Térképen található karakterek és szimbólumok felismerése és vektorizálása

Nemes Krisztián

Bevezetés

A mai fogalmak szerint a digitalizált térképek nem csak egyszerű – esetleg georeferált (koordinátákkal ellátott) – képek, hanem általában vektoros digitális állományok; az objektumaikhoz (pont, vonal, felület, szöveg) adatbázis kapcsolódik, mely tartalmazza az egyes objektumok leíró adatait (attribútumait). Egy vektoros állomány manuális előállítása azonban igen hosszadalmas folyamat. Ennek a folyamatnak a felgyorsítására készítettünk egy eljárást, ami raszteres digitális térképeken a pontszerű térképi jelek felismerését és vektorizálását automatikusan végzi el a manuális beállítások után. Ebben a cikkben bemutatjuk az eljárást, az alkalmazott algoritmust és a működés megbízhatóságára, sebességére vonatkozó vizsgálatok eredményeit is.

Lehetőségek

Az automatikus jelfelismerés területén végzett kutatások igen szerteágazók. Dinamikus rendszámtábla-felismerő szoftvereket alkalmaznak a szabálysértők gyors és automatikus kiszűrésére [1]. Különféle programok állnak rendelkezésre a szkennelt könyvek karaktereinek vektorizálására [2]. Egyes eljárások a kézzel írt karakterek felismerésére neurális hálózatokat alkalmaznak. Ezek a hálózatok az emberi agyhoz hasonlóan működnek, és az apró eltérések mellett is képesek felismerni a betűket [3]. Térképészeti vonatkozásban is használják ezt a módszert, hiszen a nyomtatott térképek szkennelésekor fellépő torzulások miatt az egyes jelek deformálódnak, ami megnehezíti a felismerésüket [4].

Valamennyi eljárás működésében kulcsszerepet játszik egy olyan háttéradatbázis, amely a felismerendő objektumok mintáit tartalmazza. A felismerés ezen adatbázis tartalmi elemeivel való összehasonlításon alapul.

E problémakör még ma is intenzíven kutatott területe a térinformatikai adatfeldolgozásnak. Idehaza [5], [6] és külföldön is egyaránt születtek figyelemre méltó megoldások [7].

Az általunk kidolgozott eljárás igen hasonló a neurális hálózati alapon működőkhöz. A szkennelt térképen található jelek felismeréséhez egy speciális referencia-adatbázist használunk, ahol megtalálható az adott térképen fellelhető összes pontszerű térképi jel (piktogramok, szimbólumok, karakterek). Ennek alapjául a térképi jelmagyarázat szolgálhat, ahol általában minden jel és megnevezése fellelhető. Az adatbázist minden digitalizált térképre egyedileg kell előállítani (kivéve, ha térképsorozatokról van szó, aminek a jelmagyarázata ugyanaz). Ezek az adatbázisba bevitt jelek lesznek a referenciajelek, egyedi azonosítóval (név vagy ID) ellátva. Az eljáráshoz írt felismerőprogram e referencia jelek alapján hozza létre a megfelelő helyen (képi X, Y koordináta) a vektorpontot, ami a felismert jel tulajdonságaival bír.

A felismerési adatbázis létrehozása

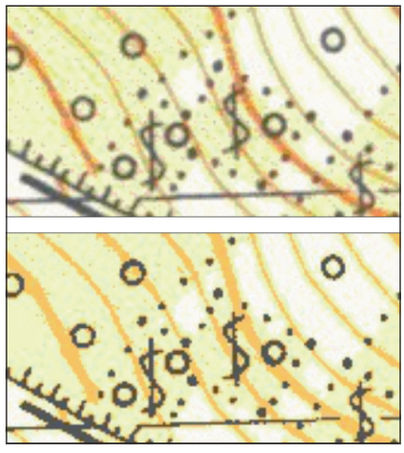
A felismerés hatékonysága arányosan növekszik a digitális térkép felbontásának növelésével. A tanulás folyamatánál, illetve a felismerési eljárásnál a program 144×144 pixelméretű jelekkel dolgozik. Hatékony felismeréshez digitalizáláskor célszerű ennek a jelméretnek a megközelítése. 300 dpi-s képnél ez 12 mm-es jelméretet jelent.

Első lépésként a referenciajeleket le kell választani az egyéb térképi tartalomtól. A folyamat során a pixelek tulajdonságait használjuk fel. A térképen azonos színnel nyomtatott elemek pixelei a szkennelt képen nem pontosan azonos színűek, hanem egy bizonyos színtartományba esnek. Az egyes nyomdai színekhez (és a fehér papírhoz) tartozó színtartományt egy-egy konkrét értékkel helyettesítjük, mintegy átszínezzük a térképet, így például a 16 millió színből álló képet korlátozott számú színből álló képpé alakítjuk át (1. ábra). Ezt nevezzük színredukciónak. Az így kapott színcsoportok színét és méretét használjuk fel a kiemelésre [8].

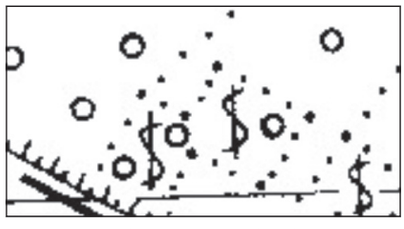
Következő lépésként binarizáljuk a képet, azaz fekete-fehérré alakítjuk (2. ábra). Binarizálás során az általunk kiválasztott színből fekete, a többiből fehér szín lesz, majd a megmaradt fekete színű elemek térbeli kiterjedését megvizsgáljuk két szempont szerint:

- a jel kiterjedése, azaz a jelet bennfoglaló téglalap szélessége és hosszúsága alapján, illetve
- a jelet alkotó pixelek száma szerint.

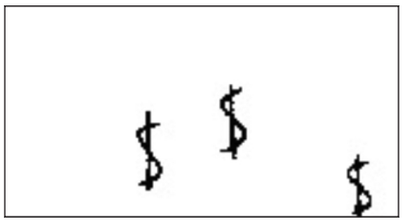
Így megfelelő mérethatárok beállításával kiszűrhetjük a felesleges elemeket, például kis pontokat, hosszú vonalakat (3. ábra).



1. ábra. A 16 millió színből felépülő képből generált 3 színű kép



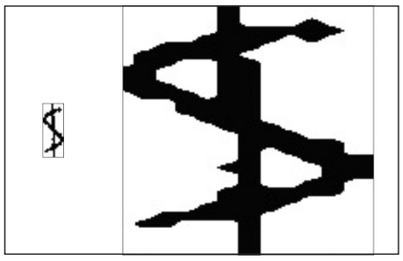
2. ábra. A binarizált kép



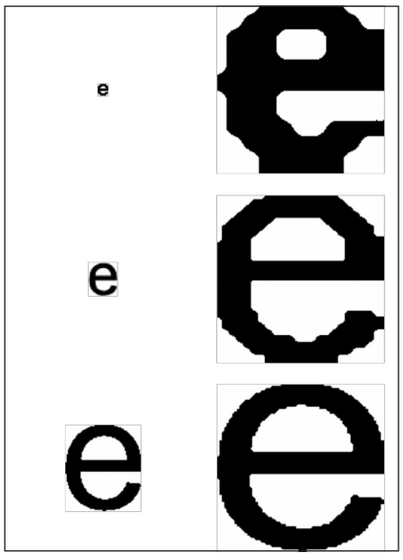
3. ábra. A szűrések után megmaradt jelek

Ezek után a szűréssel kiválasztott valamennyi térképi jelet kiemeljük a képről az összetartozó, egymással szorosan kapcsolódó pixelek mentén. A kiemelt jelet bennfoglaló téglalap bal felső sarokpontjának koordinátáit pedig elmentjük. Ezekre a koordinátákra az exportálásnál lesz szükségünk.

Az adatbázisba mentett jeleknek egységes méretűnek kell lenniük, hiszen minden térképi jelen ugyanaz a felismerő eljárás fut végig, ez pedig megköveteli az egységes méretet. Erre a 144×144 pixel a legalkalmasabb, mert elég nagy ahhoz, hogy az egyes jelek közötti különbségek érzékelhetőek legyenek, ugyanakkor nem olyan nagy, hogy feleslegesen megnövelné az adatbázis méretét. Ahhoz, hogy megkapjuk ezt a méretet, az egyes jelek képét át kell transzformálni, csökkenteni vagy növelni kell a szélességét, illetve a magasságát (4. ábra).



4. ábra. Az eredeti, 12x31 pixeles kép és a transzformálás után kapott 144x144 pixeles kép

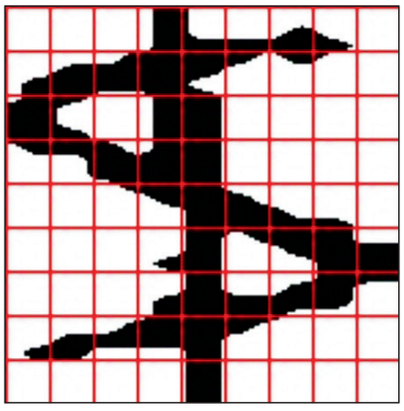


5. ábra. Különböző méretű jelek transzformálásakor fellépő torzulások

A transzformáció hatására azonban a kép torzul, a torzulás mértéke pedig annál nagyobb, minél jobban eltér a térképi jel mérete a 144×144 pixeltől. A torzulás mértékét az 5. ábra szemlélteti. Minél nagyobb ez a torzulás, annál jobban eltűnnek a jelek közötti különbségek.

A tanulás folyamata

A térképi jel megtanulásához, az adatbázisba történő beépítéséhez a program felosztja a képet 9×9 mezőre, ahogy azt az 6. ábrán a piros vonalak szemléltetik. 9 mező esetén páratlan számú mezőből álló rácsunk lesz, ami még jobban kiemeli az egyes jelek közötti különbségeket. Egy 144×144 pixeles kép esetén így egy-egy mezőben 16×16 = 256 pixel található.



6. ábra. A felosztott kép

0	0	0	0,629	0,406	0,250	0,410	0,246	0
0,215	0,621	0,719	0,980	0,746	0,254	0,219	0,086	0
0,938	0,492	0	0,625	0,934	0	0	0	0
0,262	0,449	0,813	0,879	0,938	0	0	0	0
0	0	0,070	0,422	0,945	0,672	0,301	0,148	0
0	0	0	0,145	0,879	0,227	0,586	0,953	0,629
0	0	0	0,258	0,922	0,449	0,555	0,789	0,270
0,137	0,488	0,422	0,715	0,949	0,352	0,039	0	0
0	0	0	0	0,813	0	0	0	0

1. táblázat

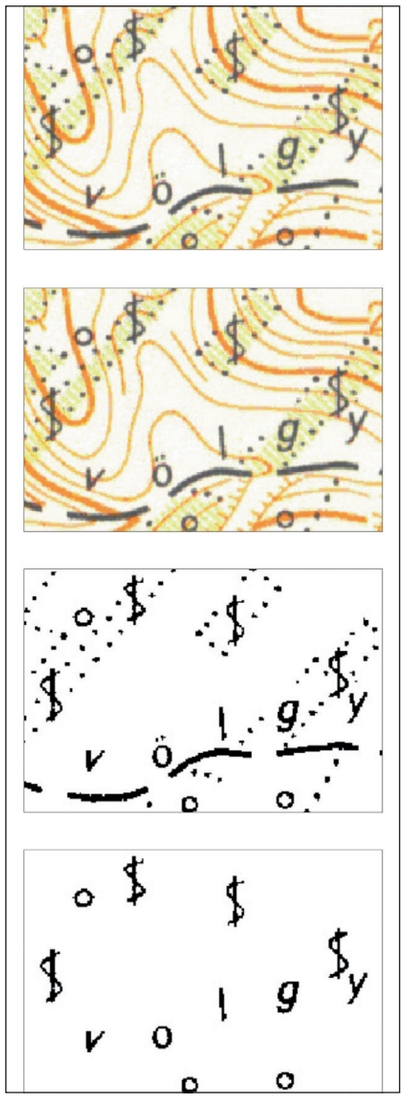
Az egyes mezőkön belül a program megszámolja a fekete pixeleket, és kiszámolja azok arányát a mezőben található összes pixel számához (256 db) képest. Így kapunk egy 0 és 1 közötti törtszámot, ahol 0 a teljesen fehér, 1 pedig a teljesen fekete mezőt jelenti. Az egyes mezőkön belüli feketepixel-arányokat az 1. sz. táblázatban láthatjuk.

A fekete pixelek arányainak értékeit egy adatbázisba mentjük. A szőlő jelét az 1. sz. táblázat reprezentálja. Elmentjük továbbá az arányszámok összegét is, amit nevezhetünk területnek is. Ha a jel teljesen kitölti a 144×144 pixel méretű négyzetet, azaz minden feketepixel-arányszám 1, akkor ezek összege 81 egység lesz. Ez a maximálisan elérhető területméret.

A felismerés folyamata

A jelfelismeréshez a digitalizált térképet elő kell készíteni. Ez az előkészítés a színek számának lecsökkentését jelenti, majd a referencia-adatbázisban tárolt jelek színeinek és méreteinek ismeretében a program megvizsgálja és kategorizálja a térképen található alakzatokat szín és méret szerint, és kiszűri azokat, melyek mérete nagyobb vagy kisebb, illetve a színük más, mint az adatbázisban tárolt jeleké (7. ábra).

Következő lépésként a szűréssel kiválasztott jelek képét egyenként át kell transzformálni 144×144 pixel méretű képpé. Az így kapott képeket külön-külön vizsgáljuk. Felbontjuk a képet 9×9 mezőre, és kiszámoljuk a fekete pixelek arányát az egyes mezőkön belül. Ezeket az értékeket összegezve megkapjuk az adott, felismerni kívánt jel területét is, ami az eljárás gyorsítására szolgál. Felismeréskor első lépésben ennek a területnek az összevetése történik a referencia-adatbázisban tárolt jelek területeivel, sorban egymás után. Ehhez meghatároztunk egy küszöbértéket, ami a felismerni kívánt jel és az adatbázisban tárolt jelek területeinek különbségén alapul. A küszöbérték megválasztásánál azonban figyelembe kellett vennünk, hogy a térképi karakterek és szimbólumok – bár ugyan az a betűtípus vagy piktogram szimbolizálja



7. ábra. Az eredeti térképrészlet, a 4 színre csökkentett, majd binarizált változat, végül a szűrés után megmaradt jelek

őket – a nyomtatás, majd szkennelés folyamán eltorzulhatnak. A 8. ábrán két, ugyanazon térképelemet (szőlő) ábrázoló jel látható, amelyek valamennyire mégis eltérnek egymástól. Ebből következően a területük is eltérő, felismeréskor azonban mindkettőt szőlő jelként kell felismernie a programnak. Tapasztalati úton ez a különbség maximum 10%-os lehet.

Ezt figyelembe véve, összevetéskor két eset lehetséges:

- Ha a két jel területének különbsége meghaladja ezt a küszöbértéket, akkor valószínűsíthető, hogy a két jel eltér egymástól. Ezzel szűkíthetjük azon jelek számát az adatbázisban, melyekkel összehasonlítjuk a felismerni kívánt jelet, csökkentve a felismerési eljárás futási idejét.
- Abban az esetben viszont, ha a felismerni kívánt jel és az adatbázisban soron következő jel területének különbsége nem haladja meg a küszöbértéket, akkor a program összehasonlítja ezt a két jelet. Kiszámolja mindkét jel feketepixel-arányszámát, majd ezek különbségét a 9×9 mező mindegyikében. Ezt követően ezeket a különbségeket összegezzük, így minden egyes összehasonlításnál kapunk egy eltérési értéket az adatbázisban található aktuális jeltől. Minél kisebb ez az érték, annál biztosabb, hogy a felismerni kívánt jel azonos az aktuálisan összehasonlított adatbázisjellel.

Ezt szemlélteti a 9. ábra. Az ábrán látható piros oszlopdiagram az adatbázisban tárolt, összevetés szempontjából éppen soron következő jel 9×9 = 81 darab feketepixel-arányszámát mutatja. Egy oszlop egy mező értékét mutatja, a 0 és 1 közötti értéktartományban. A teljesen fehér oszlop 0, míg a teljesen piros oszlop 1 értékű. A kék színű oszlopdiagram pedig a térképen felismerni kívánt, egyelőre ismeretlen jel 81 darab feketepixel-arányszámát mutatja. Az utolsó pedig az előző kettő egymásra vetítéséből áll elő.

A kettő tökéletesen megegyezik, fedi egymást, így biztosak lehetünk abban, hogy a felismerni kívánt jelünk 100%-ban megegyezik az adatbázisban tárolt aktuális jellel, ami a szőlőjel volt.

A példa kedvéért nézzünk egy másik szőlőjelet, hiszen nem minden jel egyforma a torzulások miatt (8. ábra). Ennek a jelnek a területe 17,615 egység. A 81 egységnyi területű négyzetből ekkora részt fed ki ez a jel. Ha ez a terület és a referencia-adatbázisban található szőlőjel területének különbsége nem éri el a küszöbértéket (10%-os eltérés), akkor a program ezt a jelet összeveti az adatbázisban levő szőlőjellel. A 10. ábra mutatja az oszlopdiagramok alakulását, hasonlóan az előző példához. A 9×9 mező különbségeinek összege 8,132 egység, ami 92%-os egyezést jelent az adatbázisban szereplő szőlőjellel.

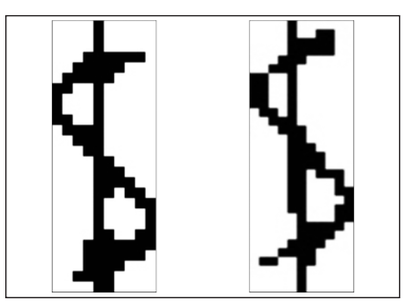
Mivel az adatbázisban található többi jel esetén ez az érték (az eltérés) csak nagyobb volt, így a program ismét sikeresen felismerte a szőlőjelet.

A 2. sz. táblázatban azt láthatjuk, hogy a további, adatbázisban tárolt jeltől mennyire tért el a felismert jelünk. Az táblázatban jól látszik, hogy a „Szőlőjel”-hez legközelebbi lehetőség az „S” betű volt, de a különbség itt is minimum 10 egységgel több volt, mint a „Szőlőjel” esetében.

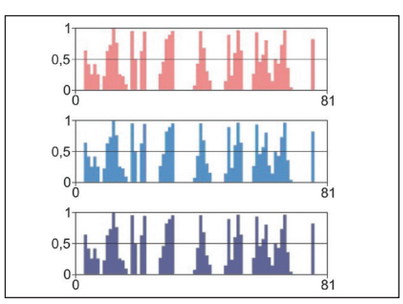
A listában szereplő számértékek a felismerni kívánt jel és az adatbázisban tárolt jelek különbségét mutatják, a számok mellett a jel megnevezése látható.

Az eljárás hatékonysága

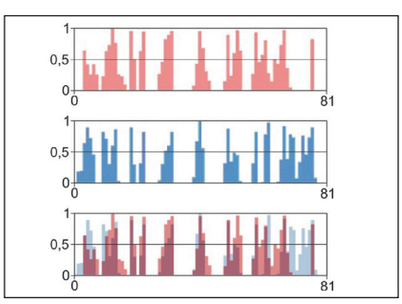
Az eljárás hatékonyságának mérésekor általában a futási sebességet és a sikeres felismerés arányát mérik. Az



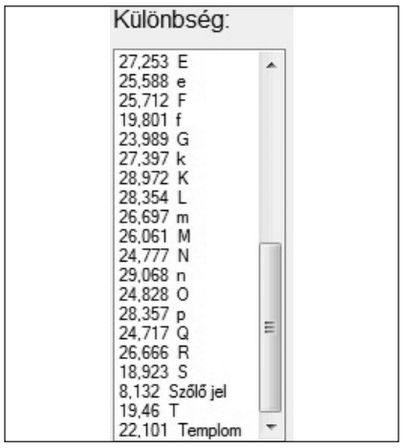
8. ábra. A térképi jelek a torzulások következtében nem egyformák (nagyítva)



9. ábra. A referencia adatbázisban tárolt jel és a felismerni kívánt jel képének összevetése egyező jelek esetén



10. ábra. A referencia adatbázisban tárolt jel és a felismerni kívánt jel képének összevetése kissé eltérő jelek esetén



2. táblázat

általunk készített eljárás hatékonyságát a 3. sz. táblázattal lehet a leginkább szemléltetni [9].

Az eredményekből látszik, hogy az eljárás 80 százalékot meghaladó hatékonysággal ismerte fel az egyes térképi jeleket. Az egyszerűbb alakzatokat könnyebben, míg a részletgazdagabb jeleket nehezebben azonosította, ami a jelek torzulására vezethető vissza (5. ábra). A kisebb térképi jelek transzformálásakor nagyobb arányú torzulás, deformálódás lép fel, aminek következtében az adatbázisban tárolt jel és a felismerni kívánt jel közötti különbség megnő, ez pedig tévesztésre ad lehetőséget. A teszt egy alsó középkategóriás számítógépen lett lefuttatva, Intel Pentium Dual Core B980 2.4 GHz-es processzor és 4 GB RAM található a gépben, az eljárás sebessége ennek ismeretében gyorsnak mondható.

A tévesen felismert jelek száma elhanyagolható a sikeres találatok számához képest, de megadja a további fejlesztés lehetőségét.

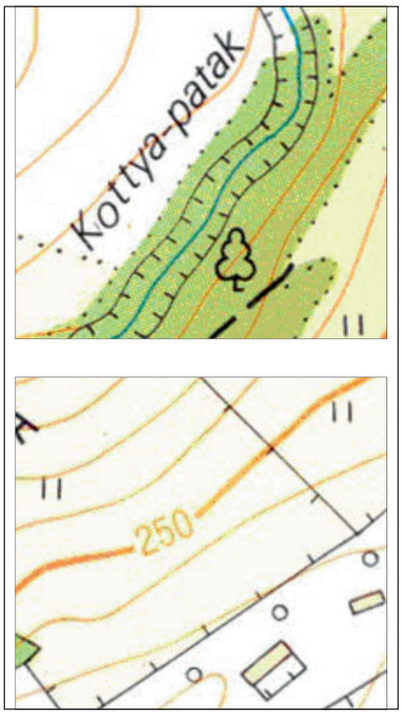
További fejlesztési lehetőségek

A térképen található pontszerű jelek egy része általában más elemekhez (vonalas vagy felületi elem) kapcsolódik, és hogy idomuljon hozzá, valamilyen szögben el van forgatva (11. ábrán látható megírások karakterei). Az ilyen jelek felismerésében nehézséget okoz, hogy az elforgatás szöge nincs megkötve, bármilyen értéket felvehet, sőt, betűnként változhat (például a 7. ábrán található völgy szó) [10] [11].

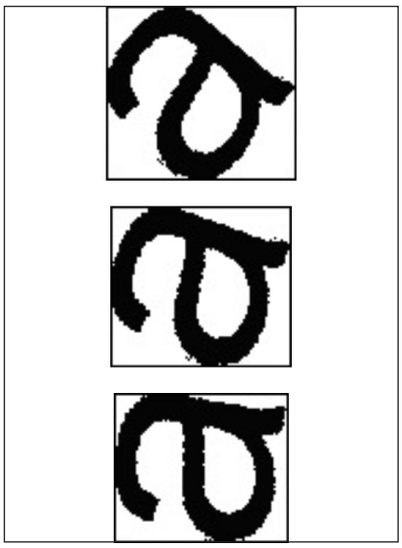
Felismeréshez az ideális eset az lenne, ha a jelet a „talpára” tudnánk állítani. De mivel még magát a jelet sem ismerjük, azt sem tudhatjuk, milyen forgatási szögnél áll a talpára. Ezért már az is elég, ha a felismerés előtt a jelet a 4 fő irány valamelyikébe forgatjuk. Általában a jelmagyarázatba foglalt jelek vagy vízszintesen, vagy pedig függőlegesen állnak, így a referencia-adatbázisba is így kerülnek be. Ha a térképen ettől eltérő szögben találjuk, akkor a jelet bennfoglaló legkisebb téglalap területe nagyobb

Kivágat mérete (pixel)	879x526	867x532	872x534	864x532
Színcsoportok száma	4	5	4	5
létrehozása (s)	4,12	5,27	4,17	5,10
Binarizálás (s)	3,53	3,53	3,59	3,54
Felismerés (s)	15,67	9,23	15,10	17,87
Jelek száma (db)	145	75	141	170
Felismert (db)	137	63	134	159
Arány (%)	94	84	95	93
	9/14	-	8/9	9/15
	3/5	12/13	1/1	2/5
	124/125	50/60	125/131	148/150
	1/1	-	-	-
	-	1/2	-	-
Tévesen felismert (db)	6	2	3	4
Nem felismert (db)	8	12	7	11

3. táblázat



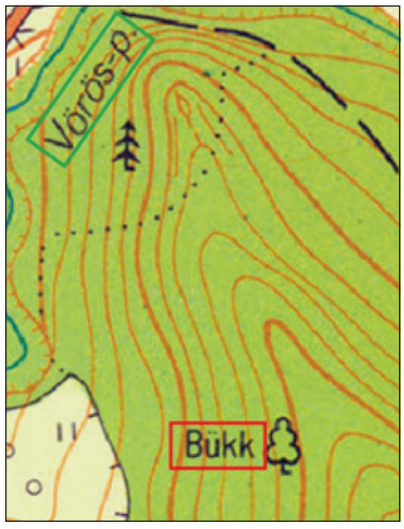
11. ábra. Elforgatott térképi megírások



12. ábra. Adott irányba forgatva a jelet, a bennfoglaló téglalap mérete csökken

lesz, mint vízszintes vagy függőleges állásnál. Ezt kihasználva addig forgatjuk a jelet valamelyik irányba, amíg az őt bennfoglaló téglalap mérete a legkisebb nem lesz (12. ábra).

A forgatás mértékétől függően a 4 fő irány valamelyikét közelítjük meg. Bár eljutunk a legkisebb bennfoglaló téglalaphoz, még mindig nem tudjuk, hogy a felismerni kívánt jelünk ezek közül pontosan melyik irányban áll. Ezért a jelet most elforgatjuk 90°-kal, majd alávetjük a felismerési eljárásnak. Ezt a folyamatot négyszer megismételve biztos, hogy egy állóképes felismerés is lefutott, ami megadja a felismert jelet. A felismeréskor eltároljuk az elforgatás szögét. A hasonló elforgatási szöggel rendelkező karaktereket ezután csoportosítjuk, majd az egyes csoportokat külön vizsgáljuk. Ha az egyes karakterek egy bizonyos távolságon belül vannak egymáshoz képest, akkor feltételezhető, hogy azok a karakterek összetartoznak, egy szót alkotnak. Például a 13. ábrán található „Vörös-p." és a „Bükk" megírás.



13. ábra. Az azonos szögben elforgatott, egymáshoz közel álló karaktereknél kapcsolatot feltételezhetünk

Az eredmények felhasználása

Az eljárásunk célja a digitalizált és felismert jelek (karakterek, szimbólumok) felhasználása más, vektorgrafikus rajzprogramokban, akár térinformatikai programokban is. Az eljáráshoz készített program MS Visual Studioban készült, működéséhez Microsoft Windows operációs rendszer szükséges. Azonban ahhoz, hogy ezek a felismert térképi jelek, karakterek más környezetben is felhasználhatóak legyenek, mindenképpen olyan exportálási formátum megválasztása szükséges, amely platformfüggetlen és amelyben a vektoros objektumokhoz attribútumok (tulajdonságok) is köthetőek. Erre a célra kiválóan alkalmas az ESRI shape-fájl, illetve a Scalable Vector Graphics (SVG) formátum.

A felismert jel képi pozíciójának és a jelet szorosan bennfoglaló téglalap méretének ismeretében a jel közepére teszünk egy vektorpontot. Ez a vektorpont már attribútumokkal ruházható fel. Ezen attribútumok lehetnek színre vonatkozóak, tárolhatják az elforgatottság szögét, tartalmazhatnak méretet, vagy éppen azt, hogy az adott karakter (szöveg) milyen típusú objektumhoz tartozik. Például egy kékkel megírt szöveg vízrajzi név lesz. Ezeket a pontokat külön-külön csoportokba téve (szimbólumoknál például áteresz-jelek, alappontok, épületek külön-külön csoportban) az exportált jeleket más programban gyorsan és könnyen lehet egységes grafikai megjelenítéssel felruházni.

Irodalomjegyzék

[1] K. Koo, J. P. Yun, S. Choi, J. Choi, D. C. Choi, S. W. Kim: Character segmentation and recognition algorithm of text region in steel images, 2009 (http://www.wseas.us/e-library/conferences/2009/cambridge/ISPRA/ISPRA45.pdf)

[2] A. Antonacopoulos, B. Gatos, D. Karatzas: ICDAR 2003 Page segmentation competition, 2003 (http://www.primaresearch.org/www/assets/papers/ICDAR2003_Antonacopoulos_Competition.pdf)

[3] S. Araokar - Visual character recognition using artificial neural networks, 2005 (https://arxiv.org/ftp/cs/papers/0505/0505016.pdf)

[4] Y. Chiang, P. Chioh, S. Moghaddam: A Training-by-Example Approach for Symbol Spotting from Raster Maps, 2014 (http://www.yoyoi.info/papers/chiang14giscience.pdf)

[5] Elek I., Dezső B., Máriás Zs.: IRIS, Automatikus raszter-vektor konverziós rendszer fejlesztése, IKKK 5. kutatási főirány, Beszámoló jelentés, ELTE Informatikai Kar, 2007 (http://lazarus.elte.hu/~elek/iris.pdf)

[6] Katona E.: Automatikus térképinterpretáció, Szegedi Tudományegyetem, 2000. (http://www.inf.u-szeged.hu/~katona/dissert.pdf)

[7] Boatto L., Consorti V., Buono M., Zenzo S., Eramo V., Esposito A., Melcarne F., Meucci M., Morelli A., Mosciatti M., Scarci S., Tucci M. (1992): An Interpretation System for Land Register Maps. Computer, Vol. 25, No. 7, pp. 25–33.

[8] Y. Chiang, C. A. Knoblock: Recognizing text in raster maps, 2014 (http://usc-isi-i2.github.io/papers/chiang14-geoinformatica.pdf)

[9] R. Szendrei, I. Elek, I. Fekete: Automatic Recognition of Topographic Map Symbols Based on Their Textures, 2011 (http://people.inf.elte.hu/fekete/tamop_2010/Cikkek/Szendrei_ICSI_2011.pdf)

[10] Y. Chiang, C. A. Knoblock: An Approach for Recognizing Text Labels in Raster Maps, 2010 (http://www.isi.edu/integration/papers/chiang10-icpr.pdf)

[11] U. Pal, S. Sinha, B. B. Chaudhuri: Multi-oriented text lines detection and their skew estimation, 2002 (https://www.ee.iitb.ac.in/~icvgip/PAPERS/242.pdf)

Summary

Recognition and Vectorization of Symbols and Characters on Maps

Automatic map recognition and vectorization is a dynamically developing area of contemporary cartography. A vectorized map could store significantly more information than a scanned one, if a database connection is also included. As an example, we can store not only the geometry of a given road, but also its width, number or pavement type and metadatas in the database, all attributes separately, so they can be searchable and filterable. In this work, we present a new process for recognizing and vectorizing map symbols and characters on a scanned map, even when they are rotated. Then the program – we made based on this process – can export the result into various file formats for further work in other softwares.

Nemes Krisztián
doktorandusz

ELTE TTK Földtudományi Doktori Iskola
e-mail: nekpaat@gmail.com