

SZINTVONALAK AUTOMATIZÁLT, HORIZONTÁLIS GENERALIZÁLÁSA AZ ELMÉLETBEN ÉS A GYAKORLATBAN

Vonalegyszerűsítő és simító eljárások

Ungvári Zsuzsanna

Eötvös Loránd Tudományegyetem, Térképtudományi és Geoinformatikai Tanszék,
ungvarizs@map.elte.hu

Összefoglalás

A földfelszín izovonalas leképezését nagy-, közepes- és kisméretarányban egyaránt gyakran használjuk térképeken. Ahogy csökken a méretarány, egyre kevesebb segédlet vagy utasítás áll rendelkezésünkre a kartográfiai generalizálás objektív végrehajtásához, egyre jobban érvényesül a szerkesztő szubjektív szemlélete, és morfológiai ismeretei. A folyamat munkaigényessége miatt manapság egyre gyakrabban alkalmazunk számítógépes algoritmusokat, amelyekkel egy-egy izovonal – szintvonal vagy bathimetrikus vonal – részletessége csökkenthető: azonban ismerni kell ezek tulajdonságait az optimális eredmény elérése céljából. Jelen fejezetben bemutatom és röviden jellemzem a legfontosabb vonalegyszerűsítő és simító eljárásokat, különös tekintettel azokra, amelyek szoftverekben is megtalálhatók. Az angolszász szakirodalomban két, fontosabb rendszerezés született, ezeket az általam ismert algoritmusokkal kibővítettem és rendszereztem. A vonalgeneralizálás két nagy csoportra bontható: az egyszerűsítés és a simítás. Ezeken belül az eljárás működésétől függően két illetve három kategóriát különböztetünk meg.

A cikk második részében az algoritmusok gyakorlati alkalmazását mutatom be. A térinformatikai szoftverekben előforduló, valamint az ismert cikkek alapján ígéretes, jó eredményt hozó algoritmusokhoz készített programokból generált eredmények várható eredményeit és az ehhez kidolgozott értékelési folyamatot ismertetem.

Abstract

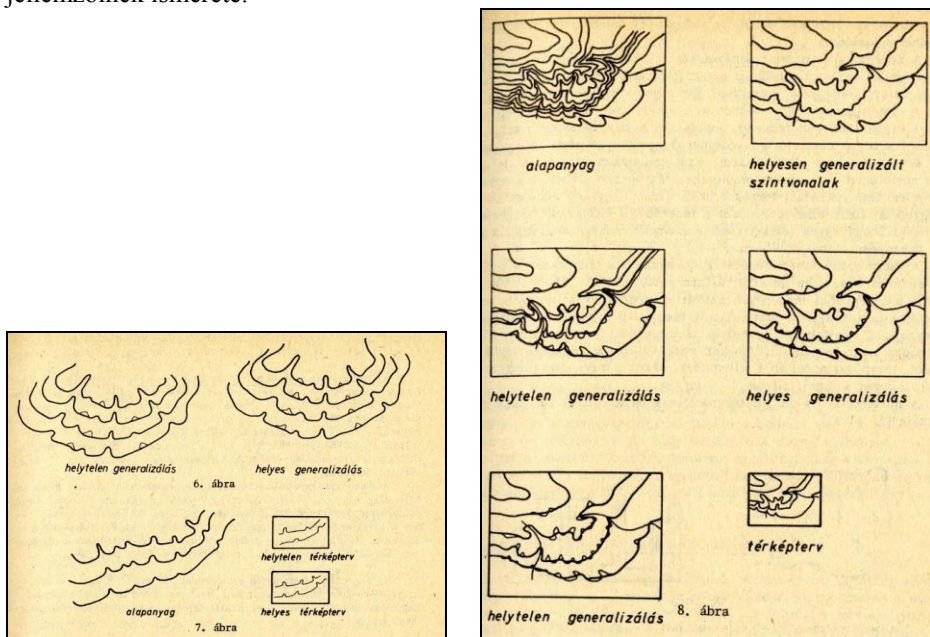
The representation of Earth surface by isolines – such as contour and bathymetric lines – is often used in multiscale maps. At topographic map scale, several directions and rules for generalization are available, but there are just a few of them for smaller scale maps. The subjective viewpoint of the cartographer is much stronger at these maps. Therefore, cartographers need thorough geographic knowledge to increase the objectivity of maps. In addition, generalizing contours needs a lot of human work, thus computer algorithms took over this task in last years. These algorithms can be classified in two main categories according to their function: line simplification routines eliminate the unnecessary vertex of polylines, and smoothing reduces the sharp peaks or broken polylines can be transformed to real curves. In this paper, these algorithms are introduced detailed into the Hungarian scientific literature for the first time. The new classification considers the previous important systems like McMaster and Li, and it is based on them. Line simplification has subgroups as scale-dependent and scale-independent routines, and smoothing includes curve fitting, weighted averages algorithms and digital filtering methods.

In the second section of this article, the author shows how these algorithms work in the practice mainly focused on routines, which can be found in geoinformatics software or in the program written by the author to test them. Finally, an evaluation sheet was created to assess the efficiency and the result of generalization algorithms.

1. Bevezetés

A szintvonal- és a batimetrikus vonalrajz a földfelszín két dimenzióban leképezett mása. A szintvonalas ábrázolás technikájával több szakirodalom is foglalkozik, így RÁTÓTI 1979-ben és KLINGHAMMER–PAPP-VÁRY 1983-ban is.

Kevesebbet olvashatunk viszont a helyes generalizálásáról. Habár találtam néhány ábrát pl. Az automatizálások a kartográfiában II. című könyvben (STEGENA, KLINGHAMMER és FÜSI, 1977), a kartográfiai tankönyvben (KISS PAPP, 1983), valamint az EOTR T.5. szabályzatában is (T.5., 1981), de részletes útmutatók – főleg kisméretarányú térképekhez – nem állnak rendelkezésünkre. A generalizált rajz lényege, hogy a terep legfontosabb idomait adja vissza, Márton Mátyás szerint hiteles „karikatúrája” legyen (IMHOF, 1965, STEGENA, 1970 és MÁRTON, 2012). Mintául szolgálhatnak korábbi, kézzel készített térképek a területről, illetve a terület földrajzi jellemzőinek ismerete.



1. ábra Példa a helyes és helytelen szintvonal-generalizálásra a *Kartográfia és térképszoftorítás* című tankönyvben (KISS PAPP, 1983)

Az automatizált generalizáláshoz vonalegyszerűsítő és simító algoritmusokat használtam, ezért a következő részben röviden ismertetem a rendelkezésünkre álló eljárásokat.

2. A vonalegyszerűsítés és simítás elmélete

A hagyományos térképek rajzolásánál akár manuálisan, akár térképrajzoló szoftverben a térképszerkesztő „fejében” zajlik a generalizálás: csökkenti a kanyarulatok számát, egyenesíti a vonalat. Ha ezt a folyamatot automatizálni szeretnénk, 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 szerkezete egyszerűsödik. Simítás során a csúcokat, szögleteket távolítjuk el, ezáltal kerekébbé, „simábbá” válik a vonallánc: új pontokat számítunk ki. Li (2007) ezzel szemben egy újabb csoportot is bevezetett, ezek a generalizáló algoritmusok. Ezekben a méretarány paraméterként megadható. Véleményem szerint ugyan mindkét csoportosítás helytálló, de a generalizáló algoritmusok besorolhatók a vonalegyszerűsítők közé, annak egy alcsoportját képezik.

2.1 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 méretarány-függő eljárások. A méretarány-független algoritmusoknál nem adható meg sem a kiindulási, sem a célméretarány. 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 tolerancia érték, ez lehet távolság, terület, szög, számosság. A teljesség igénye nélkül, a kartográfiai gyakorlatban eredményesen használható, vagy a térinformatikai szoftverekbe épített algoritmusokat tekintem át, kitérve legfontosabb jellemzőikre.

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 Jenk 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 szomszédosságát 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-rutin az egyetlen választható paraméter az éppen vizsgált

három pont által bezárt háromszög területe, vagyis a hatékony terület (VISVALINGAM és 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 és 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 és MÜLLER, 1998). Ide tartozik még a lineáris regresszió alapuló generalizálás is (AGÁRDI, 2014).

A globális eljárások a teljes vonalláncot, 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 és 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 tolerancia érték, 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.

A méretarány-függő eljárások közé sorolom a Li által generalizálónak nevezett algoritmusokat, amelyekben meghatározható a kiindulási és célméretarány is. Perkal ötlete az automatizálási gyakorlatba nem került át. Ennek lényege, hogy a vonal mentén mindkét oldalon köröket helyezünk el oda, ahol a vonalnak inflexiós pontja van. A kör vonalláncot 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 módszer, 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. Bemeneti paraméterként csak a célméretarány nevezőjét kell megadni (QGIS-es verzió), ehhez tartozik egy tapasztalati úton meghatározott minimális távolság (0,05 mm), amelynél a négy pontból képzett három szakasz összhosszának kisebbnek kell lennie ennél a számnál. A simítást csak ott hajtja végre, ahol túl éles szög keletkezett, az egyszerűsítésnél is alkalmazott a területmegőrző módszer ellentétével (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.

2.2 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 „előretékintő” algoritmus (*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 során a vonalláncot görbékkel helyettesítjük. Mivel a térinformatikai szoftverek vagy fájlformátumok többségében nem támogatják a görbék leírását, ezért ezeket *polyline*-okkal 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-görbéiv futásvonala a két végpont és a két vezérlő v. más néven kontrollpont koordinátájának ismeretében írható le. Ha egy „törött” vonalat helyettesíték görbével, akkor az egyes görbévek érintői a végpontokban párhuzamosak és folytonosak, így biztosítható 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 és 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 vonalláncre. Ha ezt a folyamatot egymás után legalább háromszor ismétljük, általában kellően sima lesz az új vonallánc, megszűnik a vonallánc „sarkossága” (CHAIKIN, 1974 és 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).

Zajszűrés alapú 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 koszinusz függvények sorára bontják fel. Boutoura bebizonyította, hogy önmagában az

izovonalak X és Y koordinátáinak használata esetén nem működik jól, helyette inkább a vonalak meredekségét használta fel (BOUTOURA, 1989).

A wavelet-transzformáció is egy spektrális felbontás, azonban nem szinusz és koszinusz összetevőkre 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 javasolom 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 energia-minimalizáción 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. A szintvonalak horizontális generalizálása a gyakorlatban

A szintvonalak generalizálása összetett feladat: nemcsak horizontális, hanem vertikális generalizálás is szükséges. A horizontális generalizálást tulajdonképpen két részre bonthatjuk: a nagy- és közepes méretarányban végrehajtott generalizálásra, valamint a kisméretarányú térképek domborzatának generalizálására. Topográfiai térképeken (vagy olyan térképeken, amely alapnak a topográfiai térképek domborzatát használják, pl. tájfutó, turista térképek) a generalizálás szorosan szabályokhoz kötött, a szerkesztési utasításba foglalt: pl. mit kell ábrázolni, milyen sűrűn és hogyan kell felvenni a felezőszintvonalakat, hol szükséges kiegészítő domborzatrajz (T.5., 1981). A kis méretarányú földrajzi, autós vagy egyéb tematikájú térképen a generalizálást jobban befolyásolják térképszerkesztő földrajzi ismeretei. A horizontális generalizálás vizsgálatát ezért kétféle méretarány-tartományban végeztem, de jelen cikkben csak a kis méretarányban szerzett tapasztalataimat írom le. Itt az algoritmusokat SRTM90-es modellből generált szintvonalakon teszteltem. A következőkben az egyes algoritmusok lehetséges használatát és a várható eredmények minőségét értékelem.

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. Többféle módszerrel is kísérleteztem, de kartográfiai szempontból az egyik legjobb eredményt a Douglas–Peucker-algoritmussal történő egyszerűsítés, 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), de helyettesíthető a polinomiális approximációval). Emellett bizonyos méretarány-tartományokban a Douglas–Peucker-algoritmussal való egyszerűsítés és utána Bézier-görbékkel történő simítás is jó eredményt hozott (kivitelezhető mindkét vezető térinformatikai szoftverben, de a QGIS-ben csak Hermite-görbékkel tudunk használni). A generalizálás menete az utóbbi módszerrel a következő:



2. ábra Douglas–Peucker-algoritmus kifejezetten alkalmas a „szögletes” formák megőrzésére, pl. óceánközépi hátságoknál akár 20-25 milliós méretarányban is

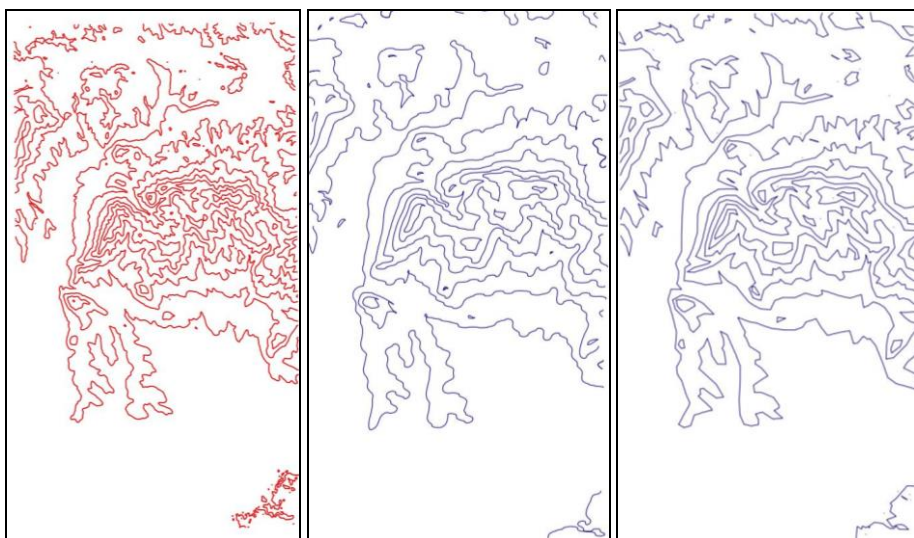
Szükség van egy szintvonalrajzot tartalmazó állományra, ezt generálhatjuk akár SRTM domborzatmodellből is. A felhasználó választja ki a generalizálás mértékét, adja meg a Douglas–Peucker-függvény paramétereit. Ha a forrásadatokat földrajzi koordinátaként tároljuk, akkor fokban; egyébként metrikus rendszerben (angolszász mértékegységekben is lehet) értelmezzük, ettől is függ a paraméter nagysága. Minél nagyobb ez a szám, annál erősebb mértékű lesz a generalizálás. Az adatsor beolvasása után vonallánccokra bontjuk az állományt. A vonallánconként megyünk végig az állományon, a Douglas–Peucker-algoritmus (rövk.: DP) rekurzívan hívja újra önmagát. Két függvényből áll, az egyikkel kiszámítja a merőleges távolságot az aktuális pontok között, míg a másik a toleranciaértékkel való összehasonlításért, illetve az új vonallánc töréspontjainak mentéséért felelős. Ha befejezte az egyszerűsítést a vonalláncon, a simítás következik. Egymáshoz folytonosan csatlakozó, harmadfokú Bézier-görbe íveket illeszt úgy, hogy a végpontok átmennek a vonallánc töréspontjain, a két kontrollpontot pedig interpolálja. Az eredményt olyan állományban érdemes tárolni, amely felismeri a Bézier-görbéket, pl. SVG (*Scalable Vector Graphics*), de lehetséges a görbe nyomvonalát megfelelő töréspontsűrűséggel pl. *shapefile*-ba is kiírni.

Másik hatékony generalizálási módszer, amelyet Agárdi Norbert készített (AGÁRDI, 2014), a lineáris regresszió módszerén alapul. Az algoritmus az egyes görbeszakaszokat egy regressziós egyenessel helyettesíti (ahol ez a meghatározott toleranciaértéken 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.

Mindkét módszernél keletkezhetnek bezárult, vagy önmagukat metsző görbeívek. Ezek többségének javítását érdemes lehet programozással megoldani. Mielőtt a vonalláncre görbét illesztene a program, megvizsgálja, hogy melyik *polyline* áll csupán két *vertex*-ből – ezek a záródott görbeívek lesznek.

Görbeillesztés nélkül, a kiindulási és a célméretarány ismeretében használhatjuk a Li–Openshaw-módszerét raszteres-vektoros módban. 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, amely az új vonal csomópontjai lesznek. Ha egy vonal „elfér” egy négyzetben, akkor ez kisebb lesz a minimális méretnél, így ezt elhagyja. Az algoritmus nem hoz létre önmetszéseket, azonban kissé szögletessé válhatnak a vonalak a kevés csomópont miatt. A szögletesség pl. a Chaikin-algoritmussal eltüntethető. Az eredeti cikk alapján hoztam létre, és teszteltem ezt az algoritmust is (3. ábra).



3. ábra Az SRTM90-ből generált kiindulási szintvonalrajz, a lineáris regresszióval egyszerűsített és a Li–Openshaw-féle egyszerűsítés

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 tolerancia érték 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 egyenes szakaszokból álló vonalszakaszok keletkeznek, amelyeket nem lehet eltüntetni simítással.

A lokális eljárású módszerek a vonal néhány csomópontból álló szakaszát vizsgálják csak az egyszerűsítésnél, ezért önmagukban általában nem adnak jó eredményt, ezért 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*) nevű algoritmus.

A Boyle „előrettekintő” simító hatású algoritmusát önállóan, többféle paraméterrel is kipróbáltam. A kapott eredmény igen hasonló a képszűrési módszerrel simított szintvonalakhoz. Boyle szerint a mélységvonalakhoz, tapasztalataim szerint magassági vonalak generalizálásához is alkalmazható. Utólagos kézi javítások itt is szükségessé válnak: annál több javítás, minél nagyobb a tolerancia-érték. 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.

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. 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, tolerancia értéként 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 nyújtja meg annyira a kiindulási és célméretarány közti különbséget, hogy érdemes legyen ezt a módszert használni.

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 és KLINGHAMMER-PAPP-VÁRY, 1983), sőt még manapság is térképek, amelyek szerint az óceáni területek kétszeresen generalizálnak kell lenniük a szárazföldekhez képest. Ezt az óceánok kevésbé változatos domborzatával magyarázták, ami azonban nem igaz. Manapság már rendelkezésünkre állnak domborzatmodellek, amelyekből kis méretarányban a batimetrikus izovonalak (izobátok) kinyerhetők.

4. A generalizálás-vizsgálatok 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 tolerancia-érték 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 geomorfológiai ismeretei.

Az egyszerűsítő és a simító algoritmusok értékelését Li is külön tárgyalja (LI, 2007), ezt szintvonalak esetén nem tartom szükségesnek, mert nem az algoritmusokat, hanem az eredményüket kell elemezni, többnyire egymás után használva az egyszerűsítést és simítást. Az értékelési folyamatot és kritériumokat a következőképpen határozom meg az automatizált generalizálásnál:

1. Vizuális értékelés: megőrizte-e az eredeti vonal karakterisztikáját? (A generalizálás minősége.)
2. A megadott paraméterekkel a célméretarányban használható eredményt kaptunk-e? Milyen mértékű utólagos korrekció szükséges?
3. Keletkeztek-e önmetszések, összezárult vonalláncok, túl kicsi objektumok? Ha igen, milyen mértékben és van-e jellegzetes előfordulásuk (LI, 2007)? (Programozással javíthatók.)
4. Keletkeznek-e metszések objektumok között? (Az algoritmus kiegészítése lehet szükséges.)
5. A vonal megváltozott futása miatt a generalizálást már korábban is megpróbálták számszerűsíteni. Stegena szerint egy szintvonal maximális helyzetváltozása a vízszintes irányú bizonytalanság minimumának kétszerese

lehet, ettől csak úgy lehet eltérni, hogy az új izovonal minél jobban megőrizze a felszín formakincsét. A vízszintes bizonytalanságot a következőképpen definiálta Koppe:

$$m = \pm(k \cdot M + c \cdot \text{ctg} \alpha),$$

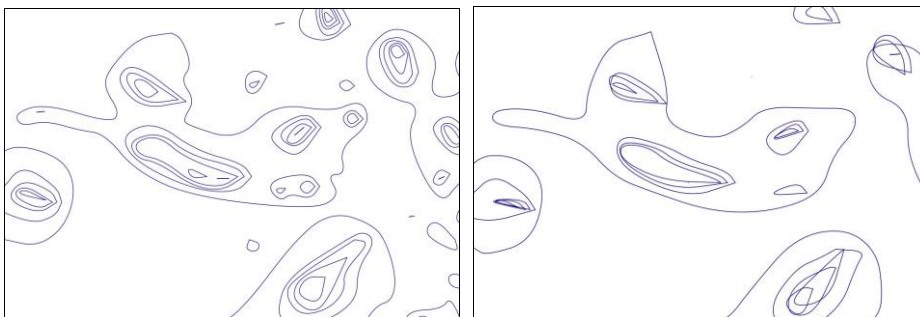
ahol az α a lejtőszög, $c=0,1$ m a magasságmeghatározási hiba és $k=0,0002$ m a helyzetkiolvadási hiba és M a méretarányszám. Vagyis pl. 1:50 000 térképen 1° -os lejtőszögnél kb. 15 méter, ennek kétszerese kb. 30 m, 45° -os lejtőnél ez 10 m és 20 m (STEGENA, 1970). Egyes térképtípusoknál nagyobb eltérés is megengedhető, pl. eltolható a szintvonal a szomszédos szintvonal fele távolságáig.

6. Megnéztem, hogy keletkeztek-e új pontok, vagy csak csökkentettük az eredeti csomópontszámot? Véleményem szerint, a mérőszámok önmagukban nem adnak tájékoztatást az eredmény minőségéről, ennél sokkal fontosabb a vizuális kiértékelés.)

7. Melyik szoftverben lévő algoritmust használtuk? Az algoritmus mennyire könnyen implementálható szoftverekbe, mennyire gyors és hatékony?

8. Ha a mintaterületen létezik topográfiai térképszelvény, vagy atlasz, akkor a régi (kézi) és új (automatizálással készített) domborzatrajz összehasonlítása milyen eredményt hoz?

A generalizálás-vizsgálatokat háromféle módszerrel próbáltam ki: ezek az egyszerűsítés DP-algoritmussal, majd simítás a Chaikin-féle módszerrel; lineáris regresszió módszere; és a Li–Openshaw-féle raszteres-vektoros módszerű generalizálás. Ezek alapján az általánosságban levonható következtetések: mindegyik algoritmusnál megállapítható egy méretarány-tartomány, ahol vizuálisan szép eredményt ad, ez a tartomány a DP-algoritmus alkalmazásánál a legszűkebb. Az 5. pontban leírtak szerint a térképen helyenként kb. max. 1 mm-es eltérések észlelhetők. Ennek mértéke és gyakorisága függ a választott algoritmustól is; legtöbbször a szűk völgyeket vagy gerinceket ábrázoló kanyarulatoknál fordul elő. A 8. pontban említett összehasonlítás kivitelezése a legnehezebb: nem biztos, hogy a célméretarányban létezik megfelelő térkép (rétegszínezéses vagy szintvonalas ábrázolású, és ez megfelelő „színvonalú” is). 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, szintvonalfűsülés, formák felismerése (4. ábra).



4. ábra *Eltérő módszerrel egyszerűsített batimetrikus vonalrajz, amely a tengerfenéki kúpokat ábrázol, a második esetben több korrekció szükséges*

5. Következtetések és kitekintés

Az itt bemutatott algoritmusok mindegyike felhasználható a kartográfiai generalizálás során, ugyanakkor nem mind alkalmas izovonalak egyszerűsítésére. 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, 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. Az egyes algoritmusok felhasználhatósági méretarány-tartománya erősen függ a terület földrajzi terület jellegétől is. Vizsgálódásaim bebizonyították, hogy más-más algoritmus alkalmasabb például a magashegységek vagy fjordok valamint az óceánközépi hátságok ábrázolására.

Az elmúlt évtizedben az interneten szabadon elérhető váltak különféle felbontású domborzatmodellek, ezekből nyertem a vizsgálataimhoz használt szintvonalakat is. Ugyanakkor lehetséges ezek egyszerűsítése is különféle módszerekkel pl. képszűréssel. Az eredmény vizsgálataim alapján itt sem tökéletes: ellenben, ha figyelembe vesszük a vízrajzi elemeket, akkor elkerülhetjük, hogy pl. a folyók „felfelé” folyjanak.

Irodalomjegyzék

- AGÁRDI, N., 2014. Automatizálási lehetőségek a tematikus kartográfiában. Doktori értekezés, Budapest: Eötvös Loránd Tudományegyetem, pp. 27–30.
- AGG REFERENCE MANUAL, 2007. The Anti-Grain Project, nyílt forráskódú C++ függvénykönyvtár. [online] Elérhetőség: http://www.antigrain.com/research/bezier_interpolation/index.html [2016. november 2.]
- BODANSKY, E., GRIBOV, A. és PILOUT, M., 2002. Smoothing and compression of lines obtained by raster-to-vector conversion. In: *Lecture Notes in Computer Science: Graphics Recognition Algorithms and Applications*. Vol. 2390, pp. 256–265.
- BORKOWSKI, A., BURGHARDT, D. és MEIER, S., 1999. A fast snakes algorithm using the tangent angle function. *International Archives of Photogrammetry and Remote Sensing*, 32 (Part 3-2W5), pp. 644–650.
- BOUTOURA, C., 1989. Line generalization using spectral techniques, *Cartographica*, 26(3-4), pp. 33–48.
- BOYLE, A. R., 1970. The quantised line. *Cartographic Journal*, Vol: 7 (2). 1970. pp. 91–94.
- CHAIKIN, G., 1974. An algorithm for high speed curve generation. *Computer Graphics and Image Processing*, Vol. 3 (1974), pp. 346–349.
- DEVEAU, T. J., 1985. Reducing the number of points in a plane curve representation. In: *Auto-Carto VII*. Washington, pp. 152–160.
- DOUGENIK, J., 1980. Whirpool A geometric processor for polygon coverage data. In: *Auto-Carto IV.*, pp. 304–311.

- DOUGLAS, D. és PEUCKER, T., 1973. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *The Canadian Cartographer* 10(2), pp. 112–122.
- IMHOF, E., 1965. *Kartographische Geländedarstellung*. Berlin: Walter de Gruyter & Co., pp. 133–172, 346–359.
- KISS PAPP, L., 1983. *Kartográfia és térképsokszorosítás*. Budapest: Tankönyvkiadó Vállalat, pp. 9–34.
- KLINGHAMMER, I. és PAPP-VÁRY, Á., 1983. *Földünk tükre a térkép*. Budapest: Gondolat Kiadó, pp. 184–210.
- KOCH, N., 1960. A tenger. In: TASNÁDI, K. A. (szerk.): *A Föld*. Budapest: Gondolat Kiadó, pp. 211–239.
- KOVÁCS, E., 2011. *Komputergrafika–Matematikai alapok*. Elektronikus jegyzet. Eger: Eszterházy Károly Főiskola, Matematikai és Informatikai Intézet, 8. fejezet: Görbék megadása, 9. fejezet: B-spline görbe és felület.
- LANG, T., 1969. Rules for robot draughtsmen. *Geographic Magazine* 42(1), pp. 50–51.
- LI, Z., 2007. *Algorithmic Foundation of Multi-Scale Spatial Representation*. Boca Raton USA: CRC Press, Taylor and Francis Group, pp. 1–25, 57–72, 91–180.
- MÁRTON, M., 2012. *A Világtenger kartográfus szemmel*. Budapest: Eötvös Loránd Tudományegyetem, Informatikai Kar, Térképtudományi és Geoinformatikai Tanszék, pp. 111–174.
- NCGIA CORE CURRICULUM, 1994. *Térinformatikai alapismeretek*. GOODCHILD, M. F. és KEMP, K. K. (eredeti szerk.), MÁRTON, M., PAKSI, J. és MÁRKUS, B. (magyar szerk.): 48. fejezet Vonalgeneralizálás. Székesfehérvár: Erdészeti és Faipari Egyetem, Földmérési és Földrendezési Főiskolai Kar, Térinformatikai Tanszék, pp. (48-1) – (48-10).
- RAMER, U., 1972. An iterative procedure for the polygonal approximation of plane curves. *Computer Graphics and Image Processing*. 1(3), pp. 244–256.
- RÁTÓTI, B., 1979. *Gyakorlati térképszerkesztés, térképtervezés*. Budapest: Kartográfiai Vállalat, pp. 127–137.
- REUMANN, K. és WITKAM, A. P. M., 1974. Optimizing curve segmentation in computer graphics. In: *Proceedings of International Computing Symposium*, North-Holland Publishing Company, pp. 467–472.
- RIESENFELD, R., 1975. On Chaikin's algorithm. *IEEE Computer Graphics and Applications* 4, 3 (1975), pp. 304–310.
- ROGERS, H. Jr., 1987. *Theory of Recursive Functions and Effective Computability*. Cambridge, USA: The MIT Press, p. 1.
- SLOCUM, T. A., McMASTER, R. B., KESSLER, F. C. és HOWARD, H. H., 2005. *Thematic Cartography and Geographic Visualization*. USA: Pearson Prentice Hall, 6. fejezet: Scale and generalization, pp. 103–120. és 15. fejezet: Symbolizing Topography, pp. 292–309.
- STEGENA, L., 1970. *Térképi generalizálás és a szűrőelmélet*. In: *Térképi generalizálás* (szerk: STEGENA, L.), kézirat. Budapest: Eötvös Loránd Tudományegyetem, Természettudományi Kar, pp. 22–44. és 74–84.
- STEGENA, L., KLINGHAMMER, I. és FÜSI, L., 1977. *Az automatizálás a kartográfiában II*. Budapest: Tankönyvkiadó Vállalat, pp. 16–23. és 42–55.
- SZIRMAY-KALOS, L., ANTAL, GY. és CSONKA, F., 2003. *Háromdimenziós grafika, animáció és játékfejlesztés*. Budapest: Computerbooks, 3. fejezet, pp. 54–69.
- T.5., 1981. *Útmutató az egységes országos térképrendszer 1:25 000–1:100 000 méretarányú levezetett topográfiai térképeinek tervezéséhez*. Budapest: MÉM OFTH Földmérési Főosztály, pp. 68–75.
- TUTIC, D. és LAPAINE, M., 2009. Area Preserving Cartographic Line Generalization. *Cartography and Geoinformation*, Vol. 8, No. 11, pp. 84–100.

VISVALINGAM, M. és WHYATT, J. D., 1993. Line generalisation by repeated elimination of points. *Cartographic Journal* 30 (1), pp. 46–51.

WANG, Z. és MÜLLER, J. C., 1998. Line generalization based on analysis of shape characteristics. *Cartography and Geographic Information Systems* 25 (1), pp. 3–15.

