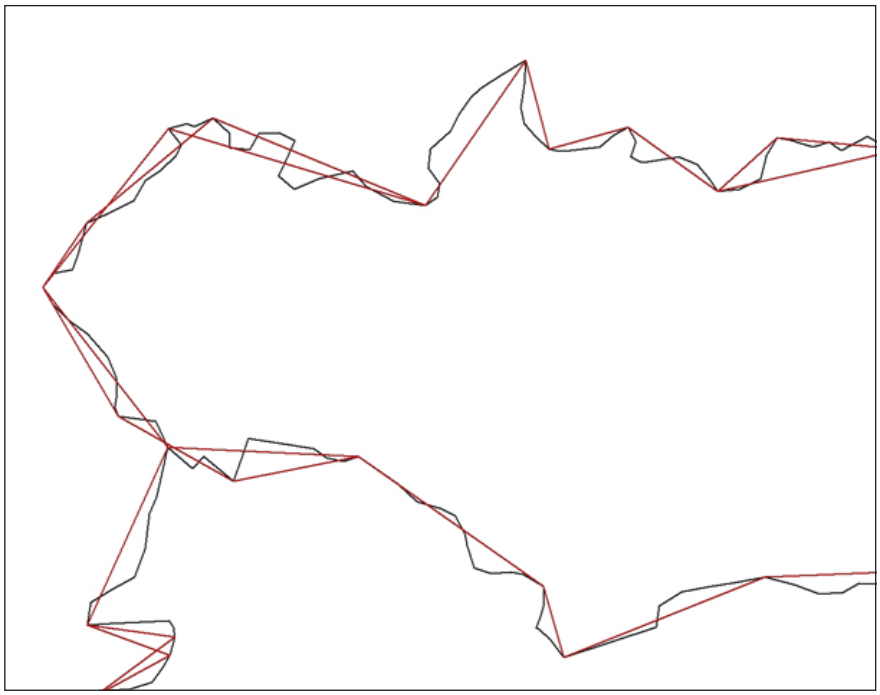
A másik lehetséges módszer a *„gyors" Fourier-transzformáció*. Többféle tudományban is gyakran használják zajszűrésre, ilyenkor a vonalakat szinusz- és koszinuszfüggvények sorára bontják fel. Boutoura (1989) bebizonyította, hogy önmagában az izovonalak X és Y koordinátáinak használata esetén nem működik jól, helyette inkább a vonalak meredekségét használta fel.

A wavelettranszformáció is egy spektrális felbontás, azonban nem szinusz- és koszinuszösszetevőkre bontjuk fel a jelet, ahogy azt a Fourier-transzformáció esetén tettük, hanem például különböző frekvenciájú négyszögjelekre (Li 2007). A számítás folyamata a gyakorlatban igen összetett, ezért a wavelet- és a Fourier-transzformációt kevésbé használják térinformatikai szoftverekben: eredményük és hatásuk a vonalláncokon a tanulmányok szerint egymáshoz igen hasonló; a tanulmányokban bemutatott ábrák alapján nem javaslom kartográfiai alkalmazásukat.

A *kígyóknak* (angolul: snakes) nevezett eljárás a számítás módszere miatt igen lassú, de egy kisebb, kevesebb töréspontból álló adathalmazon a gyakorlatban is használható eredményt ad. A módszer az energiaminimalizáción alapul: a vonal belső és külső energiáját kell a lehető legalacsonyabb szintre levinni. A függvénynek két, a felhasználó által választható paramétere van: a belső energia, amely a vonal alakját és karakterisztikáját írja le; és a külső energia, amely a vizsgálandó elem más térképi elemekre való hatását adja meg (Borkowski 1999).

3. Felületek egyszerűsítése és simítása

A felületek körvonala vonalegyszerűsítő és simító algoritmusokkal a vonalláncokhoz hasonlóan egyszerűsíthetők. Azonban ügyelni kell arra az egymással szomszédos poligonoknál, hogy a hézag- és átfedésmentességet megőrizzük. Ha egyenként egyszerűsítjük a felületeket, az egyszerűsítés a poligon első csomópontjától indul, amely két szomszédos elemnél nem esik egybe. Így ha más a kiindulópont, kissé más eredményt is kapunk generalizálásnál (1. ábra). A topológiailag helyes eredményt technikailag többféleképpen is elérhetjük. A kritikus pontoknak nevezzük azokat a pontokat, amelyekben legalább három vonalszakasz találkozik, ezeknek a pontoknak kell egy helyben maradniuk. A poligonokat felbontjuk – a kritikus pontokkal – szakaszokra. Ezeket a szakaszokat úgy kell egyszerűsíteni, hogy két poligon közötti közös szakaszt együtt, egy irányból kezelünk, a széleken levőket pedig önállóan generalizáljuk. A végén újra egyesítjük a felületeket. Ancsin Attila programtervező informatikushallgató diplomamunkájában ennél egyszerűbb megoldást mutatott be, amely jól működik pl. igazgatási egységek vagy felszínborítottság (megyék, országok) generalizálásánál. Az egyes poligonokat egyenként egyszerűsítette, majd minden hézagnál és átfedésnél megvizsgálta, hogy a keletkezett kis felület melyik eredeti poligonnal fed át nagyobb mértékben, és ahhoz kapcsolta (Ancsin 2016).



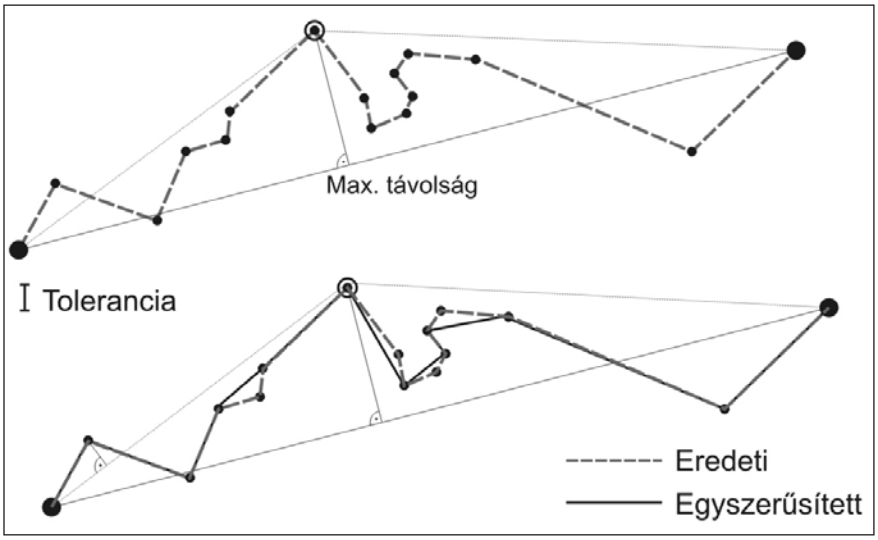
1. ábra. Határvonalak hibás egyszerűsítése (vörössel), ha a poligonokat külön-külön egyszerűsítjük, és nem vesszük figyelembe a kritikus pontokat

4. Vonalak generalizálása a gyakorlatban

Attól függően, hogy milyen térképi elemet akarunk automatizált módon generalizálni, figyelembe kell vennünk az elem geometriai tulajdonságait. Ebben a részben a legfontosabb térképi elemek generalizálását mutatom be a gyakorlatban.

Általános tapasztalatok a görbék generalizálásával a szint- és mélységvonalak példáján

A szint- és mélységvonalak generalizálása összetett feladat: nemcsak horizontális, hanem vertikális generalizálás[1] is szükséges (Márton 2012). A horizontális generalizálást tulajdonképpen két részre bonthatjuk: a nagy és közepes méretarányban végrehajtott generalizálásra, valamint a kis méretarányú térképek domborzatának generalizálására. Topográfiai térképeken (vagy olyan térképeken, amelyek alapnak a topográfiai térképek domborzatát használják, pl. tájfutó-, turistatérképek) a generalizálás szorosan szabályokhoz kötött, a szerkesztési utasításba foglalt: pl. mit kell ábrázolni, milyen sűrűn és hogyan kell felvenni a felező szintvonalakat, hol szükséges kiegészítő domborzatrajz (T.5. 1981). A kis méretarányú földrajzi, autós vagy egyéb tematikájú térképen a generalizálást jobban befolyásolják a térképszerkesztő földrajzi ismeretei. A horizontális generalizálás esetében többnyire az egyszerűsítő és simító eljárások kombinációját használhatjuk. Önmagában a vonalegyszerűsítés általában nem elég a megfelelő eredmény elérése érdekében, ugyanis túl szögletes, „sarkos" vonal keletkezik, ezért szükség lesz simításra is. Többféle módszerrel is kísérleteztem, de kartográfiai szempontból az egyik legjobb eredményt a Douglas–Peucker-algoritmussal történő egyszerűsítés (2. ábra), majd utána a Chaikin-algoritmussal való simítás után kaptam. A módszer előnye, hogy végrehajtható QGIS-ben, de ArcGIS-ben is. Az utóbbi szoftverben nem érhető el a Chaikin-algoritmus,



2. ábra. A Douglas–Peucker-algoritmus működése

de helyettesíthető a polinomiális approximációval. Emellett bizonyos méretarány-tartományokban a Douglas–Peucker-algoritmussal való egyszerűsítés és utána a Bézier-görbékkel történő simítás is jó eredményt hozott (kivitelezhető mindkét vezető térinformatikai szoftverben, de a QGIS-ben csak Hermite-görbéket tudunk használni).

Egy másik generalizálási módszer, amelyet Agárdi Norbert készített (2014), a lineáris regresszión alapul. Az algoritmus az egyes görbeszakaszokat egy regressziós egyenessel helyettesíti (ahol ez a meghatározott toleranciaérték belül található, ugyanahhoz a regressziós egyeneshez tartozik); a generalizálás mértéke a tolerancia növelésével emelhető. A regressziós egyenesek harmadfokú Bézier-görbeívek érintői lesznek, két végpontjuk két folytonosan csatlakozó görbeív kontrollpontja. Ennek segítségével kiszámíthatók a Bézier-görbeívek végpontjai. A Bézier-görbék esetén a nyílt görbék kezdő- és végpontjáig futó görbe nem számítható ki, kicsit rövidülhetnek a vonalak (3. ábra).

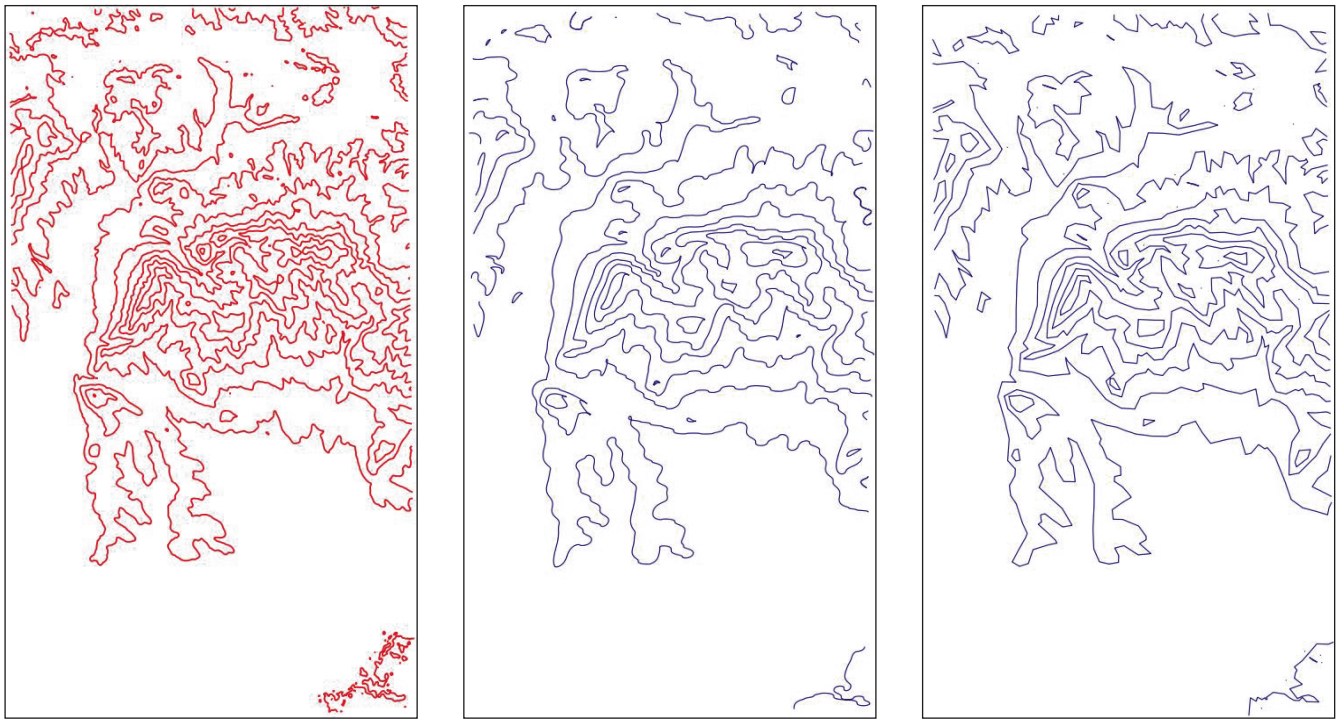
Mindkét módszernél keletkezhetnek bezárult, vagy önmagukat metsző görbeívek. Ezek többségének javítását érdemes lehet programozással megoldani. Mielőtt a vonalláncra görbét illesztene a program, megvizsgálja, hogy melyik polyline áll csupán két, vagy egymáson elhelyezkedő vertexből – ezek a záródott görbeívek lesznek. Önmetszés esetén többnyire a kezdőponthoz a második kontrollpont lesz közelebb, míg a végponthoz az első.

Görbeillesztés nélkül, a kiindulási és a célméretarány ismeretében használhatjuk a Li–Openshaw-féle raszteres-vektoros generalizáló módszert. A vonal mentén négyzeteket veszek fel úgy, hogy az első a vonal kezdőpontjának közepe lesz, és a többi négyzet az előző oldalát vagy sarokpontját érinti, a vonal irányának megfelelően. Minden négyzetben kiszámítottam a benne elhelyezkedő töréspontok átlagát, amelyek az új vonal csomópontjai lesznek. Ha egy vonal „elfér" egy négyzetben, akkor ez kisebb lesz a minimális méretnél, így ezt a vonalat elhagyja. Az algoritmus nem hoz létre önmetszéseket, azonban kissé szögletessé válhatnak a vonalak a kevés csomópont miatt (3. ábra). Az eredeti cikk alapján hoztam létre, és teszteltem ezt az algoritmust is. Simításként a Chaikin-algoritmust javaslom.

A Wang-algoritmus, vagy másképpen a kanyarulatok egyszerűsítése (ArcGIS Bend simplify) is részben eredményes lehet. Az egyszerűsítés során a kisebb kanyarokat, íveket szünteti meg először, ehhez kisebb toleranciaérték tartozik. Ha a toleranciát növeltem, egyre inkább több utólagos, kézi helyesbítést igényelt a rajz.

A Reumann–Witkam-algoritmus nem használható izovonalak egyszerűsítésére, mert utána hosszú egyenesekből álló vonalszakaszok keletkeznek, amelyeket nem lehet eltüntetni simítással.

[1] A vertikális generalizálás a megfelelő magasságok és mélységek kiválasztását jelenti, a horizontális generalizálás a szintvonalak „futásirányú" egyszerűsítését.



3. ábra. Az SRTM 90-ből generált kiindulási szintvonalrajz, a lineáris regresszióval egyszerűsített és a Li–Openshaw-féle egyszerűsítés, az egyszerűsítés után változatlan méretben (szintvonalak 100 méterenként)

A lokális eljárású módszerek a vonal néhány csomópontból álló szakaszát vizsgálják csak az egyszerűsítésnél, ezért önmagukban általában nem adnak jó eredményt, így inkább a feldolgozandó adatmennyiség előszűrésére használhatók.

A simító eljárásokkal is vegyesek a tapasztalataim. A görbeillesztések közül még eredményes lehet a Chaikin-algoritmus, kettő, vagy háromszori ismétléssel, ha nincsenek az állományban hosszú, egyenes vonalak. Ilyenkor előfordulhat, hogy csak a töréspontnál lesz kerek a vonal, de egyébként megőrzi szögletességét. Ehhez hasonló az ArcGIS-ben található PAEK (Polynomial Approximation with Exponential Kernel – Polinomiális approximáció exponenciális kernellel) nevű algoritmus.

A Boyle „előretekintő" simító hatású algoritmusát önállóan, többféle paraméterrel is kipróbáltam. A kapott eredmény igen hasonló a képszűréses módszerrel simított szintvonalakhoz. Boyle szerint a mélységvonalakhoz, tapasztalataim szerint magassági vonalak generalizálásához is alkalmazható. Utólagos kézi javítások itt is szükségessé válnak: annál több a javítás, minél nagyobb a toleranciaérték. Legfontosabb hibák: az önmagukba visszatérő szintvonalaknál (a térképkivágaton belül) a generalizálás kiinduló- és végpontja általában csúcsosan kapcsolódik össze; hasonló méretű völgyek ismétlődnek egymás után (hullámos felszín), a generalizált vonal a morfológiai forma ellentétjét veszi fel (pl. völgyből gerincforma lesz); kissé rövidülő völgyek vagy gerincek.

A súlyozott átlagokkal a szintvonalak „sarkossága" csökkenthető, de a töréspontok száma nem változik, ezért általában nem elegendő a szintvonalak generalizálásához.

Többféle algoritmus kombinálása

Korábban a térképszerkesztő által végzett generalizálás során, ha az alapanyag és a céltérkép között túl nagy volt méretarány-különbség, akkor a generalizálást több lépésben hajtották végre, ezen alapulnak például a topográfiai térképsorozatok. Például, ha 1:100 000-es alaptérképből szerettek volna egy 1:500 000-es térképet készíteni, szükséges volt egy köztes méretarányú térkép, pl. 1:250 000-es elkészítésére. A generalizálás automatizálásával a kiindulási és célméretarány között a különbség megnövekszik. A különbség nagysága leginkább a választott algoritmustól függ, de a szintvonalak sűrűsége (választott vertikális értékei) és a terület földrajzi jellege is befolyásoló tényező lehet. Ezentúl kipróbáltam, hogy mi történik, ha több, egymást követő lépésben alkalmazok automatizálási algoritmusokat: először ugyanazt az algoritmust hívtam meg a generalizált anyagon, második esetben egy másik algoritmus választottam. Azt vizsgáltam meg, befolyásolja-e az algoritmus választása a két esetet, illetve mennyire növelhető meg a méretarány-különbség a kiindulási és a célméretarány között. A különbségek többsége már az egyszerűsítésnél jelentkezik.

Vannak olyan algoritmusok – pl. Douglas–Peucker (röviden DP.) –, hogy ha két lépésben hajtom végre az egyszerűsítést, pontosan ugyanazt az eredményt kapom, mintha egy lépésben került volna sor rá.

A második esetben egymás után hívtam meg két, toleranciaértékként távolságot használó algoritmust. Ha toleranciaértékként ugyanazt a távolságot használtam, alig volt különbség az eredményben (DP, majd Lang) a két lépés folyamán. Ellenkező esetben, amikor a Lang-algoritmus volt az

első, és a DP a második, a különbség jelentősebb volt. Viszont ha a két végeredményt összevetem, némi különbség ugyan itt-ott jelentkezik, de nem számottevő. Az első esetben a rövidebb vonalak pontokká alakultak. Az előbbiekből következik, hogy ha önmagában csak a DP algoritmust használtam volna, hasonló eredményt kaptam volna. Levonható a következtetés, hogy két egyszerűsítő algoritmus használata általában nem növeli meg annyira méretarány-különbséget, hogy érdemes legyen ezt a módszert használni.

Óceáni területek megjelenítése

Az óceáni területeken a mélységvonalak generalizálása már kevésbé egyértelmű, mint a szárazföldön: nemcsak az 1960-as, '70-es, de még a '80-as években is jelentek meg olyan szakkönyvek (Koch 1960, Klinghammer–Papp-Váry 1983), sőt még manapság is térképek, amelyek szerint az óceáni területeknek kétszeresen generalizáltnak kell lenniük a szárazföldekhez képest. Ezt az óceánok kevésbé változatos domborzatával magyarázták, amely megállapítás azonban ma már nem állja meg a helyét. Manapság már rendelkezésünkre állnak domborzatmodellek, amelyekből kis méretarányban a batimetrikus izovonalak (izobátok) kinyerhetők, ezek vonalgeneralizálási módszerekkel (is) egyszerűsíthetők.

Vonalas vízrajzi elemek

Részben hasonlóak a szintvonalakhoz: görbével ábrázoljuk őket (kivételt képezhetnek a mesterséges csatornák, itt előfordulhatnak szögletes töréspontok). Két vízfolyásnál a torkolati pontot kell megőrizni. A kígyókalgoritmus jól használható vízrajzi elemek simításánál. A vízhálózat kategorizálása és a vízfolyások kiválasztása már nem a vonalgeneralizálás témakörébe tartozik. A vízrajz és a felszín generalizálása képszűrési módszerekkel is megvalósítható.

Partvonalak

A partvonalakat tekinthetjük a szintvonalak speciális esetének, a tenger szintjének, vagyis 0 méternek. Ezeket is görbeként értelmezzük, viszont ügyelni kell rá, hogy a félszigetek, fokok ne váljanak szigetekké, még kisebb méretarányokban sem (a vonalgeneralizálási algoritmusokkal a különválás elkerülhető, hiszen folytonos vonalakat egyszerűsítünk, viszont bezáródások előfordulhatnak). A partvonalak tagoltságát minden méretarányban meg kell őrizni: pl. Norvégia fjordjait; ilyenkor előfordulhat, hogy más generalizálási toleranciaértéket alkalmazunk az eltérő típusú partvonalak esetében. Tagolt partvonal esetén hamarabb bekövetkezhetnek önmetszések, ugyanis itt a vonallánc egyes részei közelebb kerülnek egymáshoz, ez utólagos kézi korrekcióra szorulhat.

Utak és vasutak

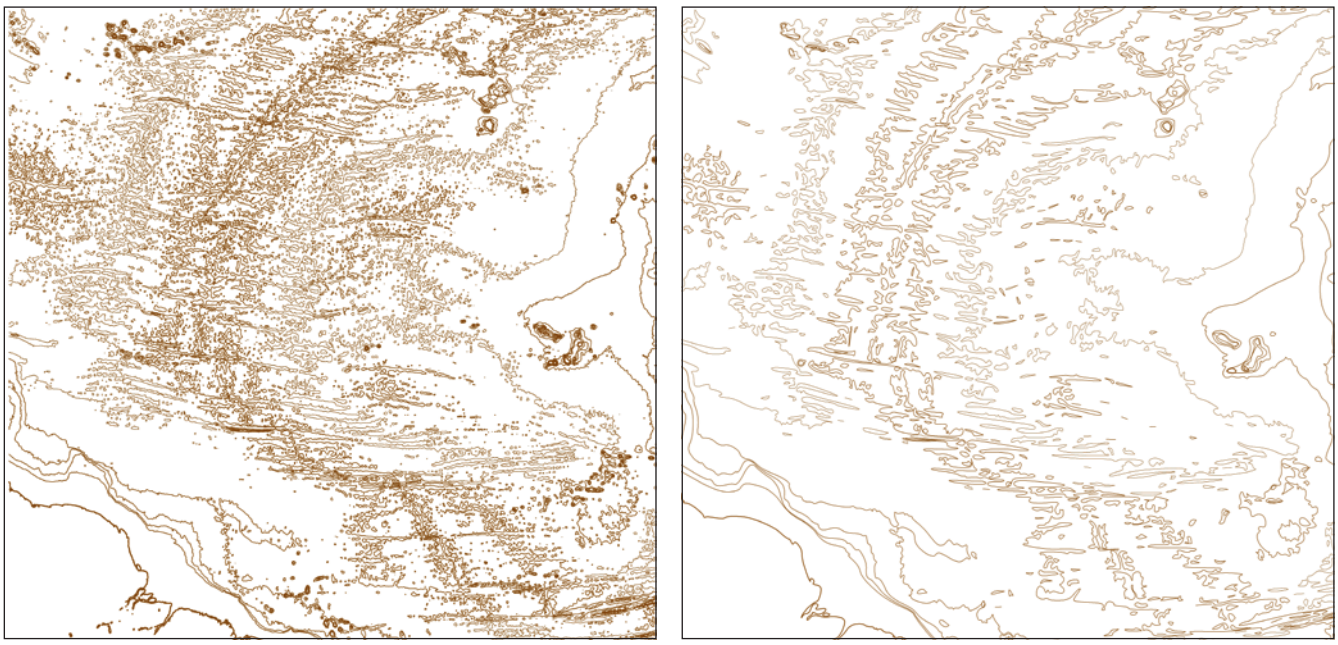
Az utaknál a legfontosabb generalizálási szabály a kiválasztás, ugyanakkor szükség van a vonalrajz részletességének csökkentésére is. A vasutaknál nem lehetnek „sarkosan" megtörő vonallánc ok, tehát simítás mindig szükséges.

Igazgatási egységek, határok, felszínborítottság és felületi vízrajzi elemek

Az igazgatási egységek egy nagyobb területet (magasabb szintű egységeket, országot) általában teljes mértékben lefednek (kivéve pl. természetvédelmi igazgatási egység). A közös határvonalaknál ügyelni kell a hézag- és átfedésmentességre. A határokat rajzban általában „szögletes" vonallánccal ábrázoljuk, kivéve, ha például természetes vízfolyáson fut – ekkor gyakran görbével helyettesítjük. A felszínborítottság és a felületi vízrajzi elemek ábrázolása az előzőhöz hasonló, ám itt már nagy szerepet kaphat az összevonás vagy az elhagyás is.

Épületek

Általánosságban elmondható, hogy a körvonaluk addig egyszerűsíthető, amíg téglalap vagy négyzet alakú épületet nem kapunk, amely csak az épület jelenlétét, vagy kiemelt voltát mutatja



4. ábra. Az óceánközépi hátság eredeti és generalizált (további utómunka nélküli) képe ugyan abban a méretben.

be (pl. közepes méretarányú térképeken). A célra speciális algoritmusok fejlesztése válhat szükségessé. Ritka kivételt képeznek a speciális alakú épületek (háromszög, ötszög). Emellett még az elhagyás, a hangsúlyozás és az eltolás is fontos szerepet játszik a generalizálásukban.

5. A generalizálásvizsgálatok eredményeinek értékelése

Az egyes algoritmusok kimenő adatainak értékelése szempontjából legfontosabbnak a vizuális összehasonlítást tartom, vagyis azt, hogy mennyire hasonlít az eredmény ahhoz, mintha „kézzel" végeztük volna el a generalizálás folyamatát. Lehetséges ugyan matematikai-statisztikai módszerekkel mérni az eredményeket a kiindulási anyag és a végeredmény között, de ez ellentmondásokhoz vezethet (pl. kedvező vizuális „megjelenés", de rosszabb statisztikai eredmény és fordítva). További nehézség az algoritmusok különbözőségéből fakad. Az sem helyes, ha hasonló toleranciaérték megadásával hasonlítjuk össze az algoritmusokat, mert más-más lehet a toleranciaérték dimenziója pl. hosszúság vagy terület. Eldöntendő kérdés az is, hogy az eljárás értékelésénél az egyes objektumokra gyakorolt hatást vesszük figyelembe, vagy az objektumok közötti viszonyokat is. Az eredmények értékeléséhez feltétlenül szükségesek a szerkesztő morfológiai ismeretei.

Egy algoritmus tervezésénél vagy használatánál felmerül a kérdés, hogy meddig érdemes az algoritmus által produkált hibákat programozással javítani pl. önmetszés, bezáródott ívek, elemek közötti viszonyok vizsgálata, metszések javítása, szintvonalfésülés, formák felismerése.

6. Következtetések

Az itt bemutatott algoritmusok mindegyike felhasználható a kartográfiai generalizálás során, ugyanakkor az egyes algoritmusok alkalmazhatósága erősen függ a térképi elem tulajdonságaitól. A generalizálási folyamat szubjektivitása miatt nagyon nehéz az algoritmust mindenre felkészíteni – csaknem lehetetlen. Sajnos, mint látszik, teljes egészében nem automatizálható a generalizálás, szükség van utólagos korrekciókra; az egyes speciális morfológiai formák (pl. fjordok, óceánközépi hátságok ábrázolása stb.) igényelhetik a szerkesztők javításait. Addig érdemes az algoritmust fejleszteni, amíg nem válik túl bonyolulttá, és a sok feltétel nem okoz ellentmondásokat, újabb hibákat. Emellett az egyes algoritmusok csak egy meghatározott méretarány-tartományokban használhatók jól: a tartomány felső határát az alapanyag minősége, szintvonalaknál a vertikális generalizálás mértéke, az egyszerűsítő és simító algoritmusok sorrendje, valamint a térképszerkesztő által vállalt kézi javítások mértéke és a kiegészítések (igazítás vízrajzi elemekhez és kiegészítő domborzatrajz elhelyezése) is befolyásolja.

Irodalomjegyzék

Agárdi Norbert: Automatizálási lehetőségek a tematikus kartográfiában. Doktori értekezés, Eötvös Loránd Tudományegyetem, Budapest, 2014. pp. 27–30.

The Anti-Grain Project, nyílt forráskódú C++ függvénykönyvtár. http://www.antigrain.com/research/bezier_interpolation/index.html [Utolsó elérés: 2016. 11. 02.]

Ancsin Attila:. Poligonok topológia-megőrző generalizálása. Diplomamunka. Eötvös Loránd Tudományegyetem, Budapest, 2016.

Eugene Bodansky–Alexander Gribov–Morakot Pilouk: Smoothing and compression of lines obtained by raster-to-vector conversion. In: Lecture Notes in Computer Science: Graphics Recongnition Algorithms and Applications. Vol. 2390, 2002. pp. 256–265.

Borkowski, A–Burghardt, D–Meier, S.: A fast snakes algorithm using the tangent angle function. International Archives of Photogrammetry and Remote Sensing 32 (Part 3-2W5), 1999. pp. 644–650.

Boutoura, Chryssoula: Line generalization using spectral techniques, Cartographica 26(3-4), 33–48, 1989.

Boyle, A. R.: The quantised line. Cartographic Journal Vol: 7 (2). 1970. pp. 91–94.

Chaikin George: An algorithm for high speed curve generation. Computer Graphics and Image Processing Vol. 3 (1974), pp. 346–349.

Deveau, Terry J.: Reducing the number of points in a plane curve representation. In: Auto-Carto VII. Washington, USA, 1985. pp. 152–160.

Dougenik, James: Whirpool A geometric processor for polygon coverage data. In: Auto-Carto IV. 1980. pp. 304–311.

Douglas, David–Peucker, Thomas: Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. The Canadian Cartographer 10(2), 1973. pp. 112–122.

Klinghammer István–Papp-Váry Árpád: Földünk tükre a térkép. Gondolat Kiadó, Budapest, 1983. Generalizálás: pp. 184–210.

Koch Nándor: A tenger. In: Tasnádi Kubacska András (szerk.): A Föld. Gondolat Kiadó, Budapest, 1960. pp. 211–239.

Kovács Emőd: Komputergrafika–Matematikai alapok. Elektronikus jegyzet. Eszterházy Károly Főiskola, Matematikai és Informatikai Intézet, 2011. 8. fejezet: Görbék megadása, 9. fejezet: B-spline görbe és felület.

Lang, T.: Rules for robot draughtsmen. Geographic Magazine 42(1), 1969. pp. 50–51.

Li, Zhilin: Algorithmic Foundation of Multi-Scale Spatial Representation. CRC Press, Taylor and Francis Group, Boca Raton USA, 2007. pp. 1–25, 57–72, 91–180.

Márton Mátyás: A Világtenger kartográfus szemmel. Eötvös Loránd Tudományegyetem, Informatikai Kar, Térképtudományi és Geoinformatikai Tanszék, Budapest, 2012. pp. 111–174.

NCGIA Core Curriculum Térinformatikai alapismeretek. M.F. Goodchild–K.K. Kemp (eredeti szerk.), Márton Mátyás–Paksi Judit–Márkus Béla (magyarul szerk.): 48. fejezet Vonalgeneralizálás. Erdészeti és Faipari Egyetem, Földmérési és Földrendezési Főiskolai Kar, Térinformatikai Tanszék, Székesfehérvár, 1994. pp. (48-1) – (48-10)

Ramer, Urs: An iterative procedure for the polygonal approximation of plane curves. Computer Graphics and Image Processing. 1(3), 1972. pp. 244–256.

Reumann, K.– Witkam, A. P. M.: Optimizing curve segmentation in computer graphics. In: Proceedings of International Computing Symposium, North-Holland Publishing Company. 1974. pp. 467–472.

Riesenfeld, R.: On Chaikin's algorithm. IEEE Computer Graphics and Applications 4, 3 (1975), pp. 304–310.

Rogers, Hartley Jr.: Theory of Recursive Functions and Effective Computability. The MIT Press. Cambridge, USA, 1987. p. 506.

Slocum, T. A. –McMaster, R. B. –Kessler, F. C. – Howard, H. H.: Thematic Cartography and geographic visualization. Pearson Prentice Hall, USA, 2005. Chapter 6. Scale and generalization pp. 103–120. és Chapter 15. Symbolizing Topography pp. 292–309.

Stegena Lajos: Térképi generalizálás és a szűrőelmélet. In: Térképi generalizálás (szerk: Stegena Lajos). Kézirat, Eötvös Loránd Tudományegyetem, Természettudományi Kar, Budapest, 1970. pp. 22–44. és 74–84.

Stegena Lajos–Klinghammer István–Füsi Lajos: Az automatizálás a kartográfiában II. Tankönyvkiadó Vállalat, Budapest, 1977. pp. 16–23. és 42–55.

Szirmay-Kalos László–Antal György–Csonka Ferenc: Háromdimenziós grafika, animáció és játékfejlesztés. Computerbooks, Budapest, 2003. 3. fejezet, pp. 54–69.

T.5. 1981. T.5. Útmutató az egységes országos térképrendszer 1:25 000–1:100 000 méretarányú levezetett topográfiai térképeinek tervezéséhez. MÉM OFTH Földmérési Főosztály, Budapest, 1981. pp. 68–75.

Tutić, Dražen–Lapaine, Miljenko: Area Preserving Cartographic Line Generalization. Cartography and Geoinformation, Vol. 8, No. 11, 2009. pp. 84–100.

Ungvári Zsuzsanna: Domborzatmodellek alkalmazása a térképkészítésben. In: Geodézia és Kartográfia 2015/11–12. pp. 23–28. (2015)
Ungvári Zsuzsanna: Az automatizált térképi generalizálás bevezetésének lehetőségei a szakmai, felsőfokú oktatásban: eddigi tapasztalatok, jövőbeli célok. In: Balázs Boglárka (szerk.): Az elmélet és a gyakorlat találkozása a térinformatikában VII. = Theory meets practice in GIS. Debreceni Egyetemi Kiadó, Debrecen, 2016. pp. 479–486.
Visvalingam, Maheswari – Whyatt, James D.: Line generalisation by repeated elimination of points. Cartographic Journal 30 (1). 1993. pp. 46–51.
Wang Zenshen – Müller, Jean Claude: Line generalization based on analysis of shape characteristics. Cartography and Geographic Information Systems 25 (1). 1998. pp. 3–15.

Summary

Automations in Line Generalization - Line Simplification and Smoothing

The cartographic generalization was always a time-consuming and labour-intensive process at analogue map making as well as at computer aided map editing. The cartographers apply the basic operators of generalization to keep the readability and the important information of maps. In geoinformatics software, it is possible to automatize this step, while using line simplification and smoothing algorithms to generalize the line and area features. In this article, these methods were collected, reclassified and expounded by the author.

Ungvári Zsuzsanna
tanársegéd
ELTE Térképtudományi és Geoinformatikai Tanszék
e-mail: ungvarizs@map.elte.hu

A Balaton klorofill-a eloszlásának monitorozása MODIS-adatok alapján

Koma Zsófia–Zlinszky András–Kern Anikó–Stephanie Palmer

Bevezetés

A vízminőség-távérzékelés (IOCCG 2000) lehetővé teszi a folyók, tavak és óceánok folyamatos, egész víztestre történő monitorozását a pontszerű mintavételeket kiegészítve térbeli mintázati információkkal a visszavert fénymennyiség és a víz biofizikai tulajdonságai között megállapított összefüggések alapján. Kezdetben a vízminőség-távérzékelés legfőbb feladata a fitoplankton-mennyiség kimutatása volt főként óceáni (ún. 1. típusú) vizek legfelső rétegeiben, mely elsődleges termelődésével a vízi táplálékláncalapja, így döntően befolyásolja a víz minőségét. Napjainkra a tudományág kiszélesedett, és az optikailag komplex (ún. 2. típusú) vízfelületeknek számító tengerek part közeli részére és tavakra is elkezdődött a különböző vízminőség-monitorozó eljárások fejlesztése, a már szélesebb hullámhossztartományban nagyobb felbontásban mérő műszerek alapján (Matthews 2011).

A Balatoni Limnológiai Intézet már a 2000-es években elkezdte az akkor rendelkezésre álló Landsat-adatok feldolgozását, ahol a lebegőanyag-tartalommal találtak összefüggéseket (Sváb et al. 2005), viszont a klorofill-a tartalom monitorozása csak speciális, szűkített időtartamokra működött (Sváb 2008). A projekt később folytatódott az ENVISAT- (ENVIronmental SATellite) műholdon lévő MERIS- (Medium Resolution Imaging Spectrometer) szenzor 2002–2012 közötti, 300 méter felbontású adatainak rendszeres feldolgozásával is. A MERIS-műszer adatait az in situ mérések segítségével sikeresen kalibrálták a Balaton klorofill-a tartalmának monitorozására (Palmer et al. 2015a), azonban a műholddal való nem várt kapcsolatvesztés miatt a MERIS-adatokra épülő hosszabb távú kutatásoknak nem volt további esélye. Ezzel szemben a Terra- és Aqua-műholdak fedélzetén elhelyezett multispektrális MODIS- (MODerate resolution Imaging Spectroradiometer) szenzorok mérései 2000 óta folytonosak, és adataikra számítani lehet a következő években is.

Bár a MODIS-adatok alkalmazása a 2. típusú vizek esetében nem elterjedt, mivel a csatornakiosztás és felbontás több más műszer esetén is ideálisabb, ugyanakkor jó néhány szakirodalmi példa bizonyítja, hogy a MODIS-szenzor adatai sikeresen alkalmazhatóak a tavak vízminőségének monitorozásában. Az afrikai Malawi-tó esetén 0,6 szórásnégyzet mellett meghatározták a tó klorofill-a tartalmát (Chavula et al. 2009), Kína területén pedig a Chaohu-tó esetén értek el szintén 0,6 szórásnégyzet pontosságú eredményt (Wu et al., 2009). Gower és munkatársai (2004) megállapították, hogy a MODIS alkalmazható vízminőség-monitorozó célokra komplex vizek esetén is. Ezenkívül a MODIS 1-es és 3-as csatornái alkalmazhatók a Sechi mélység (Wu et al., 2008) és a lebegőanyag (TSS, Total Suspended Solids, Miller és Mckee 2004) mennyiségének becslésére is.

A kutatás célja a Balaton klorofill-a (chl-a) mennyiségének becslésére alkalmas módszer fejlesztése, és működésének vizsgálata MODIS-adatok felhasználásával. A téma jelentősége, hogy a MODIS-adatok folytonos adatforrást jelentenek 2000-től egészen napjainkig, így lehetővé teszik a tó hosszabb távú ökológiai célú vizsgálatát, és a klorofill-a, illetve lebegőanyag-tartalom térképezését, mely egyéb biológiai célú kutatások kiindulópontja lehet.