

SZAKDOLGOZAT

Kocsis Viktória

2015

EÖTVÖS LORÁND TUDOMÁNYEGYETEM

TERMÉSZETTUDOMÁNYI KAR

A térképi generalizálás
automatizálásának bemutatása a
partvonalak és az országhatárok
példáján keresztül

SZAKDOLGOZAT
FÖLDTUDOMÁNYI ALAPSZAK

Készítette:

Kocsis Viktória

térképész és geoinformatikus szakirányú hallgató

Témavezető:

Ungvári Zsuzsanna

tanársegéd

ELTE Térképtudományi és Geoinformatikai Tanszék



Budapest, 2015

Tartalomjegyzék

1.	Bevezetés.....	4
2.	A generalizálás	5
	A generalizálás folyamata.....	6
	A generalizálás folyamatai, szabályai.....	8
3.	A generalizálás automatizálása	10
	A generalizálás automatizálásának definíciója, céljai	10
	A kézzel történő generalizálás előnyei, hátrányai	11
4.	A generalizálás modelljei.....	13
	Robinson és kollégái modellje (1978)	13
	Kilpeläinen modell (1997).....	13
	Brassel és Weibel modellje (1988).....	14
	McMaster és Shea modellje (1992).....	14
	Pszedo-fizikai generalizálási modell (2012).....	15
5.	Generalizálás raszter- és vektor alapú fájlformátumokban.....	18
	A vektoros- és raszteres adatmodell	18
	Raszter alapú generalizálás	19
	Vektor alapú generalizálás.....	20
	Az automatizált generalizálás műveletei	21
6.	Generalizálás térinformatikai szoftverekben	25
	QGIS	25
	SimpliPy	25
	Generalizer.....	34
	ArcGIS	43

Simplify Line	43
Simplify Polygon	44
Smooth Polygon	44
7. Kitekintés: egyéb térképi elemek generalizálása	47
8. Összegzés	48
9. Mellékletek	49
10. Köszönetnyilvánítás	55
11. Irodalomjegyzék	56
12. Ábrajegyzék	59

1. Bevezetés

A térképek már a kezdetekben a navigáció és az ismeretszerzés fontos eszközei voltak. Létrehozásuk során figyelni kellett a megjeleníthető részletek mennyiségére, hiszen a térképi felület korlátozza a lehetőségüket, ezért a részletek kiszűrésére, a térkép generalizálására van szükség. A kézzel történő generalizálás nagyon időigényes és szubjektív feladat, melynek létjogosultsága napjainkban egyre csökken. Rohanó világunkban egyre több új és naprakész térképre van szükség, aminek lendületét a kézzel végrehajtott folyamat nehezen tudja követni.

A számítástechnika megjelenése és rohamos fejlődése a térképkészítés folyamatát is befolyásolja, már szinte az egész eljárás digitálisan működik. Az adatfelmérés, adatszerzés és térkép megalkotása is digitálisan folyik: GPS felmérés, internetes adatbázisok és a különböző térinformatikai szoftverek segítségével.

Felmerül tehát a kérdés, hogy van-e lehetőség a generalizálás folyamatának teljes automatizálására, a munka minőségének megtartásával, de jelentős gyorsításával. Dolgozatomban röviden ismertetem a generalizálás alapvető műveleteit. Összehasonlítom az automatizált és kézzel végzett folyamatot, előnyeiket és hátrányaikat kiemelve, majd rátérek részletesen az automatizált folyamatra. A műveletek rövid leírása után a képzés során használt két térinformatikai szoftver, a QGIS és az ArcGIS segítségével teszteltem a jelenleg rendelkezésre álló lehetőségeket.

Dolgozatom célja, hogy bemutassam az automatizált generalizálás végrehajtásához jelenleg rendelkezésre álló eszközöket, algoritmusokat, azok eredményeit, pontosságát, esztétikusságát és hogy ezek felhasználásával lehetséges-e megfelelő minőségű térkép előállítása, emberi beavatkozás nélkül.

2. A generalizálás

A generalizálás a térkép információtartalmának csökkentése, amikor a térkép méretarányának, céljának vagy célközönségének változása ezt megköveteli. Például a méretarány csökkentése során, a térképen található objektumok némelyikét át kell helyeznünk, vagy elhagynunk őket, ugyanis a térképi megjelenítéskor rendelkezésünkre álló hely, vagyis a térkép befogadóképessége jelentősen kisebb lett. Ennek érdekében a térképen található objektumok számát, az adatok bonyolultságát, összetettségét redukálni kell, méghozzá olyan mértékben, hogy a megjelenített adat mennyisége megfeleljen a térkép méretaránya által kínált lehetőségeknek (Slocum, 2005).

A térképész munkája során a valóság egyes elemeiről, a térképraajz jelrendszerével közöl információkat a felhasználóval. Közöttük mindig vannak lényeges információk, fontos jelenségek, – az úgynevezett hírek – és akadnak mellékes elemek, ezek az úgynevezett zörejek. A térképszerkesztő egyik fontos feladata ezen információk közül kiválogatni azokat, amelyek a térkép méretaránya, rendeltetése és célja szempontjából a legmegfelelőbbek, hiszen a térképen nem ábrázolható teljességben környezetünk. A generalizálás végrehajtásához szükséges ismerni mindazt az információmennyiséget, amit egyszerűsíteni, általánosítani szeretnénk. Ehhez sok ténytet és jelenséget kell összegyűjteni, hogy lehetőségünk legyen tanulmányozásukra, valamint felismerhessük belső összefüggéseiket, a legfontosabbakat pedig kiválaszthassuk, és megkülönböztethessük a lényeges és tipikus jelenségeket (Mélykúti, 1970).

A térképi generalizálás mértékét öt tényező határozza meg: a térkép rendeltetése, méretaránya, a térképezni kívánt terület földrajzi adottságai, az ábrázolás módszere és a rendelkezésre álló alapadatok minősége. A legfontosabb ezek közül a térkép rendeltetése, ugyanis ez határozza meg a térkép tartalmát és méretarányát is. A méretarány csökkenésével a térképen kevesebb objektumot ábrázolunk, vagyis a generalizálás mértékét növeltük. A különböző földrajzi adottságokkal rendelkező területeken az azonos típusú tereptárgyak nem rendelkeznek egyforma jelentőséggel, ezért nem szabad a generalizálást általánosítani és a többi, más adottságú területen alkalmazni. Ehhez vegyük példának Norvégia és Afrika partvonalát. A norvég partok nagyon szabdalatosak, kisebb-nagyobb szigetekkel, öblökkel tarkítottak, míg Afrika partjai nem ilyen változatosak. A norvég partvonal generalizálásánál sokszor szükség lehet egyes szigetek, öblök

kiemelésére, akár méreten felüli ábrázolására, míg a kevésbé jelentőseket elhagyjuk. A térképi ábrázolás módszere nem képi, akár a fényképeken, hanem jelképi. Az egyes objektumok megjelenítésére jeleket, szimbólumokat alkalmazunk, amik általában méreten felül ábrázolják az objektumokat, néhány jellemző tulajdonságát kiemelve így szükség van a generalizálásra.



1. ábra: Afrika (Ádeni-öböl partja) és Norvégia partvonala azonos nagyításban, műholdképen

A generalizálás folyamata

Az alaptérképekből a levezetett térképek kartográfiai feldolgozás során jönnek létre. Az alaptérkép tartalmát a levezetett térkép ábrázolási módjának, céljának, és az alaptérképen található információk geometriai helyzetének, számának és sajátosságainak megfelelően a készülő térképhez igazítjuk. A térkép feladata, hogy a természetes és mesterséges objektumok, és jelenségek közül csak azokat ábrázolja, amik a térkép céljának és célközönségének szempontjából a legfontosabbak, legjellemzőbbek. Ennek érdekében a lényeges információkat kiválasztjuk, megjelenítjük, esetleg felnagyítjuk, míg

a térkép számára lényegtelen információkat elhagyjuk, vagy csak a tartalom alátámasztására szükséges mennyiségben ábrázoljuk. A térkép tartalmának generalizálása már a felmérés és az adatok felvétele során elkezdődik. A tervezett térkép típusától függően kiválogatjuk a számunkra szükséges objektumokat, hiszen más típusú adatokra van szükségünk egy tematikus, vagy topográfiai térkép elkészítéséhez. A nagy méretarányú alaptérképeken az ábrázolási lehetőségek még sokoldalúak, a rendelkezésre álló hely megfelelő geometriai lehetőségeket nyújt. A térképszerkesztéshez használt forrásadatok, térképek és mérések sokszor már önmagukban is generalizáltak. Ha ezek generalizálása helyesen zajlott, vagyis a generalizálási szabályokat megfelelően alkalmazták a nyers adatokon, megkönnyíti a további generalizálás tervezését. Ellenkező esetben szükség lehet az eredeti források alapján az adatok kijavítására, kiegészítésére (Klinghammer, 2010).

A térkép kicsinyítése során, az objektumok is egyre kisebbé válnak. Ekkor előfordulhat, hogy már nem érik el a minimálisan ábrázolható nagyságot, ezért az olvashatóság érdekében szükség van a generalizálásra. Ilyenkor vagy az ábrázolás helyességét, vagy a teljességét korlátoznunk kell. Ezt megtehetjük például nagyítással – nem mérethelyesen, egyszerűsítéssel – tehát nem alakhelyesen, vagy az objektum elhagyásával. A generalizálás elemi folyamatainak feladata, hogy csökkentse a bonyolultságot, de emellett a térbeli helyességet, az adatok pontosságát, az esztétikai értékelhetőséget, és a térkép logikus felépítését fenntartsa.

Ha a térképet olvashatóvá akarjuk tenni, annak vannak bizonyos határai. A térképen ábrázolt tények, ténycsoportok jellegét minden esetben meg kell tartani. Ezek a határok csak abban az esetben léphetők át, ha egy bizonyos térképi egységre eső, meghatározott mennyiségű térképi elemet egy új minőségi fogalommal helyettesítünk. Ezt az átmenetet hívjuk a generalizálás küszöbének.

A generalizálási küszöb átlépésének két formája van. Az első a mennyiségi generalizáláskor lép fel. Erre példa a települések, ahol a méretarány csökkenésével a házakat először felváltják a háztömbök, majd a település területét ábrázoló felületi jel. Így a pontszerű jelekből felületi jelek lesznek: habár a tájékozódási pontok száma megváltozik, de tájékoztató jellegüket megtartják. A mennyiségi generalizálás lényege, hogy az objektumok elterjedése, és annak törvényszerűségei a kisebb méretarányokban is kifejezésre jussanak.

A generalizálási küszöb átlépésének másik formája a minőségi generalizálás. Ennek során a térkép egy részletén ábrázolt tárgyak és jelenségek közül a legfontosabbakat megkeressük, és ezek ábrázolására törekedjük, úgy hogy a minőségi tények halmazát mennyiségi értékekkel helyettesítsük. Ez a települések példáját véve akkor történik meg, amikor a település területét ábrázoló felületi jelről áttérünk jelek használatára, mivel a méretarány csökkenésével a felületi jel olvashatatlanná válhat. Ezek az értékarányos jelek, amik a település lélekszámán alapulnak. További generalizálási küszöbök jelentkezhetnek, ha a mennyiségi ábrázolásról szükséges áttérni osztályokba sorolt, csoportos mennyiségi ábrázolásra, vagy relatív területi ábrázolásra.

A generalizálás során megváltozik a térkép kartometriai értékelhetősége, ha az ábrázolás módszere is változik. A mérhető jelölésekről áttérünk az egyezményes jelekre, az egyes objektumokhoz tartozó fogalmakról fogalmi osztályokra, rendszertani csoportosításra. A minőségi információk helyett mennyiségi információkat jelenítünk meg, közvetlen adatok helyett pedig származtatottakat (Klinghammer, 2010).

A generalizálás folyamatai, szabályai

A generalizálás folyamata közben különböző mértékben elemi folyamatok lépnek fel, ezek az egyszerűsítés, simítás, nagyobbítás, eltolás, összevonás, kiválasztás, klasszifikálás, értékelés, minősítés.

Egyszerűsítés: az egyik leggyakrabban használt generalizálási folyamat, amely során az objektum (vonal, vagy felület) megrajzolásához szükséges töréspontok számát csökkentjük, úgy hogy a tárgy alakját lehető legpontosabban meghagyjuk.

Nagyobbítás: a méretarány csökkentése során, a mérethelyesen ábrázolt objektumok túl kicsivé válhatnak, és nem érik el a minimálisan ábrázolható méretet, ezért méreten felül ábrázoljuk őket. Erre legjobb példa az autópályák, amelyeket kétvonalas jellel ábrázolunk.

Eltolás: ezen folyamat használatára többnyire a nagyobbítás következtében van szükség. A nagyított vagy egy szomszédos objektumot el kell tolni annak érdekében, hogy a kettő ne tömörüljön, ne érjenek össze, vagy ne fedjék egymást.

Összevonás: több hasonló, pontszerű objektum összevonása, egy nagy, több pontból álló csoportba, mint például a házak összevonása háztömbökbe.

Kiválasztás: a generalizálandó területen található különböző tereptárgyak (tereptárgy típusok) közül ki kell választani, melyek azok, amiket ábrázolni kívánunk és melyek azok, amiket elhagyunk, attól függően, hogy melyik fontosabb a térképi megjelenítés céljából.

Klasszifikálás: összetett felületi és vonalszerű objektumok alaprajz szerinti ábrázolása a méretarány csökkenése miatt nem mindig kivitelezhető. Ilyenkor az objektumot (esetleg objektumcsoportot) egy térképi jellel ábrázoljuk, az alaprajzszerű ábrázolás helyett.

Értékelés, minősítés: ezt a folyamatot akkor használjuk, ha a méretarány csökkenése után az objektum egy jellemző tulajdonsága már nem kivehető. Ilyenkor ezt a tulajdonságot eltúlozva jelenítjük meg a térképen (Klinghammer, 2010).

A generalizálás folyamatai nem függetlenek egymástól, egymásra is hatással vannak. Erre legegyszerűbb példa a nagyítás, amely sok esetben eltolást von maga után. A generalizálás során az objektumcsoportokat a térkép típusától függően más, és más sorrendben kell egyszerűsíteni, és a szabályait az új térkép méretarányától és céljától függően más súllyal és sorrendben kell alkalmazni. Ezen alapján elkülöníthetők a generalizálás típusai. A mérethez kötött generalizálás során, amikor a kiindulási és cél méretarány közti különbség nem nagy, és a térkép rendeltetése hasonló, leginkább az egyszerűsítést, nagyobbítást és eltolást alkalmazzuk. A szabad generalizálás során a kisebb méretarányokban az ábrázolás helyességét erősen korlátozni kell az olvashatóság érdekében. Ilyenkor hasonló objektumok csoportja helyett csak egy objektumot ábrázolnak, egyetlen jellemzővel helyettesítik. Ehhez főként az összevonásra, kiválasztásra, klasszifikálásra és az értékelésre, minősítésre van szükség. A témához kötött generalizálás során egy meghatározott objektumot, objektumcsoportot vagy témát kívánunk hangsúlyozni, a többit elhagyjuk vagy elnyomjuk. Megvalósítására a kiválasztás, klasszifikálás és az értékelés, minősítés alkalmazására van szükség (Klinghammer, 2010).

3. A generalizálás automatizálása

A számítástechnika rohamos fejlődésének eredménye, hogy mára minden tudományban nélkülözhetlenné vált. Az emberek igénye a különböző folyamatok automatizálására egyre nőtt, így egyszerűsítve és jelentősen meggyorsítva munkájukat. Ahogy más területeken, a kartográfában is megjelent ez az igény: napjainkban már GPS-szel történő felmérés adataiból dolgozunk, internetes adatbázisokból egészítjük ki ismeretanyagunkat, különböző programokat használunk a térképek elkészítéséhez.

A térképkészítés egyik legfontosabb feladatát, a generalizálás folyamatát is automatizálhatjuk. Mivel a generalizálás a szubjektív feladat, nehezen számszerűsíthető, ezért azon információk generalizálása automatizálható, amelyek matematikai formában is leírhatók. Az első próbálkozások már az 1960-as években megindultak. Az első algoritmusok a pontok számának csökkentését tűzték ki célul, mind a pontszerű objektumoknál, mind a vonalszerű és felületi elemeknél. Ezek egyike Töpfer pontszerű objektumokra megalkotott algoritmus, amely a térkép méretaránya lapján határozza meg az ábrázolható tereptárgyak számát. Több algoritmus is megjelent, Douglas–Peucker (1973) vonalegyszerűsítő, Pillewiser felületi elemeket egyszerűsítő algoritmus. Az algoritmusok mellett több megközelítést és modellt is megalkottak a digitális generalizálás folyamatához, ezek mindegyike segít megérteni a generalizálás komplex feladatát: Brassel és Weibel (1988) vagy McMaster és Shea modelljei (Cecconi, 2003).

A generalizálás automatizálásának definíciója, céljai

A digitális generalizálás egy adatforrásból, térbeli vagy attribútumbeli transzformációkon keresztül létrehozott térképi adat. Az automatizálás olyan eszközök, folyamatok használatát jelenti, amelyekkel a megadott programok önműködően vezérelhetők, így az emberi közreműködés a munkában szükségtelen, vagy minimális mértékű. Ehhez olyan eszközök, programok szükségesek, amelyek képesek az ember manuális és/vagy szellemi munkáját átvenni, és megegyező pontossággal, de gyorsabban el tudják végezni.

A hagyományos technológiával szemben a digitális térképeken könnyebbé vált a térképi tartalom aktualizálása, valamint egyszerűbben lehet egyes munkafolyamatokat is részben, vagy teljes egészében automatizálni. A generalizálás szubjektív folyamat,

azonban automatizálással objektívvá tehető. A kartográfiai generalizálás elemi folyamatait matematikai formulákkal írjuk le, így könnyebben elkerülhetjük az esetleges minőségbeli különbségeket, amik a kézzel történő helyesbítés során felléphetnek. A szerkesztés közben elengedhetetlen, hogy olyan legyen a végeredmény, hogy az olvasók is megértsék, lehető legkifejezőbb térképet lehessen létrehozni.

A fenti célok teljesítéséhez azonban az adatok előkészítése szükséges. Az automatizáláshoz az adatok digitális formában kell, hogy a rendelkezésünkre álljanak. Az automatizálás kihathat a jelkulcsra, az ábrázolási módszerekre és a kifejezési lehetőségekre. Ennek a hatásait, és következményeit szükséges vizsgálni, olyan szakemberek közreműködésével, akik tudása megfelelő az automatizálás végrehajtásához.

A kézzel történő generalizálás előnyei, hátrányai

A térképkészítés egész folyamata időigényes és jelentősen szubjektív feladat, de ezek közül talán a generalizálás folyamata a legnehezebben objektivizálható. A térképész választja ki az elemeket a nagyobb méretarányú térképről, amiket kisebb méretarányban ábrázolna, különböző változtatásokat eszközöl, hogy a részleteket kiszűrje, de megtartsa az információsűrűséget, a térkép méretarányát és célját figyelembe véve. Kézzel történik a rajzolás folyamata, az egyszerűsítés. A térképi objektumok típusától függően a kevésbé fontos elemeket elhagyják, lekicsinyítik, és a vonalakat elsimítják. A felületeket összevonhatják, szükség esetén szimbólumokkal helyettesítik, hogy a kicsinyítés során létrejövő zsúfoltságot csökkentsék. Mivel a generalizálás egész folyamata manuálisan történik, a végeredmény minősége jelentősen függ a térképész tapasztaltságától, megítélő képességétől és esztétikai érzékétől, ezért előfordulhatnak különbségek a végeredmények színvonalában, apró hibák mindig lehetnek a térképen. A kartográfiai munka kézzel végrehajtva sokkal lassabb: az adatok gyűjtése, kiválasztása, a generalizálás folyamata mind a készítő feladata. Így a tömegtermelés igényeit nem minden esetben tudja kielégíteni.

A számítástechnika fejlődése a térképészet minden területére kihatott. Nem csak a generalizálás, de a kartográfiai folyamat minden lépésének automatizálása is elkezdődött, megjelentek a különböző GIS programok, grafikai programok és a GPS felmérés adatait felhasználó alkalmazások. Ahogy ezek fejlődnek, úgy nőtt az igény a generalizálás kapacitásának növelésére, amit a kézzel történő generalizálás már nem tud

teljesíteni. Az automatizált generalizálás erre megoldást nyújt, a különböző algoritmusokkal jelentősen gyorsítható a folyamat, miközben a pontossága nem csökken, de akár növelhető is. A digitális folyamat megkönnyíti a térképész munkáját, hiszen napjainkban már többnyire digitális alapadatokból, adatbázisokból, és térképekből dolgozunk.

A kézzel történő kartográfiai munka kiváltása jelentős költségcsökkentéssel jár, ezért a térképek elérhetősége folyamatosan növekszik. Az automatizálással az adatok feldolgozásához és új térképek kiadásához szükséges idő csökken, így naprakészebb térképek jelenhetnek meg, egyre gyakrabban. Mivel egyre több a digitális adatbázis, ezekből digitális adatnyeréssel könnyebb dolgozni, az információkhoz való hozzájutás, azok továbbítása és szükség esetén kiegészítésük is sokkal gyorsabbá, könnyebbé válik a digitális folyamatok segítségével. Az automatizálás folyamatának fontos tulajdonsága, hogy a vetületek és méretarányok változtatása egyszerűbbé válik. Ezen kívül kiszélesedett a vizuális lehetőségek téra, így többféle adathordozó közül is választhatunk, például papír alapú, vagy virtuális megjelenítés. Napjainkban egyre nagyobb előnyre tesznek szert a digitális formátumban megjelenő térképek, amelyek több méretarányban is megtekinthetőek, ilyenek például az interneten is megjelenő térképek. Ezen térképi adatbázisokat több méretarányban kell megalkotni, ez kézzel rendkívül időigényes lenne - az automatizálással az információtartalom gyorsabban és egyszerűbben változtatható. A legnépszerűbb webes térképszolgáltatások például az OpenStreetMap és a Google Maps.

Az generalizálás folyamatának automatizálásához szükséges a térképi elemek közötti kapcsolatok megértése. A műveleteket szabályokba kell foglalni, hogy így az irányítható legyen. Az automatizálás folyamata bonyolult, nehéz azokat az algoritmusokat megalkotni, amik minden esetben, például különböző tagoltságú partvonalak esetén is, megfelelő eredményt hoznak. A térképész minden észrevétele, amely javíthatja a térkép kifejezőképességét, nehezen lekódolható, meglehetősen bonyolult minden problémát kifejezni, ugyanis ezek a problémák függenek a térkép aktuális tartalmától.

4. A generalizálás modelljei

Ahhoz, hogy a generalizálás összetettségét könnyebben megértsük, a folyamattal foglalkozó tudósok megpróbálták több modellt, rendszert alkotni a leírására. Ezek közül némelyik az alapvető műveletekre, és a közöttük levő kapcsolatra épül, míg mások összetett modelleket alkottak.

Robinson és kollégái modellje (1978)

Robinson és kollégái 1978-ban az elsők között alkották meg modelljüket, amely értelmezi a generalizálás folyamatát. A folyamatot két fő részre osztották: a kiválasztásra (előkészítés) és a generalizálás folyamatára. A kiválasztás során elkülönítik azokat az objektumokat, amelyeket a térképen ábrázolni szeretnénk, azoktól, amelyeket elhagynak. Erre példa a tematikus térkép készítéséhez szükséges információk kiválogatása és az alaptérképhez szükséges objektumok, például utak, közigazgatási határok kiválasztása. A lépés során a névrajz elemeit is szükséges vizsgálni, az alapján, hogy mire van szükség a térképen való ábrázoláshoz: tematikus térképeken csak minimális mértékben tüntetjük fel a vízrajz neveket, míg a turista térképeknél ezek fontos, akár a terepi tájékozódást segítő információk.

A generalizálás három eljárást foglal magába: egyszerűsítés, klasszifikáció és szimbolizáció. Az egyszerűsítés során a kevésbé fontos adatokat, tulajdonságokat távolítjuk el a geometriai összetettség csökkentésére. A klasszifikáció folyamata a tereptárgyak kategorizálását hajtja végre. A szimbolizáció az alaprajz szerint már nem ábrázolható objektumokhoz különböző jeleket társít, így már megfelelően ábrázolhatók kisebb méretarányokban.

Kilpeläinen modell (1997)

Különböző méretarányú térképeket tartalmazó adatbázisok leírására készült modell. Kilpeläinen egy nagy, összetett, részletes adatbázist, a Digitális Alapadatbázist (Digital Landscape Modell, röviden DLM) képzelt el, amelyből különböző, kisebb méretarányú Digitális Kartográfiai Modellt (Digital Cartographic Modell, röviden DCM) lehet létrehozni. Az elsődleges DLM a legnagyobb méretarányú, legrészletesebb és pontosabb, ebből másodlagos DLM-eket lehet levezetni, kisebb méretarányokban való használatra. A DLM egy teljesen digitális, számítógépes adatbázis, amely vizuális megjelenítésre alkalmatlan. Ezzel ellentétben a DCM-ek térképi megjelenítései az

adatbázisnak, amit a DLM generalizálásával és szimbolizációjával készíthetünk. Kilpeläinen modelljében minden DCM-et az elsődleges adatbázisból, vagy egy nagyobb méretarányú térképből vezetünk le. Külön DLM-eket hozhatunk létre minden méretarányhoz vagy felbontáshoz, így a DCM közvetlenül ebből az adatbázisból generálható. Ez a modell egyik első, részletes leírása, amelyet már némely nyugat-európai országban, például Svájcban alkalmaznak is digitális topográfiai alaptérkép adatbázisok létrehozásakor.

Brassel és Weibel modellje (1988)

Brassel és Weibel generalizálási modelljét a domborzatábrázolás példáján dolgozta ki. Ennek két fő célja volt: először is, egy megfelelő egyszerűsítési stratégiát alakítsanak ki különböző domborzattípusokhoz, méretarányokhoz és térképcélokhoz; másodsor pedig, hogy ezt megfelelően tudják automatizálni, a lehető legteljesebben. A modell öt főbb lépésből áll: a struktúra (alakzat) felismerése, folyamat felismerés, folyamat modellezése, folyamat végrehajtása, és végül az adatok megjelenítése és kiértékelése. A struktúra felismerése során kiválasztjuk a térképi objektumokat, térbeli elhelyezkedésüket, kapcsolatukat és fontosságukat figyelembe véve. A folyamat felismerése a generalizáláshoz szükséges megfelelő operátorok kiválasztása és paramétereik megszabása, azok alapján, hogy az alapadatokat pontosan hogyan kell egyszerűsíteni, milyen konfliktusok lehetnek az objektumok között, valamint melyik tereptárgyat, információkat szeretnénk megjeleníteni a generalizált adatbázisban. A folyamat modellezése ezen operátorok pontos használatát írja le: mely szabályokat és műveleteket kell követni; megszerkeszti a használandó algoritmusokat, és egy digitális rendszerbe foglalja őket. A két utolsó lépés egyikében a megtervezett folyamat végrehajtása zajlik, amely során megtörténik a generalizálás. Ezután megjelenítik az adatokat, végül pedig kiértékelik az így elkészült adatbázist.

McMaster és Shea modellje (1992)

Annak érdekében, hogy egy átfogó fogalmi modellt hozzanak létre, három kérdéssel fogalmazták meg az elméletüket: miért, mikor szükséges a generalizálás és hogyan hajtsuk végre a folyamatot.

A generalizálás elméleti céljai – miért generalizálunk – a térkép összetettségének csökkentése úgy, hogy az információtartalmát és geometriai pontosságát a lehető legnagyobb mértékben megtartsuk. Fontos kérdés, hogy a térképész hogyan tudja az

információtartalmat a térkép méretéhez optimalizálni, hiszen egy nagyméretarányú szelvény információtartalma nem jeleníthető meg egy kisebb méretarányú térképen megfelelően, ehhez több elemet el kell távolítani és a részleteket módosítani kell. Másik fontos cél az attribútumok és a térszerkezet pontosságának fenntartása. Az attribútumok pontosságára a klasszifikálás folyamata közben szükséges figyelni, hogy a mennyiségi, és minőségi különbségek megfelelően érzékelhetőek legyenek a generalizálás után. A térszerkezet fenntartása az egyszerűsítés során fontos, amikor koordinátpárokat távolítunk el a térképről. A folyamat az elemek elcsúszásához vezethet, hiba keletkezik. Eltolás közben is megjelenhet ez a hiba, amikor két tereptárgyat eltávolítunk egymástól, hogy ne csússzanak egybe. A cél az, hogy minél jobban elkerüljük ezt az elcsúszást és a térkép geometriai szerkezetét a lehető legnagyobb mértékben megtartsuk.

A digitális környezetben fel kell ismernünk azokat a helyzeteket, amelyek során a generalizálás szükséges. A térkép túlszűfolttsága esetén, amikor kis helyre túl sok tereptárgy kerül, nehezen érzékelhető a geometriai kapcsolat az elemek között. Kicsinyítés során az elemek közel kerülhetnek, érinthetik egymást, ilyenkor szükséges lehet az objektumok eltolása. Ellentmondásos helyzetek is előfordulhatnak a generalizálás folyamata során. Ilyenkor eltávolíthatunk egy elemet, amely fontos egy másik objektum térbeli elhelyezkedésének pontos megjelenítéséhez. A partvonalak generalizálása során eltávolíthatunk például egy öblöt, amely partján egy város áll. Ilyenkor vagy a várost vagy a partvonalat át kell helyezni ahhoz, hogy a város területe a partvonalon maradjon. Megtörténhet az is, hogy a van egy változó komplexitású vonal egyszerűsítése során túl bonyolulttá válik annak egy szakasza és más mértékű generalizálás szükséges ezeken a szakaszokon (Slocum, 2005).

Pszeudo-fizikai generalizálási modell (2012)

A modellt Joubran Abu Daoud és Doytsher azért kezdték el létrehozni, mert bár több generalizálási modellt is kidolgoztak a 2000-es évekig, hiányzott egy teljes körű, működő modell a generalizálás automatizálására. A modell teljesen új megközelítést alkalmaz. A térképen található objektumokat – mind a pontszerű, vonalas és felületi elemeket – az elektromos mezőben található erőkhöz hasonló erőkkel ruházza fel, így vizsgálva kölcsönhatásaikat, és egyéni viselkedésüket, hogy a generalizálás folyamata érthetőbb lehessen.

A generalizálás közben az objektumokat tulajdonságaik, típusok és az általuk megjelenített adatok alapján kezeli a modell. Az adott mértékű generalizálás a térképi objektumok közötti versengést jelenti az üres térképi felületért, melyet az erejük korlátoz, a térképi szabályokkal összhangban.

Az elektromos mezőben található nagyszámú elem viselkedését az erejük irányítja. Befolyásolják a körülöttük található elemeket és azok is befolyásolják őket. Ez a modell ezt a hatásrendszert utánozza, így a generalizálás folyamata is kölcsönös erőkön alapul. Az erő egy több változóból álló függvény eredményeképp jön létre. Ezek a változók az elemek tulajdonságai: felülete, típusa, formája; valamint a térképfelület jellemzői: a környező terület sűrűsége, üres felület nagysága, a térkép célja és méretaránya. Ezek mellett a generalizálási folyamatot különböző kartográfiai szabályok irányítják.

Az objektumok erőhatásait egy függvény adja meg, melyben megfelelő arányban megtalálhatók a tulajdonságok, amik befolyásolják az elem viselkedését a generalizálás közben.

Terület: minél nagyobb az objektum kiterjedése az adott méretarányban, annál erősebb.

Fontosság: a térkép céljától függő érték, a térkép tematikájához szorosan kapcsolódó objektumok magasabb értékűek.

Magasság: a modellben a magasabb objektumok erősebbek, mivel kevésbé alkalmasak helyváltatásra.

Alak: a négyzet alakú elemek a legerősebbek.

Típus: egy rugalmas érték, ami a tereptárgy anyagát adja meg, a kartográfiai szabályok és a térkép tartalma alapján. Ilyenek például a fák, a generalizált térképen könnyebben elmozdíthatók, mint például az épületek.

Sűrűség: a tereptárgyakat a környező terület ugyanúgy befolyásolja, így különbség van egy sűrűn beépített városi, és egy falusi terület között. Mivel a városi terület sűrűbben beépített, nehezebb az elemeket elmozdítani.

Az erők közötti kölcsönhatás, a vonzó és taszító hatások befolyásolják és irányítják az elemek mozgását a környező elemekhez képest. Hogy a modellben megmutatható legyen a hatások befolyásoló képessége, az elemek köré egy effektív héjat kell definiálni. Ez a héj egy pufferezónaként működik, amit egy toleranciaérték alapján adnak meg. Térbeli konfliktus jöhet létre, ha egy elem behatol egy másik elem effektív héjába, ilyenkor a köztük levő erők működésbe lépnek és megkísérik a probléma megoldását. Az erők mindig csak a héjban levő objektumrészre hatnak, az effektív héjra merőlegesen. A két elem közti kapcsolat a gyengébbik elemre ható erőként írható, és mint az elektromos töltések esetén, az vonzás és taszítás jön létre, de ezek megjelenése ellentétes. Az azonos elemek vonzzák egymást, az ellentétesek pedig taszítják.

A taszítás különböző típusú objektumok között jön létre, hogy egymástól távol maradjanak. Az erő két elem között a típusuk különbségével egyenesen arányos, így a hasonlóak nem hatnak egymásra semmilyen mértékben sem. Az erő a távolsággal fordítottan arányos, így a közeli jobban hatnak egymásra.

Az ugyanolyan típusú elemek között vonzóerő jön létre, mely összecsoportosítja őket, ha túl közel vannak egymáshoz, így könnyen megoldhatók vele a térbeli konfliktusok. A vonzás mértéke egyenesen arányos a kölcsönhatásban levő két elem erőinek összegével és fordítottan arányos távolságukkal és helyzetükkel.

A generalizálás folyamata során az erőhatások irányítják az elemek viselkedését, így ha sok elem található benne, nagyobb az esély a változtatásra. Mindig a leggyengébb elem módosítása oldja meg a térbeli konfliktust, hiszen a ráható összes erő eredője szerint mozog a térképfelületen. A folyamat a legkisebb jelentőségű elemek törlésével kezdődik. Ezután következik a csoportosítás, mely során a vonzóerő által felismert, egymáshoz közeli, azonos típusú objektumokat csoportokba rendezzük. Ismét a kisebb jelentőségű elem mozog a nagyobb felé, a közeli éleket egybeforrasztva és a kisebb elemet elforgatva, hogy párhuzamosak legyenek. Ez az üres térképfelület nagyításának második fázisa, először a gyengébb, majd az egyre erősebb elemeket vizsgálja. Az utolsó lépés a tereptárgyak mozgatása, átformálása és a szimbólum méretarány változtatása. Ennek során a taszítóerők a gyengébb elemeket eltávolítják az erősebbtől, arra ügyelve, hogy egy effektív héj területére se essen. Ha ez nem lehetséges, a gyengébb elem torzítására, vagy méretarányának megváltoztatására van szükség (Daoud és Doytsher, 2012).

5. Generalizálás raszter- és vektor alapú fájlformátumokban

A vektoros- és raszteres adatmodell

A térképészetben két adatmodellel találkozhatunk: a vektoros és a raszteres adatmodell. A két modell a felépítésében, logikájában, és a kartográfiában betöltött szerepében is jelentősen különbözik.

A vektoros adatmodell építőkövei a koordinátákkal rendelkező pontok. Ezekből a pontokból vonalat, vonalláncokat, majd ezekből poligonokat építhetünk fel. A tereptárgyakat jellegzetes pontjaik leírásával jellemezzük, ezeket egy szabályrendszer alapján összekötjük (vagy pontszerű tereptárgyak esetén nem) és egy definíció alapján megadjuk, hogy mely pontok alkotnak egy objektumot. A két adatpont között nem található további pontok, hiszen ezek jellegzetes pontok, így virtuálisan a valóságot jól megközelítően ábrázolják a tereptárgyakat. Az adatrendszer előnye, hogy kevesebb adattal és kisebb adatbázissal írják le az ábrázolt felületet, relációs adatok kapcsolhatók hozzájuk. Az elemi objektumokból komplex objektumokat hozhatunk létre, ezek mindegyikéhez egy azonosítót társítunk, hogy ennek segítségével attribútumokat kapcsolhassunk hozzájuk (Elek, 2006).

A raszteres adatrendszert felépítő legkisebb geometriai elem a pixel. Ezek mindegyike önálló objektum és külön paraméterrel rendelkeznek. Ezek a pixelek több forrásból származhatnak, lehetnek egy úrfelvétel raszteres adatai, de lehet egy légifotó szkennelt képe is. De nem a pixelek mérete jelentős, hanem hogy a pixelek tulajdonságai rajzolnak ki egy nagyobb objektumot. A vektoros adatrendszerrel ellentétben ezeknek az objektumoknak nincsen definíciós szabálya, mert nem lesznek elkülöníthető objektumok. Az ábrázolt felületen minden pixel állapotát szükséges ismernünk: még azt is, amely esetleg üres. Az adatmodell hátránya, hogy nehezen definiálhatók benne az objektumok, az adatbázis kapcsolatok kiépítése nehézkes, de a szomszédsági viszonyok könnyen vizsgálhatók benne (Elek, 2006).

A vektoros adatmodellt térképi adatbázisok építésére használjuk. A felméréstől kezdve minden objektumot koordinátáikkal azonosítunk, így egy felmérés adatait célszerű ebben a formátumban tárolni. A koordináták – gondoljuk akár földrajzi, vagy a munkalap koordinátáira – alkalmasak matematikai számításokban való felhasználásra,

így a térképi generalizálásban is könnyen alkalmazható ez a rendszer. Ellenben a raszteres rendszerben ilyen számítások nehézkesen végezhetőek. Esetenként mégis szükség lehet az műholdfelvételek, légifotók generalizálására, ez többnyire a felbontás csökkentésével történik.

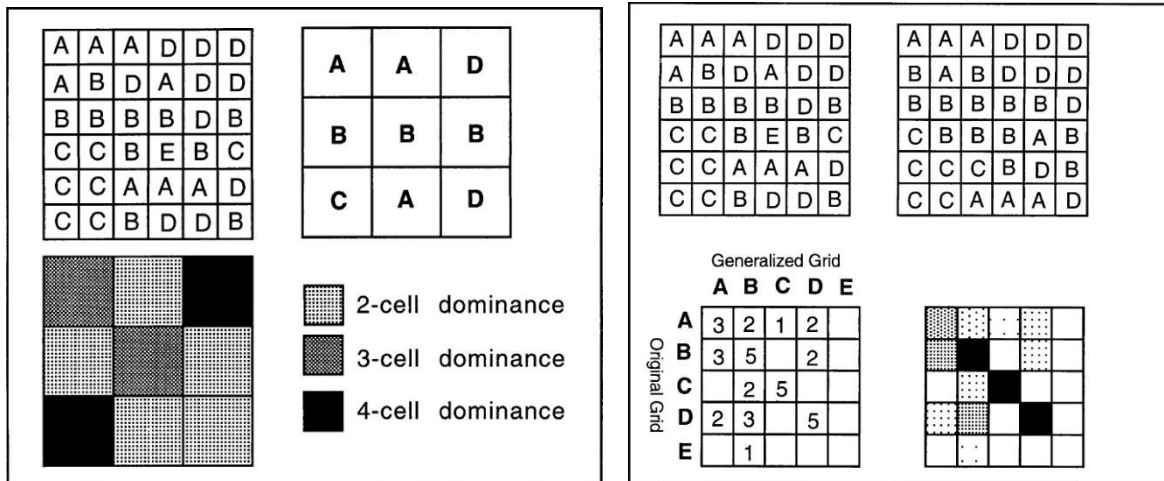
Raszter alapú generalizálás

A raszter formátumú adatok generalizálásának négy fajtája van: a szerkezeti generalizálás, numerikus generalizálás, numerikus kategorizálás és az attribútumok kategóriáinak generalizálása. Az első típus újradefiniálja a mátrix szerkezetét, miközben a cellák (pixelek) és a felületek közti kapcsolatot megváltoztatja. A numerikus generalizálás elsimítja az objektumok változásait, a raszter mátrix változásait jelentéktelenebbé teszi, emellett kiemeli vagy élesebbé teszi a felületek széleit. A numerikus kategorizálás a digitális képek klasszifikációja. A képfeldolgozás során, hagyományos eszközökkel a bemeneti raszterből egy kategorizált, kimeneti képet kapunk. A kategóriák generalizálása műveleténél csak a képen található csoportok generalizálása történik. A folyamat eredményeképp csökken a geometriai bonyolultság-úgy, hogy a kategóriák számát egy bizonyos mennyiségűre csökkentjük.

Ahhoz, hogy a fenti folyamatok végeredményeként létrejött változások még szembetűnőbbek legyenek, gyakran különböző grafikai manipulációkkal hangsúlyozzák őket. Néha az eredeti grid szerkezetét a generalizált képre helyezik, hogy így láthatóvá tegyék a szerkezet és a felbontás változását. A kategóriák egyszerűsítésénél gyakran a színek változtatásához folyamodnak, hogy a változásokat jobban kiemeljék.

Különböző vizuális technikák is alkalmazhatók, hogy láthatóvá tehesük a generalizálás hatásait. Az első módszer a pixelek nagyobbitása: e transzformáció közben több, például 2×2 pixelből hozunk létre egy nagyobbat, vagyis a raszteres kép térbeli felbontása megváltozik. Ez a cella nem minden esetben csak egy kategóriába eső pixelekből áll, ilyenkor a nagyobb számban, és nagyobb valószínűséggel előforduló attribútumot alkalmazzuk az új felbontásban. Ilyenkor a tulajdonság arányait a hozzá rendelt szín különböző telítettségű árnyalatival ábrázolhatjuk, ebből leolvasható egy attribútum dominanciája. A sötétebb cellákban az egyik attribútum adatnak a gyakorisága, míg a világosabbakban kevesebb fordul elő belőle. Ezzel a módszerrel könnyedén meg lehet különböztetni a homogén tulajdonságokkal rendelkező területeket azoktól, amelyek változatosabb tulajdonságúak.

Egy másik módszer a raszter alapú generalizálásra egy új mátrix létrehozása. Ebben egy terület jellegét vizsgáló szűrővel változtatják meg az attribútumokat, az adott cella körüli tulajdonságok előfordulása szerint. Ez a folyamat megfelelően alkalmazható számszerű adatokkal, mennyiségekkel illusztrált tulajdonságok, például változások megjelenítésére.



2. ábra: Raszter alapú állományok generalizálása dominancia megállapításával (bal oldali ábra) és szűrő (jobb oldali ábra) használatával

Számos technikát hoztak létre a numerikus raszter generalizálásra, ilyen a kép kiértékelés és a domborzatmodellek létrehozása. Ezeken az adatokon végigfuttatott generalizálás, a különböző szűrők használata számszerűsíthető különbségeket okoz az eredeti és a feldolgozott kép között. Hogy ezt láthatóvá tegyük, lehetőségünk van háromdimenziós felület létrehozására, de mivel ez elég bonyolult struktúrájú lehet, azonnal egy simító algoritmust is végig kell a felületen futtatni. Színek alkalmazásával is láthatóvá tehetők az új, és régi pixelek közötti számszerű változások, például az alkalmazott szín telítettségének változtatásával. Ezzel a módszerrel egyszerre megjeleníthető a magasság és a lejtőszög (McMaster és Veregin, 1997).

Vektor alapú generalizálás

A vektoros adatmodellben megadott pont mindegyike koordinátákkal rendelkezik. Így ezekre különböző matematikai műveletek és algoritmusok írhatók fel, így megkönnyítve a generalizálás automatizálását. A generalizálásnak különböző műveletei vannak, ezek pedig az adatsor, illetve a generalizálni kívánt objektumok geometriájától függően használhatók. Vonalas elemekre használjuk például az egyszerűsítést és a simítást, a klasszifikációt pontszerű és felületi elemekre is

alkalmazhatjuk, míg eltolásra a három különböző geometriájú elem közül bármelyiknél szükség lehet.

Az automatizált generalizálás műveletei

Egyszerűsítés

A generalizálás közben leggyakrabban használt módszer az egyszerűsítés. Az egyszerűsítés során csökken az adott vonal csomópontjainak száma. Célja, hogy az elem geometriáját a lehető legjobban megőrizze, de emellett a maximális számú koordinátát távolítsa el. Összetett geometriai elemzéseket, kritériumokat használ az operátor, hogy kiválassza a jellemző vagy az úgynevezett kritikus pontokat.

A vonalak egyszerűsítése az algoritmusok működése alapján öt csoportra bontható: független pont módszer, lokális (szomszédospont-vizsgálati) módszerű eljárások, korlátozottan kiterjesztett lokális módszerű algoritmusok, korlátlanul kiterjesztett helyi elemző algoritmus, globális eljárások.

A független pontokat vizsgáló algoritmus a vonalon elfoglalt helyük alapján választja ki a koordinátákat, ilyen például az n-edik pont módszer, amely minden n-edik pontot megőriz. Előnye, hogy számításilag hatékony és gyors, viszont nem veszi figyelembe a pontok és elemek geomorfológiai jelentőségét.

A lokális (szomszédospont-vizsgálati) módszerű eljárások az egyes pontok vizsgálatánál a közvetlenül szomszédos pontokat veszi figyelembe a vizsgált pont fontosságának megítélésére. A szomszédos koordinátapárral való kapcsolatot vizsgálja, többnyire a köztük levő távolság és az általuk bezárt szög alapján.

A korlátozottan kiterjesztett lokális módszerű algoritmusok nem csak a közvetlenül szomszédos pontokat vizsgálják, hanem mindkét irányban 3-4 pontból álló, nagyobb vonalszakaszokat elemeznek. Az előző típushoz hasonlóan, ezek az algoritmusok is a pontok közötti távolság, és a köztük levő vonalszakaszok által bezárt szöget által vizsgálják a vonalas elemeket.

A korlátlanul kiterjesztett helyi elemző algoritmusok a vonalas elemek nagyobb részeit vizsgálják. Különlegességük, hogy az algoritmus korlátozó feltételei (tolerancia értékei) zárják le a vizsgált szakaszt, hanem a geomorfológiai összetettséget.

A globális eljárások a leggyakrabban használt algoritmusok. Nemcsak a vonallancok kisebb szakaszait vizsgálják, hanem figyelembe az egész vonalas elemet. Az algoritmusok előnye, hogy általában természetesebb vonalakat jobb közelítéssel generalizál, míg mesterséges objektumoknál, például egy szögletes utcahálózatban nem megfelelőek az eredményei, sok esetben eltorzítja a derékszögeket. A vonalak közelítésére szolgáló trendvonalak az eredeti vonal szögletességének számításán alapulnak. Ahol irányt vált a vonal, inflexiós pontja van: ezen pontok összekapcsolása adja meg a trendvonalat. Célja, hogy egy vonalas elem vagy adatbázis összetettségét elemezzük vele. Alkalmazhatjuk számításokhoz: a trendvonal hosszát a teljes hosszal elosztva megkapjuk a vonal jellemzésére használt szinuszoitást. A közelítőleg egyenes szakaszokon értéke nagyobb, itt az eredeti vonal hasonló a trendvonalhoz; összetettebb vonalszakaszokon jelentősen eltérhet a két vonal egymástól. Minél nagyobb a különbség a két vonal között – és kisebb a szinuszoitás- annál komplexebb a vonalas elem (NCGIA Core Curriculum, 1994).

Simítás

A simítást sokszor az egyszerűsítéssel azonosítják, de valójában a két módszer elkülönül egymástól. Nem törli a kiválasztott pontokat, hanem eltolja őket, így javítva az elem kinézetén: áthelyezi a pontokat, hogy elsimítsa az esetleges erős éleket, kiugrásokat a vonal futásában. Más simító algoritmusok pedig új pontokat szűrnak be, ezek általában görbévé alakítják a törött vonallancot. Többnyire az egyszerűsítéssel együtt alkalmazzák, mellyel esztétikus, és geometriailag is elfogadható minőségű eredmény jön létre.

Halmazképzés

A halmazképzés folyamata során több pontszerű objektumot összeillesztünk. Csoportosítjuk őket, és felületi egységként jelenítjük meg a generalizált térképen. A folyamat nehézsége a generalizálás küszöbének meghatározásában rejlik, meg kell állapítani azt a pontsűrűséget, amely felett a már megváltozik az ábrázolási módszer.

Egyesítés

A méretarány csökkentésével az egymáshoz közel álló poligonok túl közel kerülhetnek egymáshoz, esetenként összeecsúszhatnak, ilyen a felületi elemeket egybeolvaszthatjuk. Alkalmazható kontinuumokra és diszkrét felületi elemekre is. A diszkrét felületi elemekre példa egy sokszínű területen a növényzeti borítottság változása. A túl kicsi felületeket összeolvasztjuk a nagyobb, jellegzetesebb felszínborítottság-

típussal. A kontinuumok lehetnek népszámlálás, vagy más felmérésből származó adatsorok, melyeket felületi jelekkel kívánunk megjeleníteni. Ilyenkor a hasonló értékeket összevonjuk. Elég nehezen programozható, bonyolult folyamat, és mint a legtöbb operátor, városi környezetben automatizáltan kevésbé hatékonyan alkalmazható, több emberi beavatkozást igényel.

Klasszifikáció

A klasszifikáció összetett, pont és felületi elemekből álló területet egyetlen pontszerű elemként jeleníti meg. A méretarány nagymértékű megváltozásakor van szükség használatára, hogy a tereptárgyakat, vagy azok csoportjait új szimbólumokkal jelenítsük meg. Összetett operátor, hiszen a térkép geometriájának megváltoztatását is magában foglalja. Például a komplexebb geometriájú házakból álló tömbök téglalappal való helyettesítése, ilyenkor az egybeolvasztást is használjuk.

Összeolvasztás

Az összeolvasztás során az egymáshoz közel álló vonalas elemek csoportját egybeolvasztjuk: közel párhuzamos elemeket, például egy vasútállomás vágányait. Az összeolvasztás másik fajtája, amikor a felületi kiterjedéssel rendelkező folyókból vonalas elemet képezünk. Ez kivitelezhető úgy, hogy a felület széleit átlagoljuk.

Finomítás

A finomítás az újraszimbolizálás egyik formája, olyan művelet, amely csökkenti egy nagyszámú elemből álló csoport tagjainak számát. Ilyenek például a vízhálózat: egy komplex geometriájú objektumcsoportot egyszerűbb alakban tipizálunk a folyamat során úgy, hogy megtartsuk a vízfolyások főbb eloszlását.

Nagyobbítás

Ez a folyamat egyike a gyakrabban használt operátoroknak. Sokszor szükség van egy részlet felnagyítására, hogy kicsinyítés után is megmaradjon az általa ábrázolt tulajdonság. A partvonalak generalizálása során ez a szigeteknél fordulhat elő: ha egy sziget túl kicsi az ábrázoláshoz, de egy számunkra – a térkép céljának – fontos objektum, például repülőtér, település található rajta, a szigetet nem elhagyjuk, hanem nagyítva, méretarányon felül ábrázoljuk a térképen.

Áthelyezés, eltolás

Az objektumok áthelyezése az egyik legnehezebb generalizálási operátor, mivel összetett számításokra van szükség végrehajtásához. Példaként vehetünk egy partvonalat, ahol közel egymáshoz, egy vasútvonal és egy autópálya fut. Ilyenkor az útvonalakat eltoljuk a parttól, valamint a vasútvonalat eltávolítjuk autópályától, hogy ne fedjék egymást. (McMaster és Veregin, 1997; Slocum 2005)

6. Generalizálás térinformatikai szoftverekben

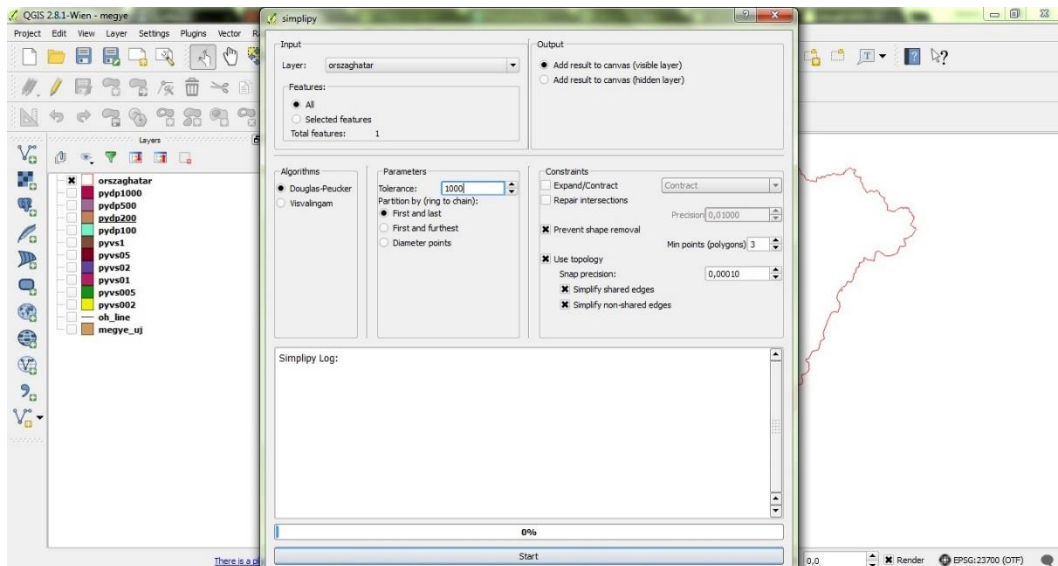
Napjainkban egyre elterjedtebb a különböző térinformatikai szoftverek használata a térképkészítés folyamatához. Ezek jelentősége, hogy az előkészületek, az adatgyűjtés és azok kiválogatása után a térképkészítés egész folyamata végrehajtható vele, a digitalizálástól egészen a térképlap összeállításáig. Két program generalizáló módszereit próbáltam ki: a QGIS két plugin-ját, és az ArcGIS Cartography Toolbox moduljának három operátorát.

QGIS

A QGIS (régábbi nevén QuantumGIS) egy nyílt forráskódú, ingyenes térinformatikai program. Alkalmas adatbázis építésére a készülő térképekhez. A szoftver telepíthető pluginjai között kettő található, ami különböző generalizálásra alkalmas algoritmusokat használ. A poligonokra alkalmazható SimpliPy, és a vonalas elemek generalizálására szolgáló Generalizer. Az utóbbi még csak kísérleti, de több algoritmus használatára van lehetőség, mint a SimpliPy-ban.

SimpliPy

A SimpliPy plugin poligonok egyszerűsítésére alkalmas algoritmusokat tartalmaz. Két algoritmus használata közül lehet választani, ezek egyike a Douglas–Peucker, a másik pedig a Visvalingam-algoritmus. Beállíthatjuk az algoritmus határértékét, mely a használt adatsor vetületétől függően lehet fokban vagy méterben. A négyzetes hengervetületű, a szoftverben WGS 84-nek nevezett (EPSG:4326) adatsoromhoz fokot, az EOVS (EPSG: 23700) vetületűhöz pedig méterben kellett megadni a toleranciaértéket. A kezelőablakban kiválaszthatjuk, hogy megőrizze-e a topológiát, vagyis a létrejövő poligonok ne fedjék egymást, és ne keletkezzen közöttük hézag. Ehhez szintén beállítható egy toleranciaérték, aminél közelebbi pontokat egymásra helyezi. Az algoritmusok használata után az egyszerűsített vonal nem metszi saját magát, és megakadályozható a poligonok törlése. Lehetőség van csak az objektum egy részletének egyszerűsítésére is. Az elkészült réteget megjeleníti a projektben, de nem lehet automatikusan fájlba menteni (QGIS Python Plugin Repository).



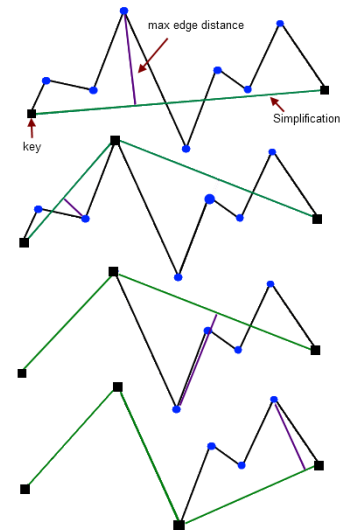
3. ábra: A SimpliPy plugin kezelőfelülete

Douglas–Peucker-algoritmus

Az algoritmus célja a törött vonalból álló vonalas vagy felületi objektumok egyszerűsítése a töréspontok csökkentésével úgy, hogy az egyszerűsítéshez kulcspontokat jelöl ki, és ezek merőleges távolságát vizsgálja egy viszonyítási vonaltól. A pontok további vizsgálatát egy küszöbérték alapján végzi, amit az algoritmus futása előtt kell megadnunk.

Az algoritmus:

1. a vonal első és utolsó csomópontjának összekötése, az eredeti polyline-on
2. mindegyik töréspont merőleges távolságát leméri az így keletkezett vonaltól
3. az így legtávolabb levő pont (*max. edge distance*), ha távolsága nagyobb, mint a küszöbérték, kulcsponttá (*key*) válik, és bekerül az egyszerűsített vonalba (*Simplification*)
4. a kulcspont és végpont összekötése



4. ábra: A Douglas–Peucker-algoritmus vizualizációja

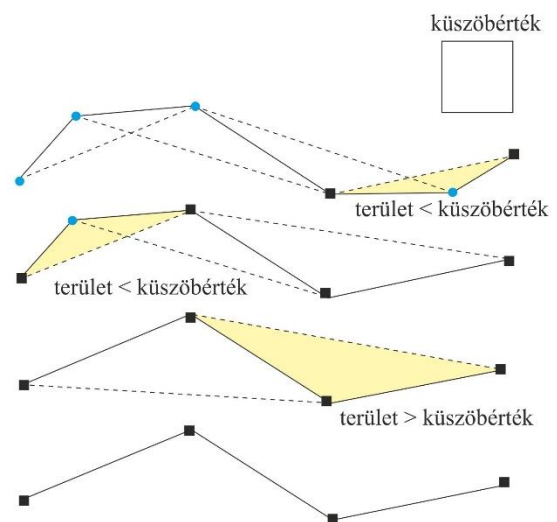
A lépések a második ponttól addig ismétlődnek, amíg a toleranciaértéknél nagyobb távolsággal rendelkező pontok el nem fogynak, tehát nem keletkezik újabb kulcspont (Douglas és Peucker, 1973).

Visvalingam-algoritmus

Az algoritmus alapötlete az, hogy kezdetben elhagyjuk azokat a pontokat, amelyek kitörlése a legkevesebb területvesztéssel jár. Az elemeket fokozatosan, a kisebb hatásútól a nagyobbig távolítja el. Az algoritmus kidolgozói azért a területet választották jellemzőnek, mert így használható a felület csökkentésére, ha egy észlelhető limit betartására van szükség. A terület a legmegbízhatóbb érték a vizsgálatra, mivel a távolságtól és a bezárt szögektől is függ a mérete. A terület vizsgálata máshol is megjelenik, Robinson és kollégái munkájában (1984) és a poligonok eltávolításához is többnyire ez az érték használatos. Minden köztes pontnak van egy effektív területe, ez az a területi elmozdulás, ami akkor jönne létre, ha az adott pontot elhagynánk a sokszögvonalból. Fontos, hogy ezt az értéket nem az eredeti vonalhoz képest számolja az algoritmus, hanem az folyamat aktuálisan egyszerűsített vonala alapján.

Az algoritmus:

1. kiszámolja az effektív területet
2. kitörli az összes pontot, ami nulla területtel rendelkezik, és eltárolja őket egy listában
3. a legkisebb effektív területtel rendelkező pont lesz az aktuális pont
4. ha a kiszámolt terület kisebb, mint a legutóbb eltávolított ponté, ezt a területet használja minimumként, elkerülve, hogy az aktuális pontot eltávolítsuk
5. kitörli az aktuális pontot az eredeti listából, és egy másikba sorolja a területével együtt, így kiszűrve a pontot
6. a két szomszédos pont effektív területét kiszámolja

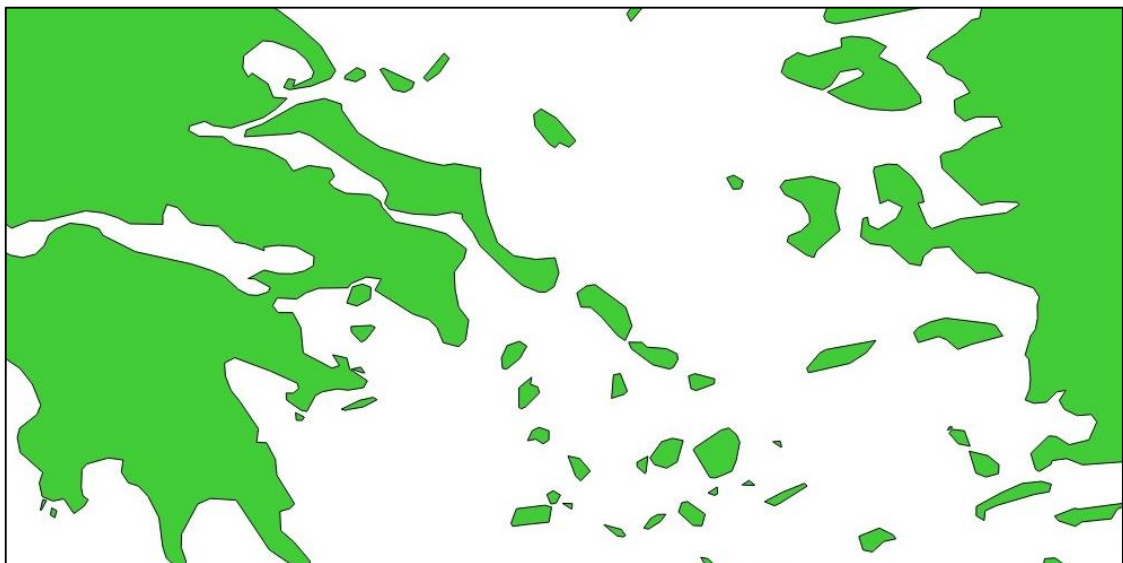


5. ábra: Visvalingam algoritmus vizualizációja

Azt addig ismétli a harmadik ponttól, amíg a kezdeti vonalból csak két pont marad: a kezdő és végpontja. (Visvalingam és Whyatt, 1993)

Partvonalak generalizálása

A partvonal generalizálásának bemutatására egy Európát ábrázoló adatsort választottam, így lehetőségem volt ugyanannak az algoritmusnak vizsgálatára az országhatárok példáján is, ugyanazon küszöbértékkel. A kiindulási méretarány 1:2 500 000-1:3 000 000 körüli lehet. A forrásnál nem volt megadva méretarány, de nyomtatott térképekkel összehasonlítva közel megállapítható. Először a Visvalingam algoritmust futattam le, a plugin által engedélyezett legkisebb küszöbértékkel, 0, 00001°-kal, és beállítottam, hogy a topológiát megőrizze. Ez a kis küszöbérték nem hozott jelentősebb változást az adatsoromban, csak a kisebb szigetek sarkait, kanyarulatait – a görög szigeteknél – vágta le, így nem jelentős az egyszerűsítés. A második küszöbérték az előző kétszerese volt, 0,00002°. Ez nem hozott jelentősebb különbséget az előzőhöz képest, itt is csupán a görög szigetek poligonjainak egyszerűsítése történt meg. Ezért a következő küszöbértéknek az eredeti tízszeresét, 0,0001°-ot választottam. Az előző értékekhez képest jelentősebb változást hozott. A norvég partvonalat és a görög szigeteket figyeltem meg részletesebben, hiszen többnyire itt lehetséges, hogy a generalizálás már eltorzította a vonalakat. Nagyobb mértékű egyszerűsítést történt meg, a pontok száma jelentősen csökkent, de nem torzította még jelentősen az eredeti vonalat. Nem záródtak be fjordok, öblök, de a szigetek már nem olyan szépek. Túlságosan szögletesek lettek, de egy simító algoritmus végigfuttatása után ez az algoritmus megfelelő lenne, úgy már természetesebb hatást keltenének a szigetek. Az algoritmus eredményeit ábrázoló kép az 1. számú mellékletben látható.



6. ábra: Görög szigetvilág egyszerűsített rajzolata Visvalingam-algoritmussal, 0,0001°-os küszöbértékkel

A következő küszöbérték $0,001^\circ$ volt. A tagoltabb partvonalaknál már torzulások jelentkeztek, a fjordok, öblök futását több esetben is eltorzította az algoritmus. Ez legtöbbször a lekerekített végű fjordoknál észlelhető, ott csak egy pont maradt. Ezek a hibák legfőképpen a norvég partvonalon, és a görög szigetvilág környékén figyelhetők meg. Találtam több helyet, ahol a poligonok fedik egymást. Ez az algoritmus hibája, mivel a beállítottam, hogy a topológiát figyelje. Ennél nagyobb küszöbérték használata nem lenne megfelelő erre az adatsorra, mivel már így is több hiba felmerült, amiket manuálisan kell javítani. Az eredményen egy simító adatsor lefuttatása után az eredmény esztétikus lehet, és a néhány fedést, és hibás öblöt ki lehet javítani.

A Douglas–Peucker-algoritmus futtatásához használt legkisebb küszöbérték $0,0001^\circ$ volt. Beállítottam a topológia figyelembevételét, és a kis poligonok törlésének megakadályozását. Ez az érték túl kicsinek bizonyult, semmilyen változást nem hozott az adatsoromban, így ennek ötszörösével próbálkoztam. Ez az érték sem volt elég, hogy változtasson a poligonokon, ezért a küszöbértéket az eredeti tízszeresére, $0,001$ -re emeltem. Ez a küszöbérték a vonalfutásban jelentősebb változást nem hozott, de láthatóan kevesebb lett a pontok száma, így az egyszerűsítés megfelelő. A kisebb szigetek, öblök, és a fjordok formája sem torzult, a kisebb sarkok eltávolítása, lekerekítése történt meg. Ezt a küszöbértéket nagy méretarány különbség esetén nem lehetne használni, mivel mértéke ehhez sajnos elég kicsi. A következő küszöbérték $0,005^\circ$ volt. Az előző küszöbértékhez hasonlóan ez is látható változást hozott. Az egyszerűsítés még nagyobb mértékben sikerült, de ugyanúgy követi az eredeti vonalak futását, csak kevesebb ponttal ábrázolja azok közelítését. A kisebb szigeteket, félszigeteket már kezdi kicsit eltorzítani, jellemzőbb íveiket már levágja, de nagyobb méretarány változtatás esetén erre a mértékre szükség lehet. Az eredményt több nagyításban is megnéztem. A legnagyobb, amiben elfogadható a minősége, az 1: 4 000 000. Nem feltűnő a generalizálás mértéke, ebben a nagyításban is szép eredményt hoz. A nagyítást folyamatosan csökkentettem, és a legkisebb mérték, amiben jelentősebb manuális javítás nélkül használható lenne, az 1: 8 000 000. Itt már esetenként szükség lehet néhány kisebb sziget eltávolítására, jellemzőbb öblök, szorosok nagyobbítására, de az egyszerűsítés önmagában szép eredményt hozott. Az előbbi határértékekkel tesztelt algoritmusok eredményei a 2. számú mellékletben találhatóak. A következő toleranciaérték az előző kétszerese, $0,01^\circ$ volt. A vonalak főbb futását ez is hasonlóan megőrizte, de az egyszerűsítés mértéke az eredményben már sokkal jelentősebb. A kisebb szigetek többsége háromszög alakult,

ez a kisebb földnyelveknél is észlelhető. Van, ahol a poligon önmagát metszi, vagy egy másik poligont fed. Mindkettő olyan hiba, amit a topológia ellenőrzésnek ki kellett volna küszöbölnie. Ez a küszöbérték alkalmazható kevésbé tagolt partvonalak, például a francia és spanyol partok egyszerűsítésére, azok jellemző vonalfutását rendkívül jól megtartja. Egész Európa partvonalaira ezzel a küszöbértékkel nem kapnánk szép eredményt, rendkívül sok benne a hiba, amely manuális javításra szorul.



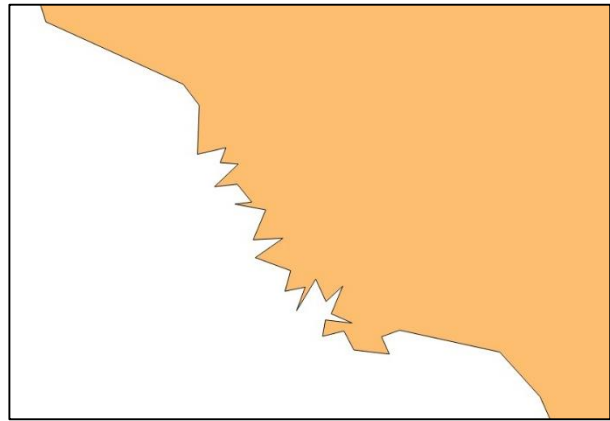
7. ábra: A görög szigetvilág egyszerűsítése Douglas-Peucker-algoritmussal, $0,01^\circ$ -os határértékkel

A két algoritmus közötti legnagyobb különbség abban rejlik, hogy más-más mértékben egyszerűsítik a felhasznált adatsort. Míg a Visvalingam-algoritmus már kisebb értéknél jelentős változást hozott, a Douglas–Peucker-algoritmus jelentős változást csak nagyságrendekkel nagyobb érték esetén mutatott. Ennek oka természetesen az eltérő számítási mód, az első esetben a küszöbérték terület alapú, míg a második esetben távolságokat számít ki. Ennek hatása érezhető a vonalak futásának megtartásában is. Míg a Douglas–Peucker-algoritmus megtartotta a vonalak jellemző futását nagyobb küszöbértékeknél is, a másik algoritmus sok esetben eltorzította azokat. Erre egy egyszerű példa a norvég fjordok, amik az előbbi esetben sokáig megtartották formájukat, nem záródtak be, míg az utóbbi algoritmus eredményeképpen ezek többször is megtörténtek.

Országhatár és megyehatárok generalizálása

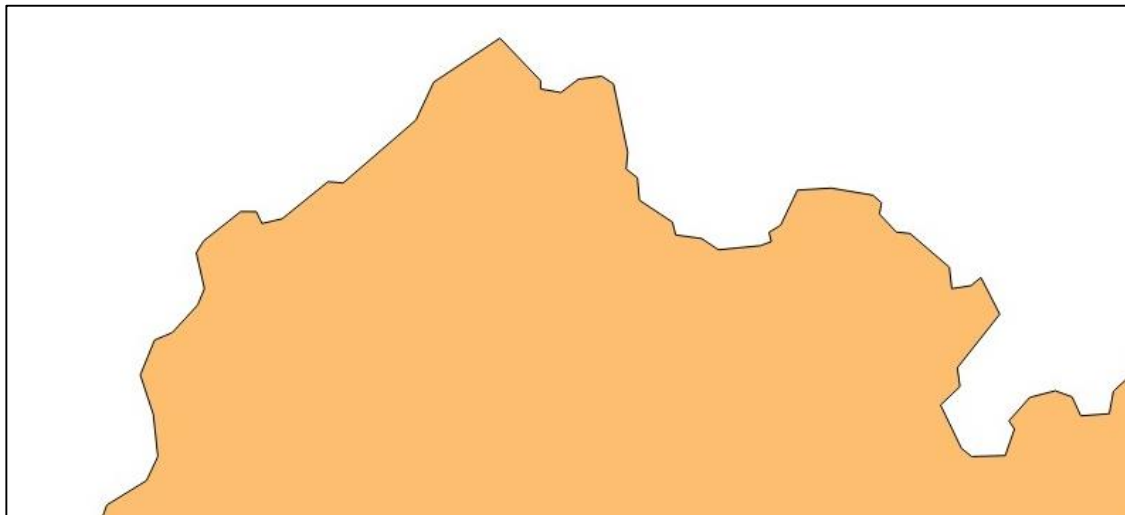
A Magyarország határát ábrázoló rétegemen először a Douglas–Peucker-algoritmust futtattam le, több általam választott küszöbértékkel. A vektoros állomány méretaránya nem volt megadva, de más térképekkel összehasonlítva, nagyjából 1:300 000- 1:400 000 körüli. A legkisebb érték, ami látható eredményt hozott, 100 méter volt. Az adatsor csak Magyarország határát ábrázolja, egy poligon egyszerűsítésére volt szükség, így a topológiai hibák figyelését nem állítottam be. Az Európa adatsorhoz hasonlóan ezen sem hozott olyan jelentős eredményt a legkisebb érték. Csökkentette a pontok számát, többnyire csak szélesebb kanyarulatok levágása történt meg. Fontos volt számomra, hogy a Duna folyását, és a délnyugati, a Drávához közel futó, jellemzően kanyargó határrészt hogyan egyszerűsíti az algoritmus. A Duna folyását jelentősen nem változtatta meg, a déli határrészen pedig semmi változást nem hozott. Úgy gondolom, hogy ez a kis küszöbérték nem hoz akkora változást, amiért érdemes lenne lefuttatni, nagy méretarány változtatás esetén nem lenne megfelelő. A második küszöbérték az első kétszerese, 200 méter volt. A határvonal fő futását megőrizve, de azt az előbbinél nagyobb mértékben egyszerűsítette, láthatóan több pontot távolított el. A kisebb, nem olyan jelentős íveket levágta, azt egy nagyobb, az ívek alapvonalát követő vonallal helyettesítette. Már előfordult, hogy a jellemző, élesebb kanyarulatokat levágja, és helyettük éles sarkokat hoz létre, de megfelelő nagyításban ezek nem észlelhetők. Mennyiségük sem olyan nagy, kézzel történő javítással megoldható a probléma. Ezután egy nagyobb küszöbértékkel, az eredeti ötszörösével, 500 méterrel futtattam az algoritmust. Az algoritmus előnye itt már jól kivehető, a vonal főbb futását megtartja, a részletek folyamatos, egyre nagyobb mértékű eltávolításával. Egyre kevesebb kisebb kanyarulat található az egyszerűsített poligon körvonalában, és a kisebb részletek is folyamatosan eltűnnek, nagyon szögletes lesz az egész összhatás. A kanyargósabb déli határszakaszon hegyes sarkok jelennek meg, kezd eltűnni a jellege, de még mindig felismerhető. A Duna folyását már az eredetitől eltorzította, és túl szögletes lett, de ez manuálisan javítható, és javítása szükséges is. Ezután, és egy simító algoritmus futtatása után a rajzolat valószínűleg megfelelő lenne, egy kisebb méretarányú térképalapként való használatához. Lefuttattam az algoritmust még egy küszöbértékkel, ami az első tízszere, 1000 méter volt. Ez a küszöbérték már nagyon jelentős változásokat hozott az adatsorban. Az előző egyszerűsített vonallal összehasonlítva, rendkívül egyszerű vonalrajzot eredményezett, de a főbb vonalfutást még itt is megtartotta. Az északi határszakasz, amit a Duna alkot, teljesen elvesztette természetes, ívesen kanyargó

jellegét, a déli határszakasz Dráva menti része pedig éles sarkok sorozatává vált. Egy simító algoritmus talán elsimítaná ezeket a nagyon éles részeket, de úgy vélem, hogy ez a küszöbérték már nem lenne megfelelő, és nincs is szükség ilyen mértékű generalizálására az országhatárnak. Az algoritmus teszteredményei a 3. számú mellékletben láthatóak.



8. ábra: A déli, Dráva menti határszakasz, Douglas-Peucker-algoritmussal, 1000 méteres küszöbértékkel

A Visvalingam-algoritmust több kisebb értékkel is kipróbáltam, de nem hozott változást adatsoromban. A legkisebb küszöbérték a $0,002^\circ$ volt. A Douglas–Peucker-algoritmushoz hasonlóan kezdetben ez is csak a szélesebb kanyarok, csücskök levágását okozta, bár az előbbitől eltérő módon. A déli, szaggatott határ egyszerűsítése ennél az értéknél még nem történt meg, de a Duna folyásáé igen. Nem jelentős a változás, a folyam természetes jellegét nagyon jól megtartotta, még nem találhatók éles kanyarulatok, sarkok benne.



9. ábra: Magyarország nógrádi határszakaszának egyszerűsítése $0,002^\circ$ -os határértékű Visvalingam-algoritmussal

Ezután ennek két és félszeresével próbálkoztam. Azt vettem észre, hogy ekkora küszöbérték változtatás nem hoz eltérő eredményt, az előző értékhez viszonyítva. Így a következő vizsgálati érték $0,01^\circ$ volt. Nagymértékű egyszerűsítést ez az érték sem okozott az előző értékhez képest, de a szélesebb hajlatokat, kanyarokat kiegyenesítette. Ami eltérő a Douglas–Peucker-algoritmushoz képest, hogy a kisebb elemek egyszerűsítése

még nem kezdődött el. Itt a kisebb ívek, sarkok, érintetlenek maradnak, inkább a szélesebb, laposabb kanyarulatok egyszerűsítése történik meg először. Ezután tovább növeltem a küszöbértéket, először $0,02^\circ$, $0,05^\circ$, $0,1^\circ$, majd így tovább, ezen érték nagyságrendekkel való növelésével folytattam, egészen tízezerig. Ezek a küszöbértékek a $0,01$ -es küszöbértékkel megegyező eredményt hoztak, többszöri próbálkozásra is. Ez valószínűleg egy programhiba, mivel az Európa adatsoromon jelentősebb változást is sikerült végrehajtanom.

A két algoritmus közül határvonalak egyszerűsítésére mindenképpen a Douglas–Peucker-algoritmust alkalmaznám. A kisebb részleteket fokozatosan, legjelentékenyebbtől az egyre fontosabb felé haladva távolítja el, így rendkívül jól felhasználható egy térkép több különböző célméretarányba való generalizálásához. Ez sem hoz tökéletes eredményt, az apróbb, ám fontos részleteket eltorzítja, a természetes eredetű vonalak futását nagyobb küszöbértékkel már egészen természetellenes hatásúra egyszerűsíti. Ezen részeknél szükség van manuális javításra, és a nagyon szögletes, éles részeken esetenként egy simító algoritmus végigfuttatása is szükséges lehet.

A plugin más közigazgatási határok, például megyehatárok egyszerűsítésére is megfelelő. Mivel az algoritmus futtatása előtt be lehet állítani a topológiai hibák kiküszöbölését, a megyehatárok találkozását, sarokpontjait megfelelően ábrázolja, azok nem csúsznak el egymástól. A küszöbértékek, amik az országhatárra megfelelőek, használhatók a megyék generalizálására is.

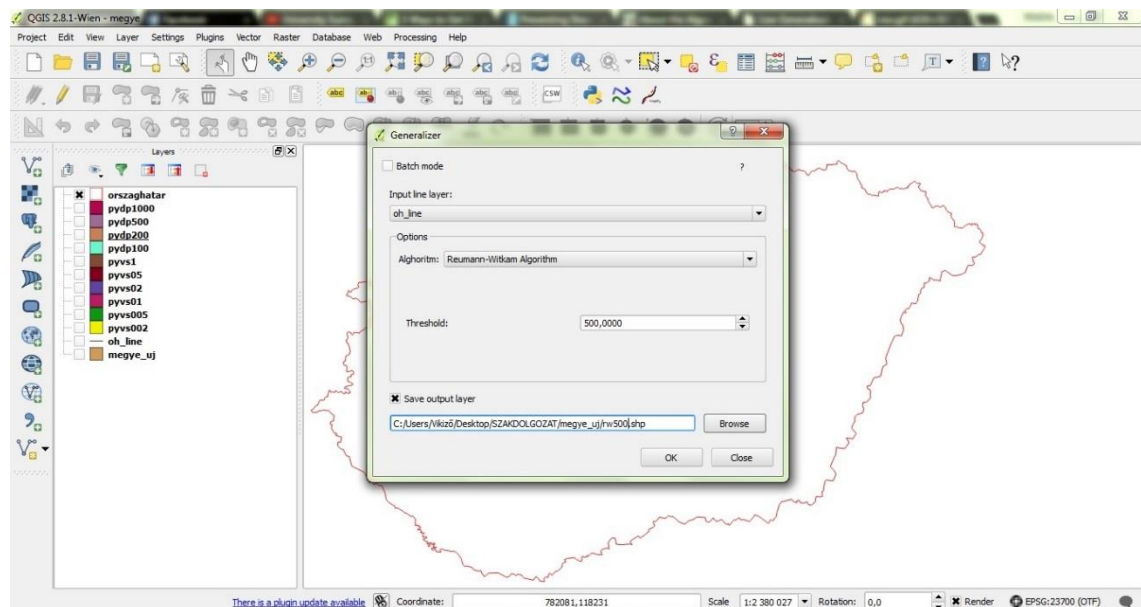
A plugin általam észlelt hibái

A plugin használata közben több hibával is találkoztam. Többször történt az, hogy az algoritmusok lefuttatása után egy üres réteg jött létre, egy elemet sem tartalmazott a fájl. A másik rendszeres hiba a „láthatatlan” réteg volt. Az algoritmus látszólag végigfutott az adatsoromon, és új réteg is létrejött eredményeképpen, de a munkafelület nem mutatott semmit. Az attribútum táblában (*jobb klikk, majd Open Attribute Table*) látható volt az összes objektum, ugyanúgy, mint a többi, sikeresen végrehajtott algoritmus után keletkezett réteg attribútum táblájában. Ezen kívül előfordult, hogy az algoritmus nem futott végig, egy hibát találva kilépett, és csak az adatsor egy részét egyszerűsítette. Ami rendkívül zavaró még, hogy nem lehet a rétegeket fájlba menteni. Mentésüket egyenként kell végrehajtani, de sajnos ez sem mindig sikerült. A nagyobb küszöbértékű egyszerűsítéseknél, amikor esetleg egy poligon eltávolításra került, már nem lehetett

elmenteni, mivel különbözött az attribútum táblában található, és a térképen megjelenített elemek száma. Szerencsére ez a hiba kiküszöbölhető: ilyenkor meg kell keresni a térbeli kiterjedés nélkül maradt attribútumot, kitörölni, így a réteg mentése lehetségessé válik. A plugin automatikusan a legutóbbi réteget állítja be az alapréteggként (*Input: Layer*), erre figyelni kell. Sajnos, néha a mentett fájlokban sem jelentek meg a korábban már sikeresen elkészült rétegek, így munkámat többször újra kellett kezdenem.

Generalizer

A plugin egy, a GRASSGIS v.generalize modulján alapuló generalizáló plugin, vonalas elemekre. A felhasználható algoritmusok között vannak egyszerűsítő, simító algoritmusok, és a kicsi objektumok eltávolítására használható algoritmus. Az egyszerűsítő algoritmusok a Douglas–Peucker, Jenk, Lang, Reumann–Witkam és egy csomópontcsökkentő algoritmus. A simító algoritmusok között megtalálható Boyle előretétkintő algoritmus, Chaiken-algoritmus, Hermite spline (görbe) interpoláció, McMaster távolsággal súlyozott és csúszó-átlagoló algoritmus és a kígyó (Snakes) algoritmus. A küszöbértéket a vetülettől függően fokban illetve méterben lehet megadni az egyszerűsítő algoritmusoknál, míg a simító algoritmusoknál több különböző érték beállítása szükséges algoritmustól függően. Az algoritmusok alkalmazása során a topológia megőrzésére nincs lehetőség. A generalizálás eredményeképpen létrejött réteget hozzá lehet adni a projekthez és fájlba is lehetséges menteni (QGIS Python Plugin Repository).



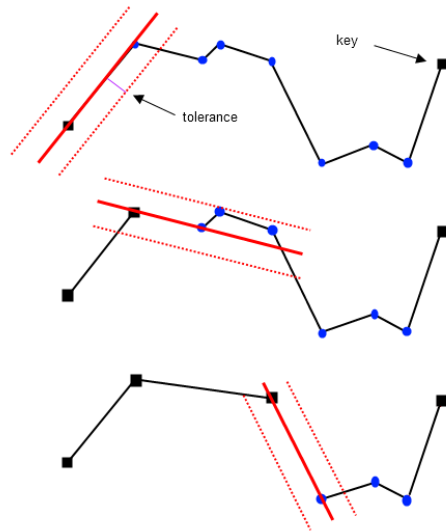
10. ábra: A Generalizer plugin kezelőfelülete

Reumann–Witkam algoritmus

A Douglas–Peucker-algoritmushoz hasonlóan a merőleges távolságon alapul. A különbség, hogy a viszonyító vonalat a kulcspontra, és az azt követő ponton keresztül húzza, nem pedig két kulcspontra (kulcspontra lehet a végpont és kezdőpont is) között.

Az algoritmus

1. az első pont kiválasztása kulcspontként (*key*)
2. ha kevesebb, mint három pont van utána, kilép
3. vonal húzása a kulcspontra és az azt követő ponton keresztül
4. a kulcspontra utáni második pont lesz a tesztpontra
 - a. ha a merőleges távolság a tesztpontratól a viszonyító vonalig kisebb, mint a toleranciaérték (*tolerance*):
 - i. hozzáadjuk a pontot
 - ii. megismételjük az új tesztpontra
 - b. ha a merőleges távolság nagyobb, mint a toleranciaérték:
 - i. töröljük az összes pontot a kulcspontra és az aktuális tesztpontra között, így a legnagyobb merőleges távolsággal rendelkező pont marad
 - ii. ha nincs köztük egy pont sem, nem törölünk egy pontot sem
5. hozzáadjuk a kulcspontra az egyszerűsített vonalhoz, és a második lépéstől megismételjük (Curve simplification turk project).



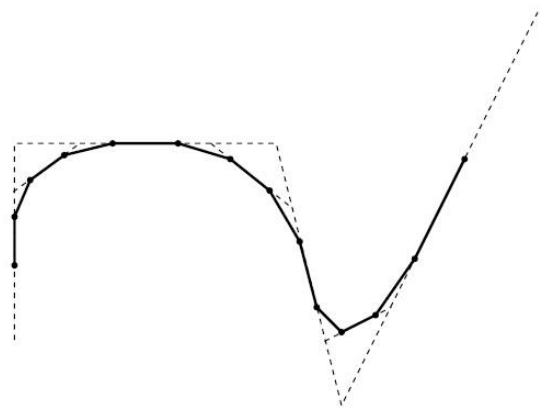
11. ábra: A Reumann–Witkam algoritmus vizualizációja

Csomópontcsökkentő algoritmus (vertex reduction)

Ez az algoritmus a legegyszerűbb a pluginben található algoritmusok közül. Lényege, hogy minden pontot töröl, ami közelebb áll szomszédjához, mint a megadott küszöbérték. Egyszerűsége miatt gyorsan végrehajtható, de nem veszi figyelembe a vonal futását, ezért eredménye sok esetben nem megfelelő.

Chaiken-algoritmus

Az algoritmus alapjául az objektumban található sarkok levágására szolgál. A levágás fix arányokkal történik: két pont közé egy új pontpárt illeszt be az algoritmus úgy, hogy az első pont a két pont közötti szakasz egynegyedénél, a másik pont pedig a háromnegyedénél legyen. Ezután a beillesztett pontok fogják alkotni a vonalat.



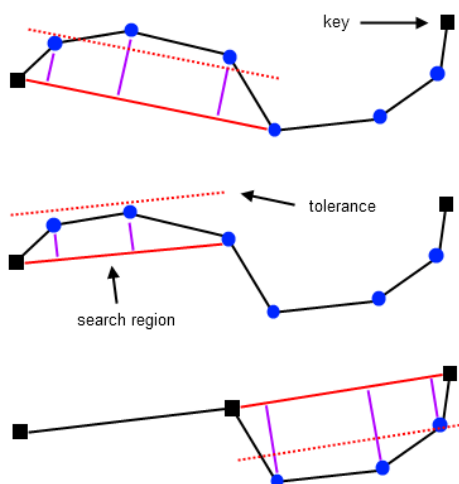
12. ábra: A Chaiken-algoritmus folyamata

További pontokat illeszt be az algoritmust, így közelítve az ívet. Polyline-okra és zárt alakzatokra egyaránt alkalmazható, például egy négyzeten végigfuttatva az algoritmust, egy körívet közelítünk (Joy, 1999).

Snakes algoritmus

A snakes algoritmus egy vonalsimító és közelítő algoritmus. Eredményét két paraméter irányítja: [alfa] a vonal rugalmasságát, [béta] pedig a merevségét szabályozza. Igen számításigényes módszer, ezért futási ideje hosszú.

Lang-algoritmus



13. ábra: A Lang algoritmus vizualizációja

A Douglas–Peucker-algoritmushoz hasonlóan egy vonalhoz számított merőleges távolság alapján vizsgálja a pontokat. Ebben az esetben a kulcspont (az ábrán *key*) és a végpont között húzunk egy vonalat (az ábrán *search region*), amitől legtávolabb eső pontot vizsgáljuk. Ha ez távolabb található a ponttól, mint a tolerancia (*tolerance*), új végponttal folytatódik az algoritmus, ha megegyezik vele, vagy kisebb, a kulcspont és végpont között található összes pont törlésre kerül. Ez addig ismétlődik, amíg a kulcspont és a végpont között található pont (Curve simplification turk project).

Boyle előrettekintő algoritmus

Az algoritmus a vizsgált pont előtt található megadott számú, és az utána található pont alapján számítja a pont feltételezett helyét, így simítva a vonalat. Sajnos, ez sokszor a vonal futásával ellentétes vonalat eredményez (14. ábrán vastagított kék vonalrajz), ezért részletesen nem vizsgáltam működését az adatsoraimon (Grass GIS Manual).



14. ábra: A Boyle algoritmus eredménye, a görög szigetek példáján

Jenk-algoritmus

Az algoritmus az adatokat csoportokba rendezi, úgy hogy ezekben a csoportokban található adatok, és azok értékei a lehető legmegfelelőbb elrendezésben szerepeljenek. Ez az egyes csoportok átlagos középértékétől való eltérés minimalizálása mellett az egyes csoportok középértékeinek maximalizálására törekszik.

Hermite spline interpoláció

A Hermite-görbék a Bézier-görbékhez hasonló vonalközelítő módszerek. A Hermite-görbék harmadfokú polinomok, amelyek adatai irányvonalakkal vannak megadva. (Kovács, 2011)

McMaster távolsággal súlyozott és csúszó-átlagoló algoritmus

A csúszó-átlagoló algoritmus (egyes helyeken csak McMaster-algoritmus) a vizsgált pont, és a szomszédos pontjainak x és y koordinátáit felhasználva egy átlagot számít. Ez az átlag a vizsgált pont új helye, de egy áthelyezési értéknek megfelelően ezt

a pontot az eredeti helyhez közelebb helyezi el az algoritmus. Eredményeképp a pont közelebb kerül szomszédjaihoz. A távolsággal súlyozott változat a pont új helyet súlyozott reciprok távolság alapján átlagolja (FME Transformers).

A partvonalak generalizálása

Európa partvonalának generalizálását ugyanazon az adatsoron végeztem, mint az előző plugin esetén, így az esetleges különbségeket is megfigyelhettem. Mivel ez vonalas elemek egyszerűsítésére használható, nem található benne a topológiát ellenőrző művelet, így a partvonalakkal együtt nem alkalmas az országhatárok egyszerűsítésére is. Először a Douglas–Peucker-algoritmust futtattam le az adatsoron, a másik pluginnál is használt küszöbértékekkel. Az első érték $0,0001^\circ$ volt, ami kevés változást hozott. Néhány hosszú, de keskeny objektum, például öböl vagy sziget egyszerűsítése, csúcsaik levágása történt meg. Mivel ez nem jelentős változás, a küszöbértéket tízszeresére, $0,001^\circ$ -ra emeltem. Ekkora értékváltozás nem befolyásolta az egyszerűsítés eredményét, ugyanazt az eredményt kaptam, mint az algoritmus első futtatásakor. A küszöbértéket ismét tízszeresére emeltem. Ezzel az értékkel már elkezdődött az ívek jelentősebb egyszerűsítése, a szigetek sokszöggé alakulása. A keskeny öblök lerövidültek, és néhány kisebb sziget már csak egy vonal formájában jelenik meg. Összességében nagyon jól megtartotta a vonal eredeti futását, azt megfelelően egyszerűsítette. A generalizált vonalrajzolat egy kb. 1:5 000 000 és 1:6 000 000 közötti méretarányú térképi alaphoz felel meg. Manuális javításra szükség van, simítás, a kisebb szigetek eltávolítása, és a bezáródó öblök, fjordok esetén a pontok áthelyezése, a végük lekerekítése szükséges. A Reumann–Witkam algoritmust a plugin által engedett legkisebb küszöbértékkel, $0,0001^\circ$ -kal futtattam le. Az adatsoromra, főleg kisebb szigetes, és tagolt partvonalas részeken ez

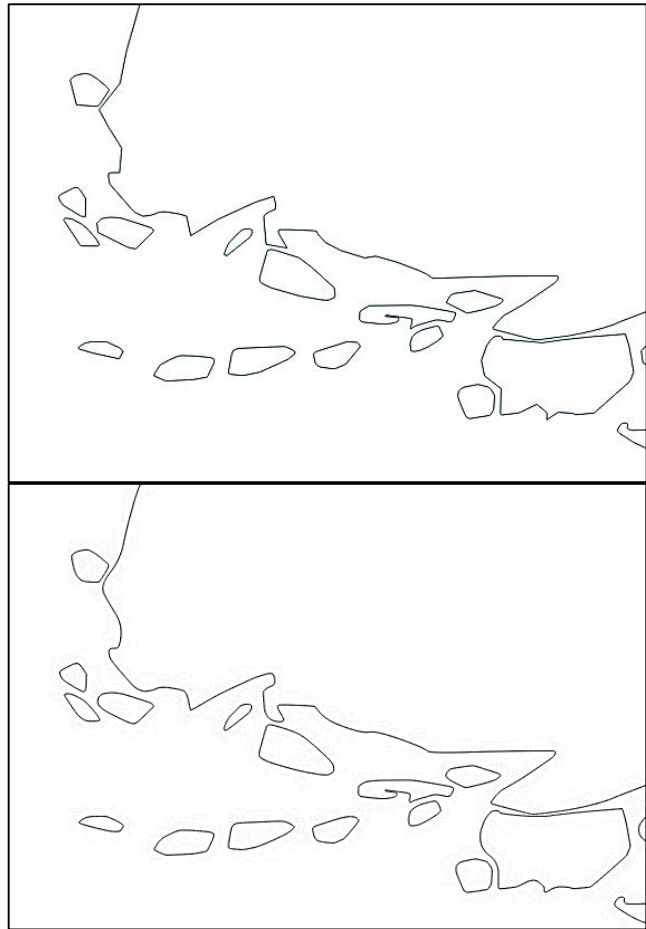
az algoritmus nem megfelelő, a partvonalak rendkívül sokszögesekké váltak, a szigetek többsége pedig egy vonallá alakult. Ezek az elemek egymásba csúsztak, rendkívül torz képet eredményezve.



15. ábra: Reumann-Witkam algoritmus futtatásának eredménye a Görög-szigetekenél

Kipróbáltam az operátor által kínált legegyszerűbb, a csomópontcsökkentő (*Vertex Reduction*) algoritmust. Ismét a legkisebb beállítható küszöbértékkel dolgoztam, ami $0,0001^\circ$ volt. Az algoritmus végrehajtása gyors, de az általa nyújtott eredmény nem megfelelő, hiszen a vonalfutást nem követi, csak a pontok távolsága alapján egyszerűsít. A szigetek többsége háromszögekké, vonalakká vált, és a partvonalak is nagyon eltorzultak. Ez az algoritmus nem megfelelő partvonal egyszerűsítésére, javítása nagyon nagy munka lenne.

A Chaiken-algoritmust is megpróbáltam az adatsoron. Legelőször a szintet (*Level*) egyre állítottam. Az élek elsimítása ennél az értéknél elkezdődött, a sarkokat kis mértékben levágta. Kettes és hármas szinten már szép, lekerekített minden él. Úgy vélem, hogy egy simító algoritmus használatára egy kiindulási adatsoron még nincs szükség, bár még esztétikusabb vonalrajzot eredményez. Egy nagyobb mértékben generalizált rajzolat simítása során pedig létrejöhetnek nem kívánt torzulások. Az európai országok határait nagyon szépen elsimította, így talán határrajz generalizálására megfelelőbb lehet.



16. ábra: Finnország délnyugati sarkának eredeti (felső ábra), majd Chaiken algoritmus hármas szintjével simított rajzolata (alsó ábra)

A Lang algoritmust $0,001^\circ$ küszöbértékkel, és 8 pont távolsággal próbáltam először. Az egyszerűsítés láthatóan elkezdődött, sok helyen, az új körvonalak nem különböznek jelentősen az eredetitől, azt szépen követik. A rajzolat nem lett szögletes, így akár az eredeti adatsorral közel megegyező méretarányban is használható lehet. A következő küszöbérték, az első tízszerese volt, a távolságot nem változtattam. Az már észrevehetőbb egyszerűsítést hajtott végre, de a rajzolat még mindig természetes hatású, és nagyon jól követi az eredeti vonal futását. Kis mértékben kezd szögletesedni, és az

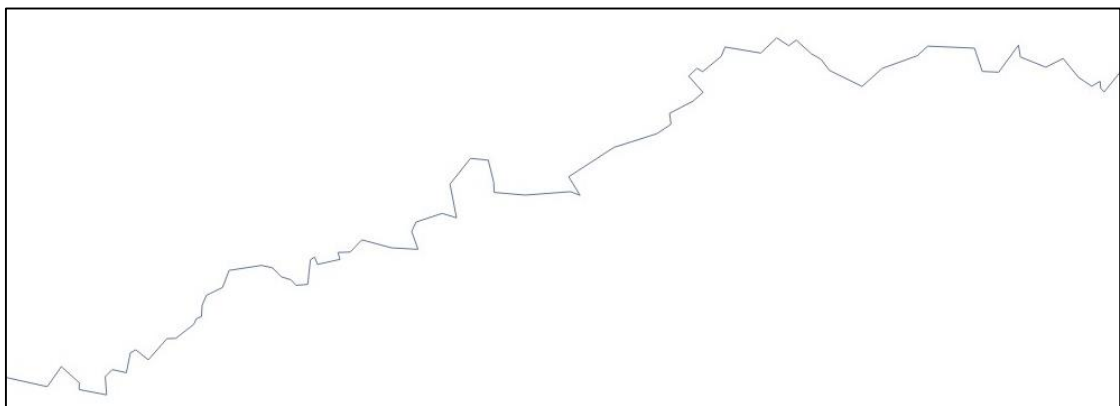
apró szigetek formája már eltorzul, de összességében nagyon jól használható az eredmény. Manuális javításra nincs szükség, esetleg a szigetek közötti területek és a fjordok kiszélesítésére, ha a méretarány azt kívánja, ami ideálisan 1:4 000 000 és 1:5 000 000 között lehet. Az említett algoritmusok eredményei a 4. számú mellékletben láthatóak.

Országhatár és megyehatárok generalizálása

Magyarország határát először a Douglas–Peucker-algoritmussal egyszerűsítettem. A legkisebb küszöbérték, amit kipróbáltam 100 méter volt. A vonal futását jól megtartva kezdődött el az országhatár egyszerűsítése, a széles sarkok, kanyarok levágása kezdődött el, de a pontok száma láthatóan csökkent. A Duna folyását jelentősen nem változtatta meg, jól megtartotta természetes jellegét, a déli határrészen pedig semmi változást nem hozott. Ez a küszöbérték csak kicsi méretarány csökkentésnél lenne megfelelő, mivel nem hozott nagy változást. Ezután a küszöbértéket kétszeresére emeltem, 200 méterre. Itt a pontok csökkentése már jelentősebb mértékben elkezdődött, egyre jelentősebb kanyarulatok, élesebb ívek levágása történt meg, és nagyobb ívszakaszokat helyettesítette az algoritmus egy-egy vonallal. Néhány jellemzőbb ívet is levágott, de ezek kézzel javíthatók, mivel mennyiségük nem nagy. A Duna természetes folyását már elkezdte eltorzítani, szögletessé formázni, de az is könnyen javítható, simítható. A déli, Dráva környéki határszakasz egyszerűsítése is elkezdődött, de még nem olyan jelentős mértékben. Ez az egyszerűsítési mérték a jelentős ívek javítása után egy 1:600 000-1:900 000 méretarány körüli térképalapnak megfelelő. A következő küszöbérték az eredeti érték ötszöröse, 500 méter volt. Ennek eredménye már egy jelentős mértékben egyszerűsített vonal, szinte már minden vonalszakasz egyszerűsítése megtörtént. Sok helyen már eltorzultak a jellegzetes kanyarulatok, már a nagyobb ívek kiegyenesítése is megtörtént, de az országhatár fő futását rendkívül jól megtartotta. A Duna vonala már nem olyan természetes, és a Dráva környéke kicsit eltorzult. Ezeknél néhány pont áthelyezése, és új beszúrása, szükséges. Ezek javítások, és az egész határvonalon végigfuttatott simító algoritmus után ez a vonalrajzolat egy 1:1 000 000-1:1 800 000 méretarányú térképével hasonlítható össze. Az utolsó küszöbérték, amivel a Douglas–Peucker-algoritmust lefuttattam 1000 méter volt. Ez az előzőnél is nagyobb mértékű generalizálást hajtott végre, a jelentősebb íveket levágta, vagy alapvonalukat egy vonallal helyettesítette. Az előző eredményekkel összehasonlítva rendkívül egyszerű a rajzolata, de az országhatár főbb futásirányát még így is megtartotta. A Duna vonalának

természetesebbé tétele, a déli határvonal torzult rajzolatának kijavítása, és néhány fontosabb határvonal részlet visszahelyezése után, egy simítást követően 1:2-3 000 000 méretarányú térkép határvonalának megfelelő lenne. Az algoritmus előnye, hogy fokozatosan távolítja el a részleteket, a jelentéktelenebbtől a fontosabbak felé haladva. Emellett megtartja a határvonal főbb futását, így simítás és hibajavítás után megfelelő rajzolatot kapunk.

A második algoritmus a Reumann–Witkam algoritmus volt. Ugyanazon küszöbértékekkel futtattam le, mint előzőleg a Douglas–Peuckert, így a változások mértékét össze lehet hasonlítani. A 100 méteres küszöbérték eredményeképp létrejött vonalrajz jelentősen nem tér el az eredetitől. A pontok csökkentése megkezdődött, de a jelentékenyebb ívek eltávolítása, esetleg pontok áthelyezése történt csak meg. Mivel nagy változás nem történt, a rajzolat az eredeti rajzzal nagyjából megegyező méretarányban alkalmazható. A második küszöbérték az első kétszerese, 200 méter volt. Itt már több széles ív egyenessé alakítása, kisebb, jellemzőbb ívek levágása megtörtént. Úgy tapasztaltam, hogy a vonalat nem egységesen egyszerűsíti, néhol nagyobb mértékűnek tűnik. A vonal futását jól megtartja, és túl sok fontos elemet nem távolított el. A Duna folyását jól közelíti. A határ futása simábbnak látszik, de a részletek eltávolítása miatt az eredetinel mindenképpen kisebb, nagyjából 1:600 000 méretarányban lenne megfelelő. A következő küszöbérték az eredeti ötszöröse, 500 méter volt. Ez már eltorzítja egy kicsit a határvonalat, az ívek levágása nem mindig követi a főbb vonalfutást, és több, jelentősebb ívet is eltávolított, amit generalizálás után is szükséges lenne a határrajzolatban hagyni. A Duna folyását túlságosan leegyszerűsítette, nagyon egyenes lett, ezt kézzel mindenképpen javítani kellene.

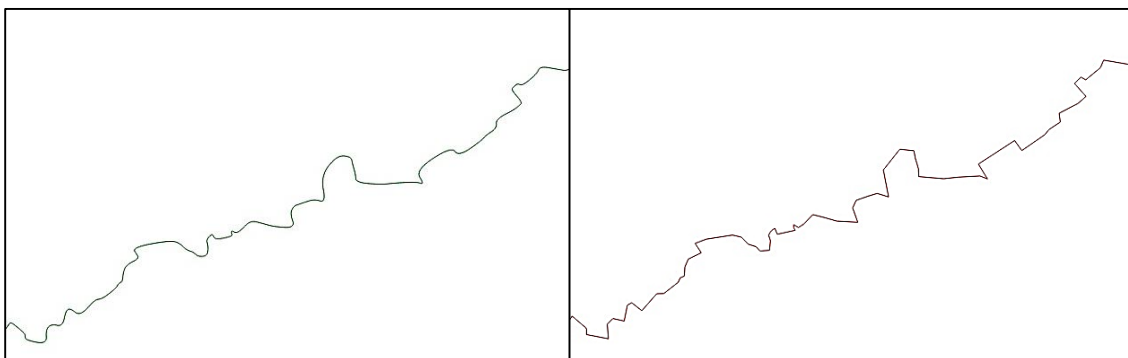


17. ábra: A nógrádi határszakasz 500 méteres küszöbértékű Reumann-Witkam algoritmussal egyszerűsített rajza

A legnagyobb küszöbérték 1000 méter volt. Ez egy rendkívül egyszerű vonalrajzolatot hozott eredményül. A fontosabb ívek, sarkok is eltávolításra kerültek, és a vonal nem hú a vonal fő futásához. A Douglas–Peucker-algoritmus 500 és 1000 méteres küszöbértékkel való lefuttatásával sokkal tisztább, és a határ jellegéhez hübb rajzolatot kaptam. Úgy gondolom, hogy az előbbi algoritmus sokkal megfelelőbb erre a célra, kevesebb utómunkát is igényel. Az algoritmus eredményei az % . számú mellékletben található.

A többi algoritmust nem ilyen részletességgel próbáltam ki. Ezek egyike az egyszerű csomópontcsökkentő (*Vertex Reduction*) algoritmus, 1-es küszöbértékkel. Mivel ez nem tartalmaz a vonal futását figyelembe vevő formulákat, az eredmény nem hú az eredeti határhoz. Már ezzel az értékkel is jelentősen egyszerűsített rajzolatot kaptam, aminek összetettsége nagyjából a 1000 méteres küszöbértékkel futtatott Douglas–Peucker-algoritmussal egyezik meg.

A Chaiken-algoritmus három szinten futtattam le, egytől háromig. Az algoritmus első két lefuttatása már kicsit simább rajzolatot eredményezett, de a csomópontok túlságosan eltávolodtak az eredeti hatástól, így eltorzítva azt. Ezzel szemben az algoritmus harmas szinten futtatva már nagyon szép, simított határvonalat eredményezett, ami teljesen hú volt az eredeti rajzolathoz, csupán a sarkok lekerekítése történt meg. Úgy gondolom, határvonalak generalizálásához ez a módszer mindenképpen szükséges. Az algoritmussal egyszerűsített vonalak sok helyen túlságosan szögletesek, ezért az eredetitől nagyobb mértékben különböznek. Azonban ha a simítást is alkalmazzuk rajtuk, jobban közelítik az eredeti vonalat.



18. ábra: Az országhatár eredeti, majd harmas szintű Chaiken algoritmussal egyszerűsített rajzolata

A Generalizer plugin algoritmusai más közigazgatási határok egyszerűsítésére is alkalmazhatók, mint például a megyehatárok. Ha ezek egyedül álló határvonalak, nincsen gond az egyszerűsítés minőségével. Egy Magyarország megyéit ábrázoló adatsor egyszerűsítésére már nem megfelelő, mivel nem poligonok, hanem vonalak egyszerűsítésére szolgáló műveletek találhatók benne. A topológia figyelésére és megtartására nincs lehetőség, ezért csak vonalas elemek generalizálására megfelelő a plugin. Az algoritmusok eredményei (az előző ábra nagyobb megfelelőjével együtt) a 6. számú mellékletben található.

ArcGIS

Az ArcGIS az ESRI térinformatikai szoftvere, mely adatok szerkesztésére, módosítására és térbeli lekérdezések, vizsgálatok végrehajtására szolgál. A szoftver különböző moduljai közül a Cartography Toolbox (Térképészeti eszköztár) Generalization csoportja tartalmaz különböző, generalizálásra alkalmas műveleteket. A munkám során ebből négy használatát ismertem meg: Simplify Line, Smooth Line, Simplify Polygon és a Smooth Polygon operátorok. A vizsgált adatsor Magyarország határa volt, méretaránya 1:300 000-1:400 000 körüli volt.

Simplify Line

Egyszerűsíti a vonalakat felesleges ívek eltávolításával, de a vonal fő futásának megtartásával. Két egyszerűsítő algoritmus közül választhatunk. A pontok eltávolítására alkalmazott Point remove gyorsabb, de kevésbé finomított eredményt hoz. Eltávolítja a nélkülözhető pontokat, a Douglas–Peucker-algoritmus használatával. Az adatsor nagyságának csökkentésére, durvább egyszerűsítésre alkalmas. Az ívegyszerűsítő Bend simplify lassabb, de az eredeti vonalhoz hű eredményt hoz létre. A Wang-algoritmus használatával eltávolítja a jelentéktelen íveket, finomabb vonalfutást eredményezve.

A Wang-algoritmus a vonalakat ívek sorozatának tekinti, amiket az inflexiós szögek változása határoz meg. Minden ívnek jellemzője a méret, forma, így különböző típusú ívek vannak. Az algoritmus futásához egy küszöbértéket kell beállítani, ami egy félkörív minimális átmérője. Egyenként kezeli az íveket, helyi minimumokat, azaz az öt körülvevő íveknél kisebb íveket választ kis, és a toleranciaértékhez hasonlítja. Ha ennél nagyobb az íve, változatlanul hagyja, ha kisebb, egyszerűsíti. Az egyszerűsítésnek három formája van: törlés, összevonás és nagyítás. (Wang és Müller, 1998)

Beállítható az algoritmusokhoz használt küszöbérték, amit bármilyen általunk választott mértékegységben megadhatunk. Ugyanazzal az értékkel a ponteltávolítás módszere jóval egyszerűbb eredményt hoz. Két réteg jön létre automatikusan, az egyszerűsített vonalrajzolat és egy pont geometriájú réteg, ami az esetleges nulla hosszúságúvá vált szakaszokat tartalmazza. Alkalmazható topológiai hibák, vagyis az átfedések, és a vonalak közötti hézagok keresése, ilyenkor a hibajavítás is automatikusan megtörténik az algoritmus lefuttatása közben (ArcGIS Help).

Simplify Polygon

Poligonok egyszerűsítésére szolgál, felesleges íveinek, pontjainak eltávolításával. A Simplify line művelethez hasonlóan a Douglas–Peucker-algoritmus használja fel a Point remove, a Wang-algoritmust a Bend simplify algoritmus végrehajtásához és hasonló paraméterekkel működik: beállítható a küszöbérték és a minimálisan megjeleníthető terület. Minden poligon, ami nem éri el ezt a küszöbértéket, eltávolításra kerül. A topológia vizsgálatnál három különböző módszer választható: ne vizsgálja a topológiát, jelölje a hibákat, és javítsa a hibákat (ArcGIS Help).

Smooth Polygon

Az éles kanyarok, sarkok elsimítására használt eszköz, hogy esztétikailag és kartográfiaiilag elfogadhatóbb minőségű vonalat hozzunk létre. Két algoritmus közül választhatunk. A *PAEK* -Polynomial Approximation with Exponential Kernel-algoritmus a korábban már említett Chaiken-algoritmushoz hasonlóan, a pontok növelésével simítja el a sarkokat ívekké. A simítási küszöbérték (*Smoothing Tolerance*) megadja az új pontok egymástól való távolságát, minél kisebb az érték, a végeredmény annál részletesebb lesz, de lassabb a feldolgozási ideje. A másik módszer, *Bézier Interpolation*, Bézier-görbékkel közelíti a simítani kívánt adatsort, toleranciaérték megadása nélkül. Mivel a görbék nem tárolhatók shapefájlban, csak a közelítésüket tároljuk. Topológiai hibák megjelenhetnek, ezeket a végrehajtás után megjelölheti a program, de nem javítja ki. Megtörténhet, hogy egy poligon metszi saját magát, ilyenkor a javítás után ketté válik (ArcGIS Help).

Országhatár és partvonalak generalizálása

A Magyarország határát ábrázoló adatsoromon először a Simplify Line algoritmusait futtattuk. Amellett, hogy megfelelően egyszerűsíti az ország határát, a jellegét meghagyva, kíváncsi voltam arra is, hogy a Duna folyásának irányát milyen mértékben befolyásolja. A művelet nem működik megfelelően, ha az Arc Toolbox-ból indítjuk (duplakattintás a menüsorban), a különböző értékek beállítására nincs lehetőség. Ezt a hibát oly módon lehetséges kikerülni, hogy jobb egérgombbal a menüsorban rákattintunk, majd Batch módban indítjuk el a műveletet. Így más formában jelenik meg a kommunikációs ablak, és lehetőség van a paraméterek beállítására.

Elsőként a Bend Simplify algoritmust futtattuk az adatsoron, 1000 méteres küszöbértékkel. Ez az érték kis méretarány különbség esetén megfelelően alkalmazható lehet, mivel nem módosítja jelentősen a vonal futását, megőrzi jellegét. Az élesebb sarkok kis mértékben lekerekítettek lesznek, ezeket az algoritmus levágja, jelentősebb változás nem történik.

2000 méteres küszöbértékkel már jóval szembetűnőbbek a változások. Sok fontosabb kanyarulatot eltávolított az alkalmazás az adatsorból, így már nem megfelelő kismértékű méretarány változtatás esetén. A vonal főbb futását követi, és sok esetben a jelentősebb görbületeket is levágja, mivel ezeket is zajnak tekinti. Ez a küszöbérték már csak nagyobb méretarány változás esetén lenne használható, de úgy is szükség lehet a végeredmény manuális javítására. Azon a szlovák–magyar határszakaszon, ahol a Duna folyik, a vonal futásában jelentősebb változást nem okozott, de néhány szélesebb kanyarulatot levágott.

A Point Remove algoritmus már kisebb küszöbértékekkel is jól látható eredményt hoz. Előfordulhat, hogy olyan pontokat távolít el, amelyek esszenciálisak a vonal futásában, mivel elsődleges célja az adatsor nagyságának csökkentése, a pontot csökkentésével. Az adatsoron 100 méteres küszöbértékkel lefuttatva szembetűnő változás nem volt. A szélesebb íveket több helyen leegyszerűsítette, a benne található pontok csökkentésével, így egy ív helyett csak egy sarok lett belőle, de nagy és indokolatlan változást sehol sem okozott. Úgy vélem, hogy ezzel a küszöbértékkel megfelelő lehet a fő céljára, a vonalas elemet alkotó pontok számának csökkentésére, de mivel jelentősebb változást nem eredményez, csak kisebb méretarány változtatás esetén lenne használható. A zajokat megfelelően kiszűri, a kisebb változásokat rendkívül jól elsimítja.

A következő próbálkozás a Point Remove algoritmus volt, 300 méteres küszöbértékkel. A 100 méteres küszöbértéknél észlelt változásoknál jóval több változást hozott az algoritmus futtatása. Az országhatár vonalának főbb futását meghagyta, de a vonal futásában való kisebbnek tűnő változásokat elhagyta. Ezek legfőképpen az ország északi határánál érzékelhetők. Sok kisebb kanyarulatot levágott és egy hosszabb vonalszakasszal helyettesítette őket, a vonalfutás jellegzetes, ám kisebb részeit elhagyva.

A Simplify Polygon műveletsort egy Magyarország megyéit ábrázoló adatsoron teszteltük. A főbb hangsúly itt is az országhatáron volt, de a topológiai hibák ellenőrzése céljából a megyehatárokat is megfigyeltük. Az először a Bend Simplify algoritmust futtattuk 1000 méter küszöbértékkel, így lehetőségem van összehasonlítani a vonalegyszerűsítés eredményeképp létrejött vonallal. A két művelet elvégzése rendkívül hasonló vonalat eredményezett, egy különbséget találtam csupán: a Duna és az Ipoly összefolyásánál.

Az így keletkezett vonalat felhasználva teszteltük a Smooth Polygon művelet PAEK algoritmusát, szintén 1000 méteres küszöbértékkel. Az eredményképp kapott poligon főbb vonalfutása megegyezik az eredetivel, de annak élei jelentős mértékben elsimítottak. A főbb formákat ugyanúgy megőrizte, de az elsimított éleknek köszönhetően kevésbé zsúfolt és jóval esztétikusabb a végeredmény. A legfontosabb jellemzője az, hogy az eredeti adatsorhoz képest, a Duna folyása ezen tűnik legtermészetesebbnek, és a legközelebbi eredménynek az eredetihez, a lekerekített élek miatt.

A partvonalak generalizálását egy, az Adriai-tenger partvonalát ábrázoló adatsoron próbáltuk ki. A generalizáláshoz elsősorban egyszerűsítő algoritmusokra van szükség, nem pedig simítást végrehajtó műveletekre. Így tudjuk a sok részletet kiszűrni, és a részletes adatsort kisebb méretarányban felhasználni. Sokszor szükség lehet a túl kevés pontból álló poligonok törlésére, ha a célméretarány, és a sziget jelentősége megengedi. A kisebb lagúnák, szorosok bezáródhatnak. Mivel ezek navigációs jelentősége elég nagy, nem távolíthatjuk el őket a térképről. Ilyenkor az eredmény kézzel történő javítására van szükség: az egymáshoz túl közel került szigeteket messzebb kell helyezni egymástól, esetenként a túl kicsi szigetek nagyítására is szükség lehet.

7. Kitekintés: egyéb térképi elemek generalizálása

A határvonalak generalizálása során találkozhatunk speciális esetekkel, amikor egy természetes képződmény, folyó vagy tó alkotja a határvonalat. Ilyenkor ezeket külön kell venni a határ többi, mesterséges részétől, hiszen generalizálásához más eszközök szükségesek.

Folyók esetén figyelni kell arra, hogy a vonal a lehető legtermészetesebb, és az eredetit közelítő maradjon, még kisebb méretarányokban is. Ha a határral együtt kezeljük, javítással szükséges az eredetit minél jobban közelítő vonalrajzot létrehozni. A tavak generalizálásának automatizálása nehézkes feladat. Ennek oka a sok mesterséges eredetű tó, a bányatavak, halász- és horgásztavak és a folyószabályozások eredményeképpen létrejött holtágak. Ezek formája többnyire elég szabályos, így a rajtuk végigfuttatott generalizáló algoritmusok eltorzíthatják az eredeti formájukat. Mivel így nem lehet az összes objektumra egyszerre felhasználni az algoritmust, több lépésre van szükség. A természetes eredetű vizek partvonalának egyszerűsítését végezhetjük egy ponteltávolító algoritmussal, hogy csökkentsük csomópontjainak számát, a mesterséges tavaknál mindenképpen manuális generalizálásra van szükség. Ezeket összevonhatjuk, esetleg távolabb helyezhetjük egymástól. A holtágakat több esetben eltávolíthatjuk, nagyíthatjuk, összevonásuk nem hoz megfelelő eredményt.

8. Összegzés

Dolgozatom során megismertem a kartográfiai generalizálás lényegét, szabályait és alapvető műveleteit. Ezután az automatizált generalizálás alapjaival, az adatbázisokhoz, térképrendszerekhez használt különböző modellekkel ismerkedtem meg, melyeket munkámban be is mutattam. Ezek alapvetően befolyásolják a generalizálás folyamatának felépítését, a felhasznált műveleteket és azok sorrendjét.

A generalizálás alapvető műveleteit automatizált környezetben is bemutattam, példákkal illusztráltam, külön raszteres és vektoros adatrendszerekre. Ezután a képzés során általunk használt két térinformatikai szoftver, a QGIS és ArcGIS generalizálásra alkalmas moduljait elemeztem. Az általuk használt algoritmusok leírása, bemutatása után kettő, egy Európát, és egy Magyarország határvonalát ábrázoló adatsoron teszteltem a rendelkezésre álló lehetőségeket. Az adatokat különböző algoritmusok használatával egyszerűsítettem, hibáikat, előnyeiket és hiányaikat figyelembe véve. Munkám során talákoztam több hibával is. A QGIS esetén több esetben nem futott le megfelelően az algoritmus, míg az ArcGIS esetén a műveletek indítása, az értékek megadása okozott gondot.

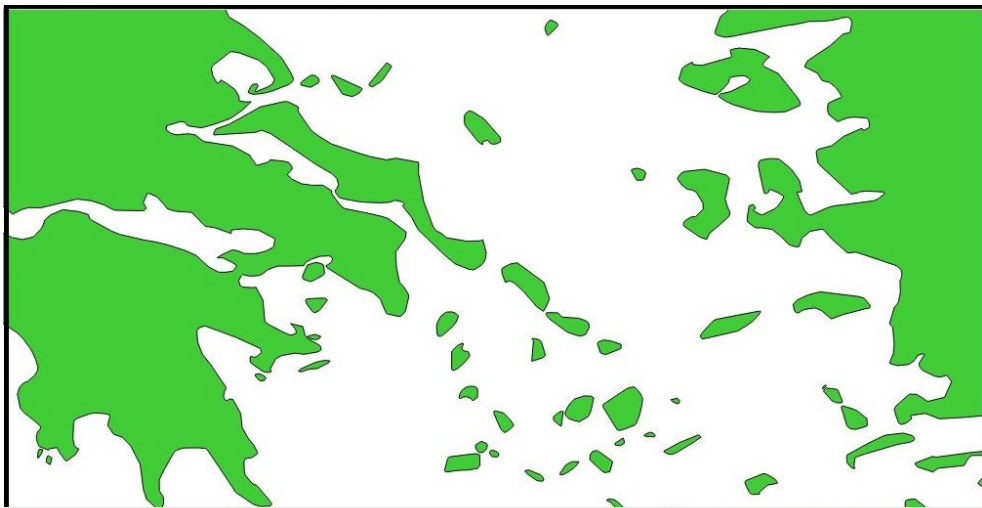
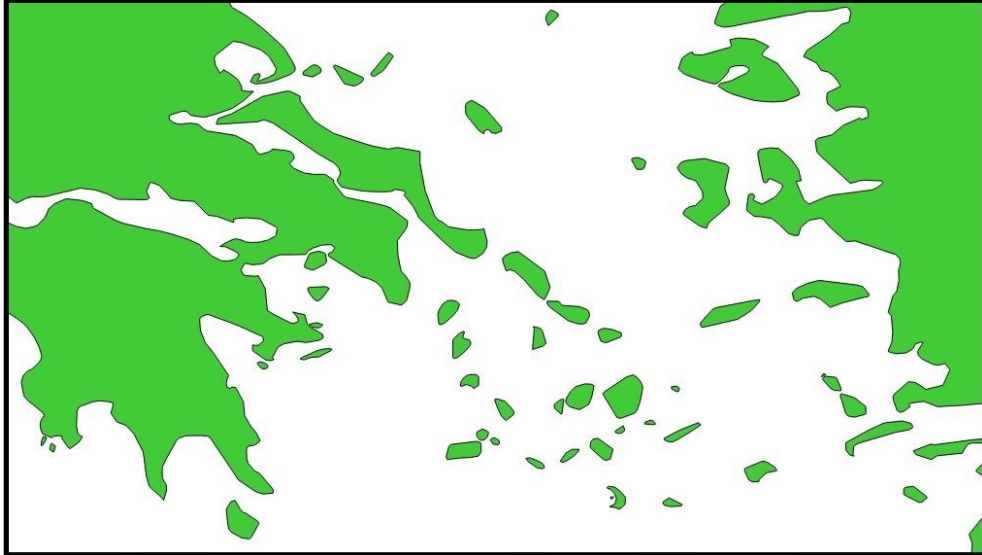
Az automatizált generalizálás legnagyobb nehézsége, hogy nem cél méretarányokkal, hanem az algoritmusokhoz használt küszöbértékekkel dolgozunk. Így szükségünk van kész térképekre, vektoros adatokra, amikhez a generalizált vonalrajzot hasonlítani tudjuk, hogy a méretarányát – legtöbb esetben méretarány tartományát – megadhassuk. Mivel ilyen ellenőrzésekre, és sok esetben kézzel történő javításra van szükség, a kézzel történő generalizálás nem váltható ki teljesen. Ezek ellenére az automatizált generalizálás rendkívül hasznos, elsősorban legnagyobb előnye, a gyorsasága miatt.

Úgy vélem, hogy a különböző algoritmusok fejlesztésével, az algoritmusok pontosításával, és javító-kereső operátorok használatával később a generalizálás teljesen automatizálható folyamat lesz.

9. Mellékletek

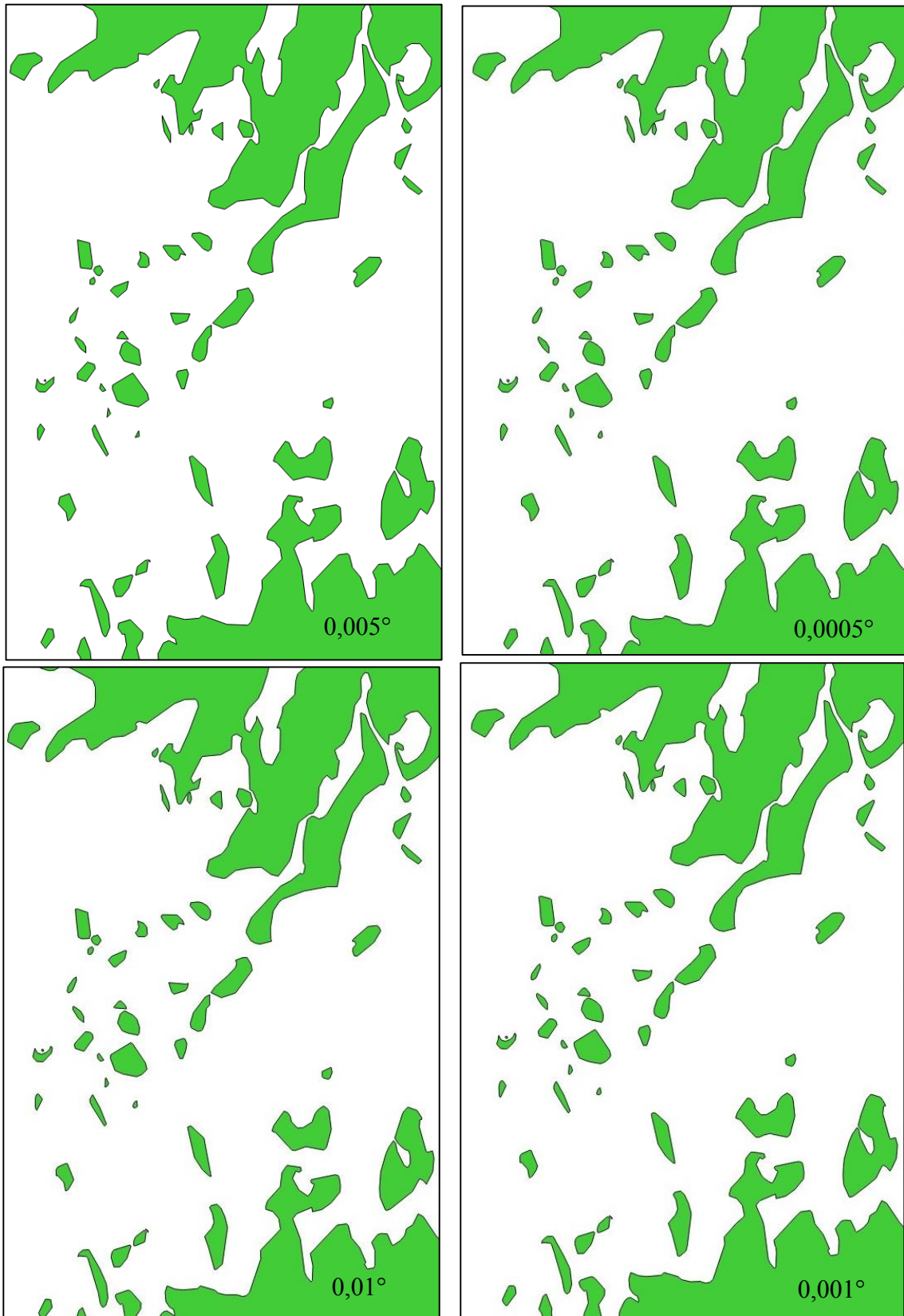
1. számú melléklet

A Visvalingam algoritmus futtatása (SimpliPy) $0,0001^\circ$, $0,0002^\circ$ és $0,001^\circ$ nagyságú küszöbértékkel



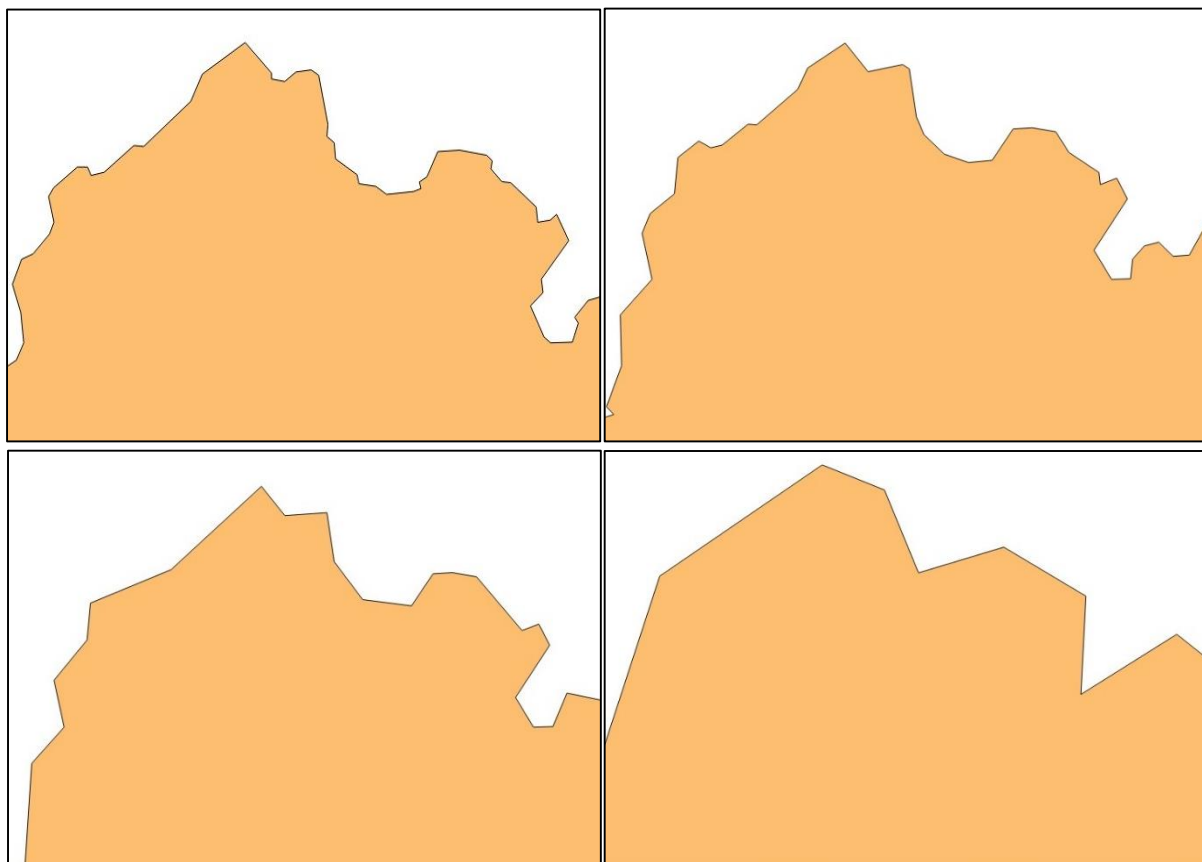
2. számú melléklet

A SimpliPy plugin Douglas-Peucker algoritmusának futtatása, Görög-szigetek

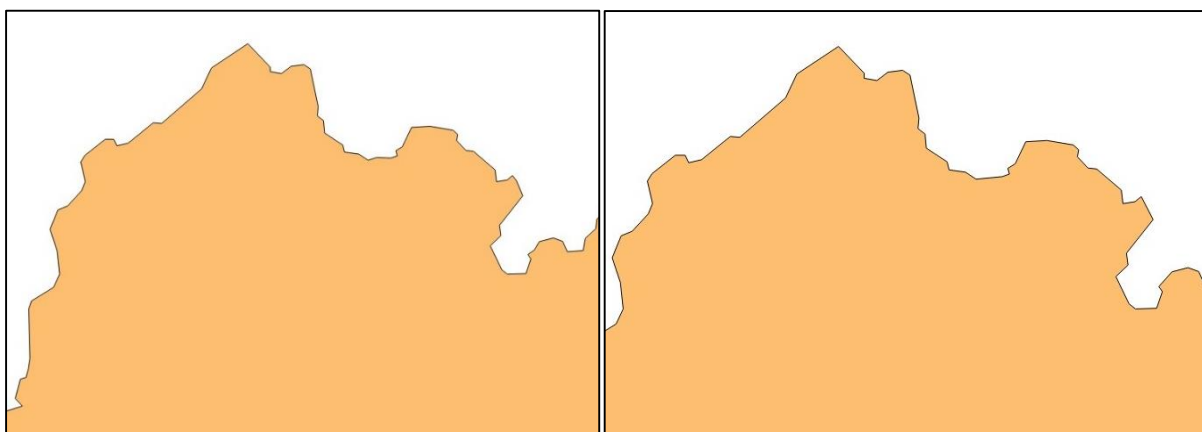


3. számú melléklet

A SimpliPy plugin Douglas-Peucker-algoritmusának használata különböző küszöbértékekkel: 100 m, 200 m, 500 m és 1000 méter



Az eredeti országhatár és a Visvalingam algoritmus eredménye 0,002°-os határértékkel (SimpliPy plugin)



4. számú melléklet

A Lang algoritmus futtatása $0,1^\circ$ toleranciával, és 8 pontos kulcsértékkel (Generalizer)

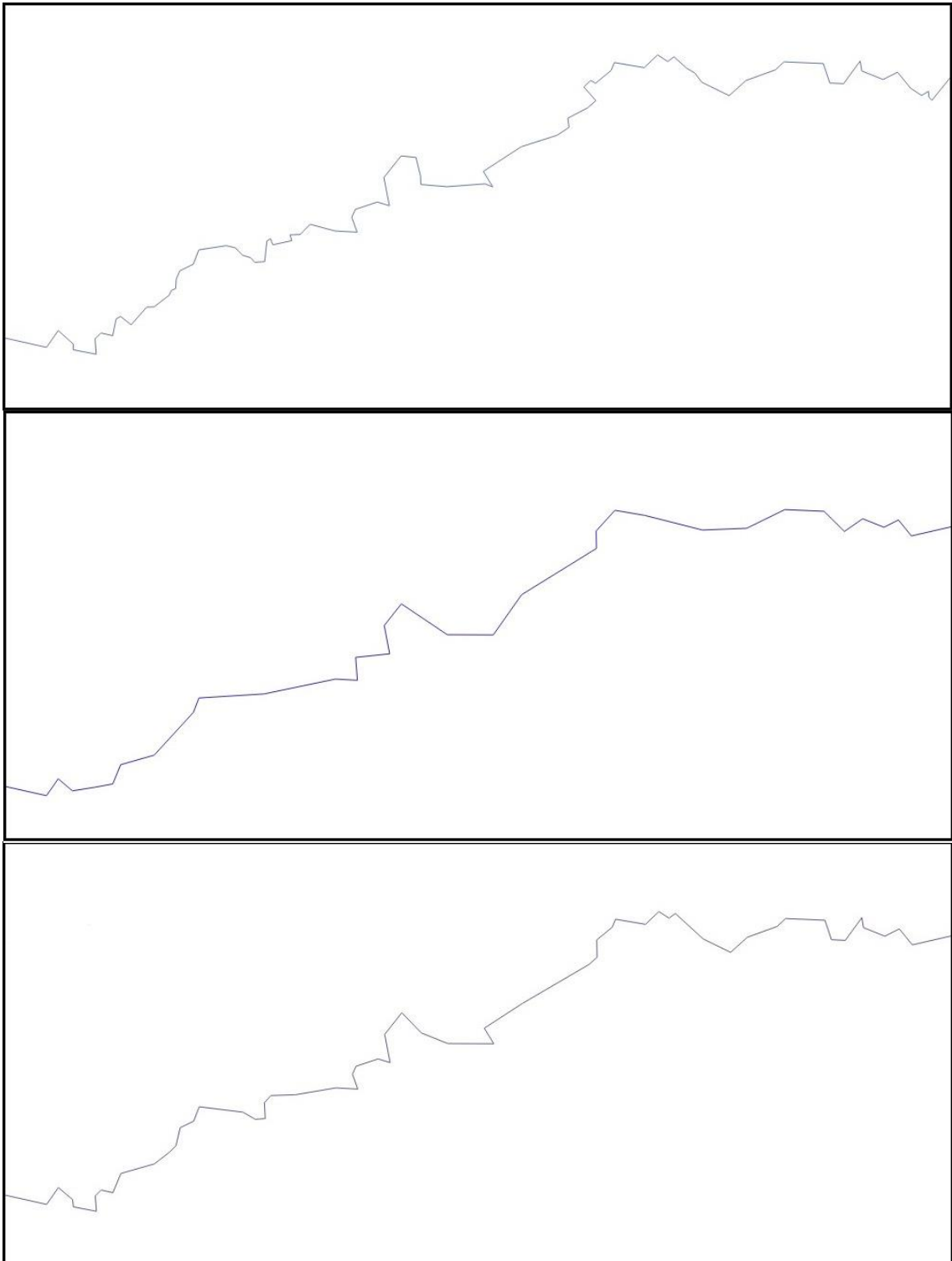


A Vertex reduction algoritmus eredménye $0,0001^\circ$ -os határértékkel (Generalizer)



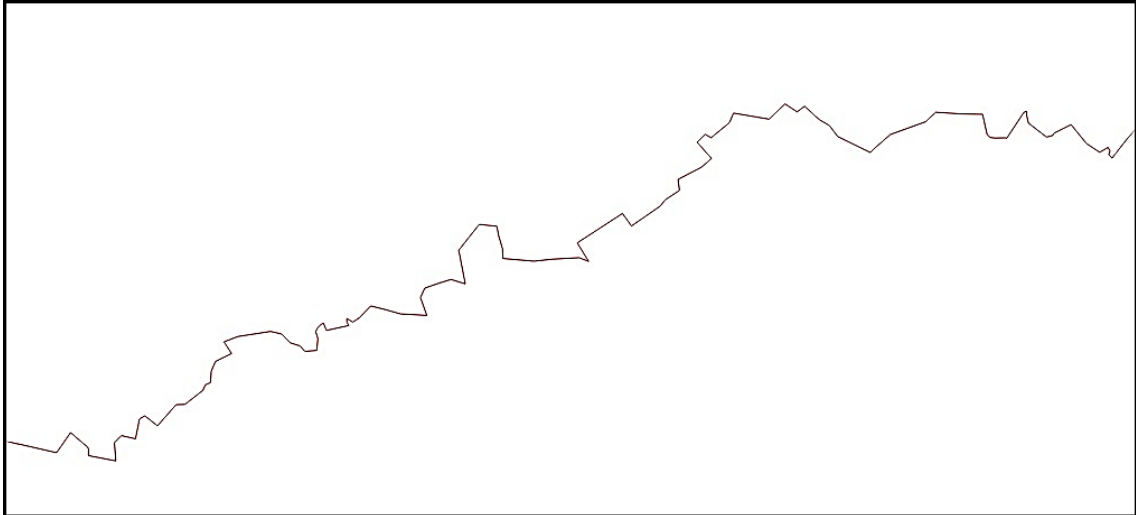
5.számú melléklet

A Reumann-Witkam algoritmus futtatásának eredményei a Generalizer pluginban
(fentről lefelé: 100 méter, 500 méter, 1000 méter)

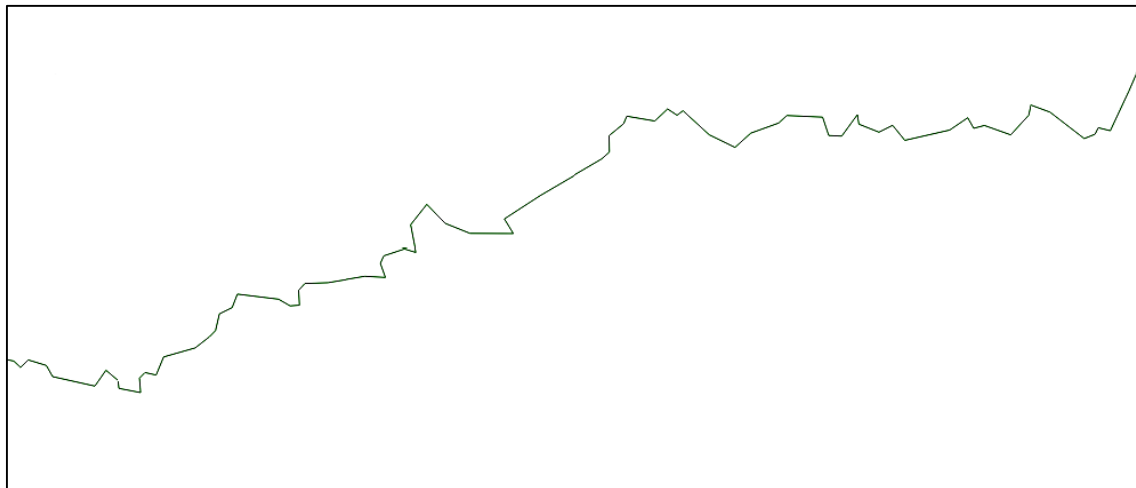


6. számú melléklet

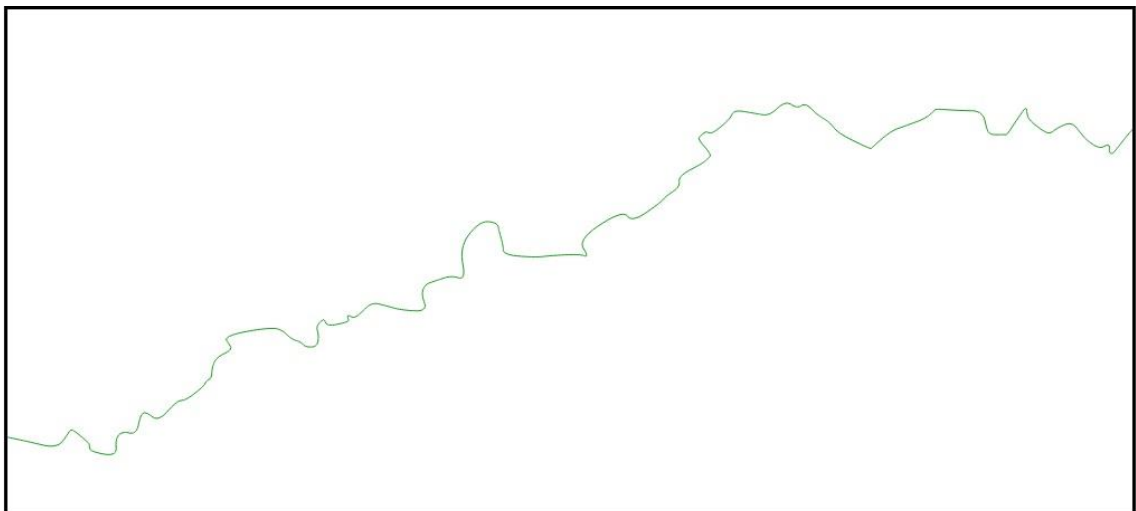
Az országhatár eredeti rajzolata (Nógrád környéke)



A csomópontcsökkentő algoritmus (Vertex Reduction) eredménye a Generalizer pluginban



A Chaiken algoritmus futtatása hármas szinten (Generalizer)



10. Köszönetnyilvánítás

Ezúton szeretném megköszönni témavezetőmnek, Ungvári Zsuzsannának a kezdeti útmutatást a téma kiválasztásában, és munkám folyamán a hasznos tanácsokat, biztatást, és a javítást. Köszönöm Agárdi Norbertnek, hogy lehetőséget nyújtott az ArcGIS moduljainak megismerésére, és tanácsokkal látott el a generalizálás folyamatát illetően.

11. Irodalomjegyzék

Bardócz, 1970: Bardócz László (1970): A generalizálás lényege és automatizálása. In: Dr. Stegena Lajos (szerk.): Térképi generalizálás, ELTE Természettudományi Kar, Budapest, pp.15-21.

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

Cecconi, 2003: Alessandro Cecconi (2003): Integration of Cartographic Generalization and Multi-Scale Databases for Enhanced Web Mapping, doktori disszertáció, Universität Zürich

Daoud és Doytsher, 2008: J. Joubran Abu Daoud, Y. Doytsher (2008): An Automated Cartographic Generalisation Process: A Pseudo-physical Model. In: The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences. Vol. XXXVII. Part B2. Beijing, 2008 pp. 419-424.

Elek, 2006: Elek István (2006): Bevezetés a geoinformatikába, ELTE Eötvös Kiadó, Budapest

Joy, 1999: Kenneth I. Joy (1999): On-line Geometric Modeling Notes: Chaikin's Algorithm for Curves

<http://www.idav.ucdavis.edu/education/CAGDNotes/Chaikins-Algorithm/Chaikins-Algorithm.html> Utolsó elérés: 2015. 04. 22.

Kovács, 2011: Dr. Kovács Emőd (2011): Komputergrafika – Matematikai alapok on-line jegyzet, Eszterházy Károly Főiskola

<http://aries.ektf.hu/~hz/pdf-tamop/pdf-01/html/index.html>

Utolsó elérés: 2015.05.10

Klinghammer, 1977: Klinghammer István (1977): Az automatizált térképezés problémái. In: Stegena, Klinghammer, Füsü: Az automatizálás a kartográfiában II., ELTE Természettudományi Kar Tankönyvkiadó, Budapest, pp. 43-51.

Klinghammer, 2010: Klinghammer István (2010), szerk.: Térképészet és geoinformatika I., ELTE Eötvös Kiadó, Budapest

McMaster és Veregin, 1997: Robert B. McMaster és Howard Veregin (1997) Visualizing Cartographic Generalization. In: Auto Carto 13, Proceedings of the Thirteenth Annual Convention and Exposition Technical Papers, pp. 174-183.

Mélykuti, 1970: Mélykuti Mihály (1970): A kartográfiai generalizálás lényege, tényezői és módszerei. In: Dr. Stegena Lajos (szerk.): Térképi generalizálás, ELTE Természettudományi Kar, Budapest, pp. 5-14.

NCGIA Core Curriculum, 1994: Térinformatikai alapismeretek. Szerk: Márkus Béla, Márton Mátyás, Paksi Judit. 1994. 48. fejezet: Vonalgeneralizálás, ford.: Márton Mátyás.

Slocum, 2005: Terry A. Slocum, Robert B. McMaster, Fritz C. Kessler, Hough H. Howard (2005): Scale and Generalisation. In: Terry A. Slocum, Robert B. McMaster, Fritz C. Kessler, Hough H. Howard: Thematic Cartography and Geographic Visualisation, 2nd edition, Pearson Education Inc. Egyesült Államok

Visvalingam és Whyatt, 1993: Visvalingam, M. and Whyatt J. D. (1993): Line Generalisation by Repeated Elimination of Points. Cartographic Journal, 30(1) pp. 46-51.

Wang és Müller, 1998: Zeshen Wang – Jean-Claude Müller (1998): Line generalization Based on Analysis of Shape Characteristics. Cartography and Geographic Information Systems, 1998/1, pp. 3-15.

Internetes források

ArcGIS Help 10.1:

<http://resources.arcgis.com/en/help/main/10.1/index.html#/007000000010000000>

http://resources.arcgis.com/en/help/main/10.1/index.html#/Simplify_Polygon/007000000010000000/

http://resources.arcgis.com/en/help/main/10.1/index.html#/Smooth_Polygon/00700000001030000000/

Utolsó elérés: 2015.04.19.

Curve simplification turk project:

http://web.cs.sunyit.edu/~poissad/projects/Curve/about_algorithms/lang.php

http://web.cs.sunyit.edu/~poissad/projects/Curve/about_algorithms/douglas

http://web.cs.sunyit.edu/~poissad/projects/Curve/about_algorithms/reumann.php

Utolsó elérés: 2015.04.20.

FME Transformers:

http://docs.safe.com/fme/html/FME_Transformers/FME_Transformers.htm#Transformers/generalizer.htm

Utolsó elérés: 2015.05.10.

Grass GIS Manual:

<http://grass.osgeo.org/grass71/manuals/v.generalize.html>

Utolsó elérés: 2015.05.08.

QGIS Python Plugins Repository:

<https://plugins.qgis.org/plugins/generalizer/>

<https://plugins.qgis.org/plugins/simplipy/>

Utolsó elérés: 2015.04.18.

A felhasznált adatsorok forrása:

Európa: <http://www.gadm.org/>

Magyarország: régi Mapinfo adatsor

12. Ábrajegyzék

1. ábra:	6
https://www.google.com/maps (2015.05.08.)	
2. ábra:	20
Robert B. McMaster és Howard Veregin (1997) Visualizing Cartographic Generalization. In: Auto Carto 13, Proceedings of the Thirteenth Annual Convention and Exposition Technical Papers	
3. ábra:	26
4. ábra:	26
http://psimpl.sourceforge.net/douglas-peucker.html (2015.05.05.)	
5. ábra:	27
6. ábra:	28
7. ábra:	30
8. ábra:	32
9. ábra:	32
10. ábra:	34
11. ábra:	35
http://psimpl.sourceforge.net/reumann-witkam.html (2015.05.05.)	
12. ábra:	36
http://www.idav.ucdavis.edu/education/CAGDNotes/Chaikins-Algorithm/Chaikins-Algorithm.html (2014.04.22.)	
13. ábra:	36
http://psimpl.sourceforge.net/lang.html (2015.05.05.)	
14. ábra:	37
15. ábra:	38
16. ábra:	39
17. ábra:	41
18. ábra:	42
<i>Az ábrajegyzékben forrás nélkül feltüntetett képernyőképek és ábrák saját készítésűek.</i>	

Nyilatkozat

Alulírott, Kocsis Viktória (GOIPXZ) nyilatkozom, hogy jelen szakdolgozatom teljes egészében saját, önálló szellemi termékem. A szakdolgozatot sem részben, sem egészében semmilyen más felsőfokú oktatási vagy egyéb intézménybe nem nyújtottam be. A szakdolgozatomban felhasznált, szerzői joggal védett anyagokra vonatkozó engedély a mellékletben megtalálható.

A témavezető által benyújtásra elfogadott szakdolgozat PDF formátumban való elektronikus publikálásához a tanszéki honlapon

HOZZÁJÁRULOK

NEM JÁRULOK HOZZÁ

Budapest, 2015. május 15.

.....
a hallgató aláírása