

Kiss Péter József, Samu Zsolt

TDK DOLGOZAT

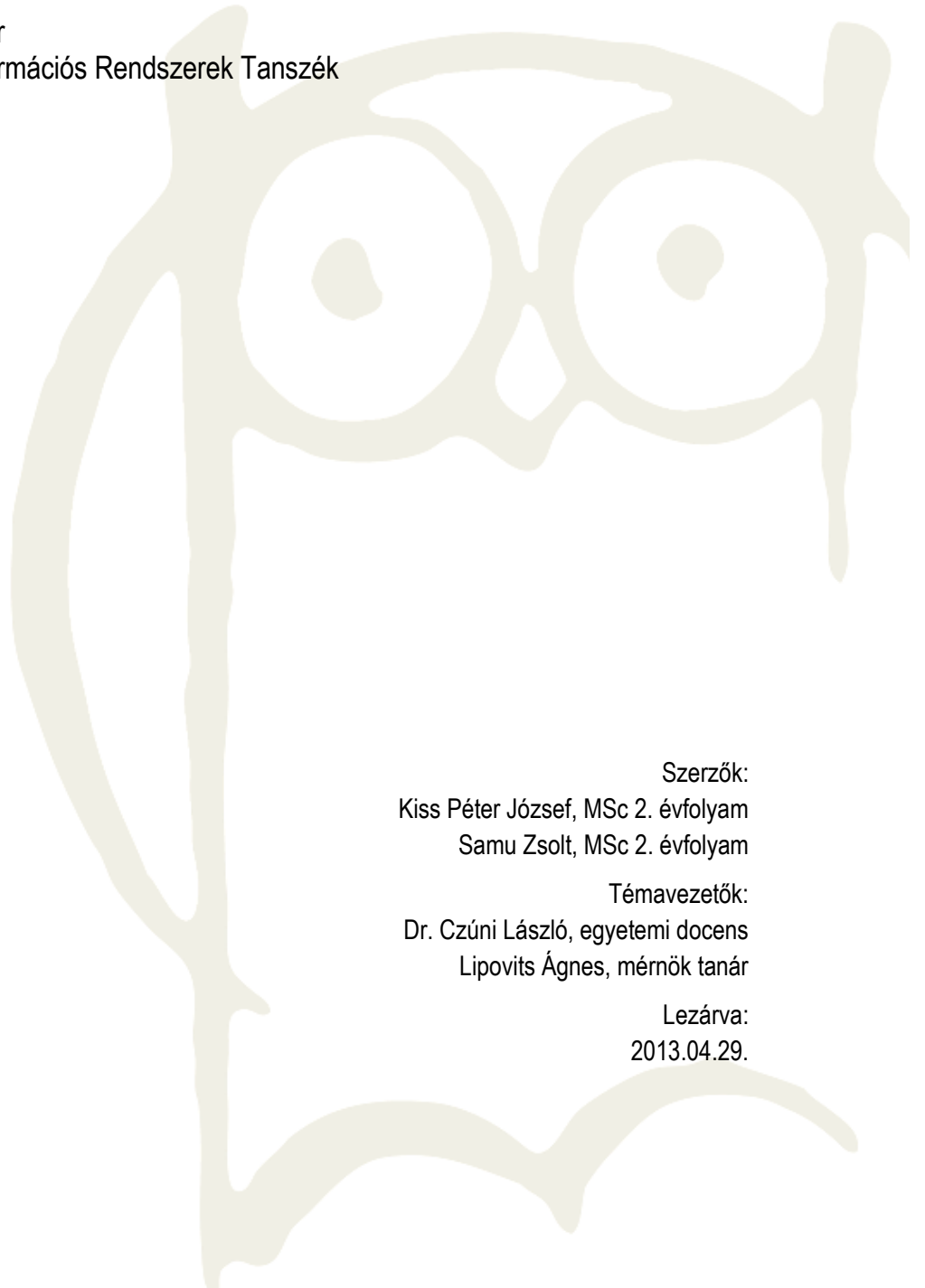


Kiss Péter József, Samu Zsolt

SZÓKÉP ALAPÚ KÉZÍRÁS FELISMERŐ RENDSZER ARCHÍV LEVÉLTÁRI DOKUMENTUMOKHOZ

Témavezetők: Dr. Czúni László, Lipovits Ágnes

Pannon Egyetem
Műszaki Informatikai Kar
Villamosmérnöki és Információs Rendszerek Tanszék



Szerzők:

Kiss Péter József, MSc 2. évfolyam

Samu Zsolt, MSc 2. évfolyam

Témavezetők:

Dr. Czúni László, egyetemi docens

Lipovits Ágnes, mérnök tanár

Lezárva:

2013.04.29.

Köszönetnyilvánítás

Köszönetünket szeretnénk kifejezni témavezetőinknek, Dr. Czúni Lászlónak és Lipovits Ágnesnek, valamint Gál Mónikának és a Veszprém Megyei Levéltár munkatásainak a segítségért és a hasznos tanácsokért! A kutatási tevékenységet és az eredmények publikálását a Magyar Kutatási Alap támogatta (OTKA CNK 80368).

Tartalomjegyzék

1. Bevezetés	6
2. Fogalmi áttekintés, elméleti háttér	7
2.1 Elektronikus dokumentum-feldolgozás	7
2.1.1 Karakter és szó alapú felismerés	7
2.1.2 Online és offline felismerés	8
2.2 Archív dokumentumok sajátosságai	8
2.3 Kép alapú összehasonlítás	11
2.3.1 Közvetlen képtartalom alapú megközelítés	11
2.3.2 Globális és lokális képleírók	12
2.4 Irodalmi áttekintés	14
2.5 Következtetések	16
3. Szókép alapú felismerő algoritmus	17
3.1 Az algoritmus főbb lépései	17
3.2 Kísérletek és eredmények	19
3.2.1 Tesztadatbázis	19
3.2.2 Lokális leírók összehasonlítása	20
3.2.3 Skála és elforgatás invariancia vizsgálata	21
3.2.4 További kísérletek	23
3.2.5 Eredmények értékelése	24
4. Archív kézírást dokumentum feldolgozó szoftver	26
4.1 Igények felmérése	26
4.2 Használati esetek	27
4.3 A folyamatokhoz biztosított szoftvereszközök	29
4.3.1 Szegmentálás	29
4.3.2 Eredmények értékelése	30
4.4 Technikai megvalósítás	31
4.4.1 Handwritten Document Processor	31
4.4.2 SIFT pont kivonó szoftver	31
4.4.3 Keresőmotor	32
4.4.4 Tudásbázis	32
4.5 A dokumentum feldolgozó szoftver kimenete és értékelése	33

1. Bevezetés

A levéltári gyakorlatban sokszor merül fel az archív dokumentumokban való keresés, illetve azok statisztikai elemzésének szükségessége. Jelenleg a döntő többségében kézírásos történelmi dokumentumok digitális feldolgozására informatikai támogatás csak minimális mértékben áll rendelkezésre. A dokumentumok feldolgozását alapvetően kézi munkával, hosszú idő alatt tudják elvégezni. Csak Magyarországon több millió digitalizálásra különösen érdemes dokumentum található, melyek elektronikus formában értékes információforrássul szolgálhatnak a levéltári kutatók, illetve a történész szakma számára.

A kézírásos anyagok feldolgozása az egyik legkevésbé megoldott terület a dokumentumok elektronikus feldolgozásában. Készülnek ugyan kézírás felismerő szoftverek, azonban hatékony offline kézírás felismerő algoritmusok egy-egy speciális területhez kapcsolhatók (pl. kínai karakterek felismerése, számjegyek felismerése). Az archív kézírásos dokumentumokra jellemző sajátosságok - a gazdag díszítés, a betűképek változása, az átlapolódó szavak, az elszíneződött foltos, sérült papír - szintén megnehezítik a felismerés feladatát.

A Veszprém Megyei Levéltár munkatársaival együttműködve, dokumentum feldolgozási munkájuk szoftveres támogatására kezdtük meg kutatásunkat a szókép alapú kézírás felismerés területén. A Pannon Egyetem Képfeldolgozás Kutatólaboratóriumának munkatársai korábban biztató eredményeket értek el SIFT alapú összehasonlító algoritmus alkalmazásával. A korábbi munkát továbbfejlesztve új összehasonlító algoritmust dolgoztunk ki a felismerési hatékonyság javítására, valamint vizsgáltuk a leíró különböző tulajdonságainak fontosságát. A felismerés biztató hatékonyságára alapozva az algoritmust a teljes dokumentum feldolgozási folyamatot támogató, könnyen kezelhető szoftver környezetbe foglaltuk.

A következő fejezetekben áttekintést adunk az archív dokumentum feldolgozás lehetőségeiről, menetéről. Bemutatjuk az általunk kidolgozott szókép alapú szöveg felismerési algoritmust és a hozzá kapcsolódó szoftver környezetet, mely kézírásos dokumentumok feldolgozását segítheti a levéltári munkában.

2. Fogalmi áttekintés, elméleti háttér

2.1. Elektronikus dokumentum-feldolgozás

Az elektronikus dokumentum-feldolgozás jellemző területe a papír alapú dokumentumok utólagos digitalizálása és feldolgozása (számítógépes szövegfelismerés). A feldolgozási feladatok bemenetei legtöbb esetben fotózott, vagy szkennelt oldalképek (teljes oldalak egy képen), az elvárt kimenet az oldalakon található szöveg, többé-kevésbé az eredetivel megegyező elrendezésben. A dokumentumon található írásos (esetünkben kézírásos) szöveget tehát első lépésben lokalizálni, alapegységekre bontani, azaz szegmentálni kell, majd a tartalmat (szöveget) kép alapján kell felismerni. A szöveg felismerési feladatok során alapvető kérdést jelent a felismerés legkisebb egységének meghatározása. Dokumentumok digitális feldolgozására két megközelítést találhatunk az irodalomban: az egyik a betűkép, vagy graféma alapú felismerés, mely során a szavak részleteit azok formája alapján ismerjük fel (pl. [14, 16]). A másik típus a teljes szó képét hasonlítja össze korábban azonosított, annotált példányokkal [2, 20]. Utóbbi módszert nevezzük szókép alapú felismerésnek, vagy „*word spotting*”-nak.

2.1.1. Karakter és szó alapú felismerés

Karakter alapú felismerés OCR (*Optical Character Recognition*) esetén a karaktereket azonosítani, szegmentálni kell. Ez gépi szöveg esetén, ahol a karakterek közel állandó (betűtípustól függő) formát vesznek fel, illetve aránylag szépen elválnak egymástól, alkalmazható megoldást jelent. Karakter alapú felismerés könnyen alkalmazható még úgynevezett rublika alapú adatbevitel esetén is, mellyel a közigazgatási ügyintézés esetén naponta találkozhatunk. A felismerést ekkor a korlátozott betűkészlet alkalmazása (nagy nyomtatott karakterek megkövetelése kitöltésnél), illetve pozicionálási segédjelek segítik. A szegmentálás ilyenkor az előre meghatározott rublikák alapján könnyen elvégezhető. Karakter alapú dokumentum feldolgozásánál alkalmazható támpontot jelent – és tovább növelheti az alkalmazás hatékonyságát – az azonosítani kívánt szöveg nyelvének ismerete. A nyelv sajátosságaira általában HMM (*Hidden Markov Modell*) [3] alapú leírásokat készítenek, melyek a karakterek egymás utáni előfordulásait valószínűsítik. A nyelvre vonatkozó sajátosságok felhasználásával egy karaktert nem csak a formája, hanem az előtte és utána álló karakterek is meghatároznak. Karakter alapú felismerés is kiegészíthető szótárral, mely esetben csak a szótárban található szavak fogadhatóak el a felismerés kimeneteként.

Amennyiben a karakterek nem szegmentálhatók a szövegben, karakter alapú felismerés nehezen alkalmazható. Folyóírásos szövegekben például a karakterek nem különülnek el egymástól, átlapolódnak, néha egymás alá, illetve fölé lógnak. A betűk írásképe is széles tartományban mozog, az ember írásképe egyénfüggő, sőt életkortól, hangulattól függően időben rövid és hosszú távon is változik. A karakterek, illetve a karakterkészletek tehát széles skálán változhatnak, azonosításuk nehéz, a feldolgozás során legtöbbször nem állnak rendelkezésre.

Speciális csoportot képeznek a dokumentumok között a hivatalos iratok, összeírások, mely írott anyagok viszonylag szűk szókészlettel rendelkeznek. A szűk szókészletű dokumentumok feldolgozása esetében szóképp alapú felismerés alkalmazható. Ez esetben a teljes szó képét hasonlítjuk korábban tanított példányokhoz. Szó alapú felismerés esetén is alkalmazhatjuk a nyelvi sajátosságokat (pl. HMM-et ezúttal a szavak egymás utáni előfordulására), de a nyelvi modell erőssége ez esetben gyengébb lesz, mint karakter alapú felismerés esetén. A szavak képeinek összehasonlításánál az irodalomban széles körben alkalmazott módszer az úgynevezett *Dinamic Time Warping (DTW)*. A DTW egy összehasonlító módszertan, mely például időben vagy sebességben változó mintafolyamok hasonlóságának mérésére alkalmazható. A DTW előnyének tekinthető írás felismerés esetén, hogy bizonyos mértékű nem lineáris torzulást megenged az írásképp hosszában, így a módszer hibatűrőbb lesz például egy díszesebb kezdőbetű, vagy az oldal szélén, hely hiányában összenyomott szóképp esetében.

2.1.2. Online és offline felismerés

Az eddig említett dokumentum feldolgozási módszertanok úgynevezett offline, vagyis utólagos megközelítést jelentenek. Offline szövegfelismerés vagy feldolgozás esetében a dokumentum (írás) keletkezésének körülményeiről nincsenek információink, a felismeréshez csak a kész szöveg képére (esetlegesen a várt szöveg nyelvére, szókészletére) támaszkodhatunk. Említést kell tennünk emellett a szövegfelismerés egy másik típusáról is, melyet általában real time alkalmazásokban, például szövegbevitel esetén alkalmaznak. Online felismerés esetén az adott írásról a keletkezés (általában időbeli, dinamikai) adatai is rendelkezésre állnak, vagyis tudhatjuk melyik vonal, milyen sorrendben, milyen sebességgel keletkezett. Online felismerés esetén tehát több információ áll rendelkezésre, hiszen a karakterek formálását végig nyomon követhetjük. Online felismerést alkalmaznak a kézírást felismerő érintőképernyők, vagy például a banki aláírás ellenőrző rendszerek esetében is. Dolgozatunkban archív történelmi dokumentumok feldolgozásával foglalkozunk, melyekről online adatok nem állnak rendelkezésre – a továbbiakban kizárólag offline felismerési módszerekkel foglalkozunk.

2.2. Archív dokumentumok sajátosságai

A levéltári munkában mindennapos gyakorlatnak tekinthető az archív szövegekben való keresés, illetve az archív dokumentumok digitalizálásának igénye. Csak Magyarországon több mint 3,5 milliárd archív dokumentum található, melyből kb. 200 millió elektronikus feldolgozása különösen fontos lenne a levéltári munka számára [7]. Jelenleg

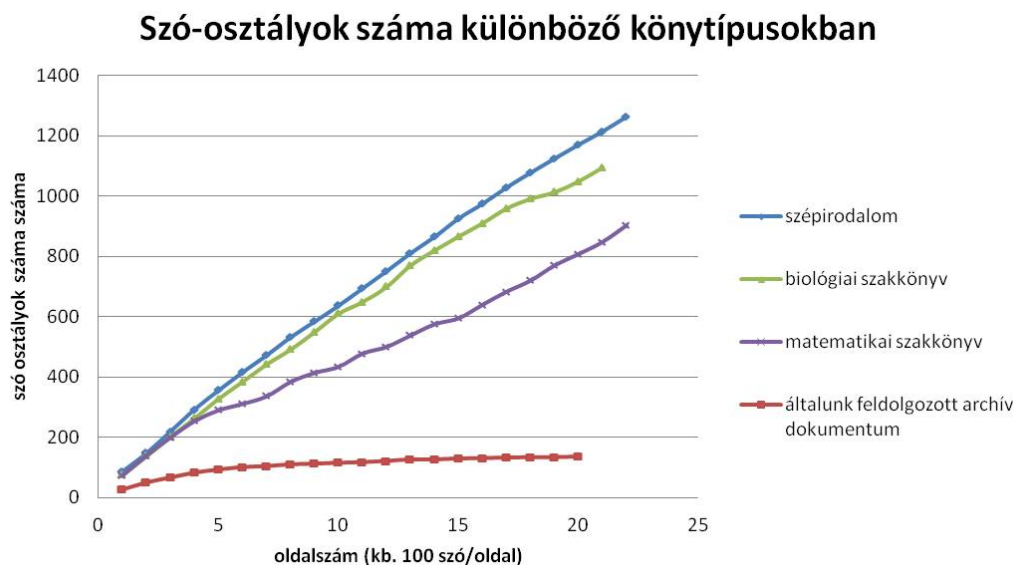
az archív dokumentumok feldolgozását emberi munkával végzik, minimális számítástechnikai támogatás mellett, ezek gyakorlatilag az MS Excel által nyújtott szolgáltatások.

The image shows a handwritten document, likely a church register or family record, dated 1771. The document is written in a cursive script and is organized into a grid-like structure with columns and rows. The columns are headed by names: 'Plomma', 'Friedrich', 'Friedrich', 'Friedrich', 'Friedrich', 'Friedrich', 'Friedrich', and 'Friedrich'. The rows contain various entries, including names, dates, and possibly ages or relationships. For example, the first row contains 'Plomma', 'Friedrich', 'Friedrich', 'Friedrich', 'Friedrich', 'Friedrich', 'Friedrich', and 'Friedrich'. The second row contains 'Friedrich', 'Friedrich', 'Friedrich', 'Friedrich', 'Friedrich', 'Friedrich', 'Friedrich', and 'Friedrich'. The third row contains 'Friedrich', 'Friedrich', 'Friedrich', 'Friedrich', 'Friedrich', 'Friedrich', 'Friedrich', and 'Friedrich'. The fourth row contains 'Friedrich', 'Friedrich', 'Friedrich', 'Friedrich', 'Friedrich', 'Friedrich', 'Friedrich', and 'Friedrich'. The fifth row contains 'Friedrich', 'Friedrich', 'Friedrich', 'Friedrich', 'Friedrich', 'Friedrich', 'Friedrich', and 'Friedrich'. The sixth row contains 'Friedrich', 'Friedrich', 'Friedrich', 'Friedrich', 'Friedrich', 'Friedrich', 'Friedrich', and 'Friedrich'. The seventh row contains 'Friedrich', 'Friedrich', 'Friedrich', 'Friedrich', 'Friedrich', 'Friedrich', 'Friedrich', and 'Friedrich'. The eighth row contains 'Friedrich', 'Friedrich', 'Friedrich', 'Friedrich', 'Friedrich', 'Friedrich', 'Friedrich', and 'Friedrich'. The ninth row contains 'Friedrich', 'Friedrich', 'Friedrich', 'Friedrich', 'Friedrich', 'Friedrich', 'Friedrich', and 'Friedrich'. The tenth row contains 'Friedrich', 'Friedrich', 'Friedrich', 'Friedrich', 'Friedrich', 'Friedrich', 'Friedrich', and 'Friedrich'. The eleventh row contains 'Friedrich', 'Friedrich', 'Friedrich', 'Friedrich', 'Friedrich', 'Friedrich', 'Friedrich', and 'Friedrich'. The twelfth row contains 'Friedrich', 'Friedrich', 'Friedrich', 'Friedrich', 'Friedrich', 'Friedrich', 'Friedrich', and 'Friedrich'. The thirteenth row contains 'Friedrich', 'Friedrich', 'Friedrich', 'Friedrich', 'Friedrich', 'Friedrich', 'Friedrich', and 'Friedrich'. The fourteenth row contains 'Friedrich', 'Friedrich', 'Friedrich', 'Friedrich', 'Friedrich', 'Friedrich', 'Friedrich', and 'Friedrich'. The fifteenth row contains 'Friedrich', 'Friedrich', 'Friedrich', 'Friedrich', 'Friedrich', 'Friedrich', 'Friedrich', and 'Friedrich'. The sixteenth row contains 'Friedrich', 'Friedrich', 'Friedrich', 'Friedrich', 'Friedrich', 'Friedrich', 'Friedrich', and 'Friedrich'. The seventeenth row contains 'Friedrich', 'Friedrich', 'Friedrich', 'Friedrich', 'Friedrich', 'Friedrich', 'Friedrich', and 'Friedrich'. The eighteenth row contains 'Friedrich', 'Friedrich', 'Friedrich', 'Friedrich', 'Friedrich', 'Friedrich', 'Friedrich', and 'Friedrich'. The nineteenth row contains 'Friedrich', 'Friedrich', 'Friedrich', 'Friedrich', 'Friedrich', 'Friedrich', 'Friedrich', and 'Friedrich'. The twentieth row contains 'Friedrich', 'Friedrich', 'Friedrich', 'Friedrich', 'Friedrich', 'Friedrich', 'Friedrich', and 'Friedrich'. The bottom of the page shows a summary row with numbers: 24, 10, 19, 4, 8, 3, 2.

2.1. ábra. Egy 1771-ből származó kézírásos dokumentum tipikus oldala

A levéltári dokumentumok nagy része akár több száz éves kézírásos szövegeket tartalmaz sárgult, foltos papírokon, melyek közül több korlátozott szókészlettel rendelkezik – ilyenek például az egészségügyi, jogi, anyakönyvi vagy kormányzati dokumentumok. Néhány dokumentum csoport például az anyakönyvek, illetve összeírások adatai – bár meglehetősen nagy szakmai információtartalmuk van, jelenleg digitalizálásuk meglehetősen sok időt és emberi energiát (munkaidő) vesz igénybe. A szókép alapú szó visszakereső algoritmus készítésekor kutatásunk alapjául egy közepes nagyságú magyar város (Veszprém) összeírását választottuk 1771-ből, mely pár ezer szó példánya mindössze-

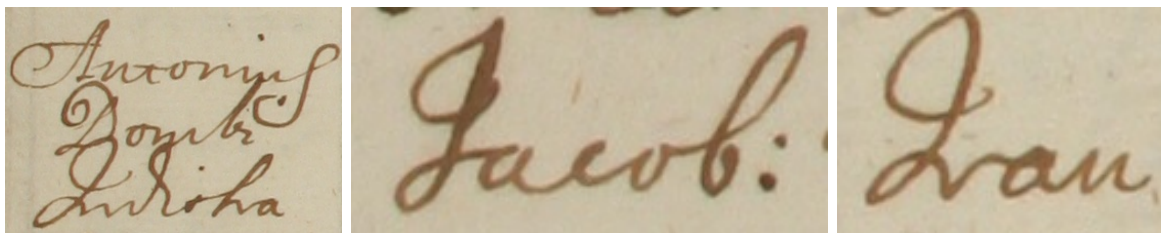
sen pár száz osztályból kerül ki. Az összeírás többnyire vezeték- illetve keresztnéveket tartalmaz, táblázatos formában, archív kézírással. A Veszprém Megyei Levéltár által rendelkezésünkre bocsájtott dokumentum egy oldalát a 2.1. ábra mutatja be.



2.2. ábra. A szó osztályok száma az oldalak függvényében különböző típusú dokumentumok esetén

A 2.2. ábrán a különböző szó osztályok számát ábrázoltuk különböző dokumentum típusok – egy magyar novella, egy biológiai, egy matematikai tárgyú szakkönyv és az általunk használt 1771-es összeírás – esetében, az oldalak függvényében. Az ábráról leolvasható, hogy a szavak száma az általunk vizsgált archív levéltári dokumentumok esetében nem nagymértékben növekszik, így a szókép alapú felismerés alkalmazása releváns lehet a hasonló típusú dokumentumok digitalizálása esetében. Az általunk vizsgált összeírás 175 oldalt tartalmaz oldalanként átlagosan 50-100 szóval. Ugyan a táblázatos forma segíti a szavak szegmentálását, a szóképek elkülönítését, a korabeli kézírás sajátosságai miatt azonban a betűk elkülönítése különösen nehéz feladatnak tekinthető. A korabeli dokumentumok írásképei számos egymásba lógó, átlapolódó betűt tartalmaznak – számos esetben találkozhatunk díszített írásképpel, vékony, vagy megszakított vonalakkal, rövidítések alkalmazásával. Sok esetben a szavak felismerése még „szakértő szemmel” is nehéz – a levéltári dolgozónak a hasonló alakú szóképek, vagy a szövegkörnyezet segíthetnek a kézi feldolgozásban. Néhány sokszor előforduló, tipikusnak nevezhető nehézséget a 2.3. ábrán szemléltetünk.

A szavak felismerését segíti a vizsgált dokumentum csoport esetében, hogy azokat néhány, de mindenképpen limitált számú szerző írta, viszonylag hasonló „hivatali” írásképet használva. Az archív levéltári dokumentumok egy csoportja tehát meglehetősen alkalmasnak bizonyul szókép alapú digitalizálásra, mivel a szó osztályok száma lassan nő az oldalak számának növekedésével, a betűk azonban nehezen azonosíthatóak.



2.3. ábra. A dokumentumokban előforduló tipikus felismerési nehézségek, átlapolódó szavak, hasonló „I” és „J” íráskép a Jacob, illetve Ivan szavakban

2.3. Kép alapú összehasonlítás

Szövegfelismerés esetén gyakorlati szempontot tekintve két kép hasonlóságát kell megállapítani. A tanító adatbázisban találhatóak a karakterek vagy szavak annotált (felcímkézett) képei. A tanítómintában (tudásbázis) található képeket *candidate*-nek (jelölt), a felismerni kívánt szó képét *query* (kérdés) képnek nevezzük. Célunk egy query kép leginkább hasonló candidate kép megkeresése, melyhez rendelt címke (annotáció) adja a keresett szó jelentését. A szövegfelismerés problémája tehát – a 2.1.1. pontban bemutatott nyelvi sajátosságokon alapuló modellektől eltekintve – két kép összehasonlításának problémájára vezethető vissza.

Képek összehasonlítására számos eljárás létezik, melyek különböző nézőpontból közelítik a hasonlóság fogalmát. Az eljárások két fő csoportba sorolhatók – két kép között értelmezhetünk fizikai, és tartalmi hasonlóságot. A fizikai összehasonlító algoritmusok a képminőség értékelésben alkalmazott eljárások, a két kép közötti fizikai (pixelenkénti) eltérés mértékét adják vissza. Az összehasonlító eljárások másik nagy csoportja a tartalmi összehasonlítás, mely esetben a kép szemantikai tartalmát, a képen található objektumokat próbáljuk összehasonlítani. Az összehasonlítás történhet globális jellemzők (színi jellemzők, hisztogramok) alapján, vagy lokális jellemzőkön alapulva, mikor a kép egyes részleteit vizsgáljuk.

2.3.1. Közvetlen képtartalom alapú megközelítés

Közvetlen képtartalom alapú összehasonlítás esetén két kép pixelenként, vagy kicsit nagyobb egységenként kerül összehasonlításra. Az egyik legelterjedtebb módszer két jel, kép összehasonlítására például a *Mean Square Error*, vagyis átlagos négyzetes hiba használata. A közvetlen képtartalom összehasonlításnak a képminőség mérés területén széles irodalma van, a fenti módszereknek számos továbbfejlesztése létezik, melyek invariánsak például az eltolásra, vagy nagyításra – tartalom alapú visszakeresésre való felhasználhatóságuk esetünkben (vékony vonalas objektumok) azonban korlátozott (legfeljebb 50% körüli eredményekkel[19]), így a módszerek használatát kézirás felismerésre elvetettük.

A közvetlen képtartalom alapú megközelítés területéhez sorolhatóak a mintaillesztés, vagy *template matching* módszerek is. Ezekben a módszerekben kis képrészleteket állapítunk meg, melyek jellemzőek egy-egy objektumra vagy azok egy részletére, és ezekkel

a kis mintákkal hasonlítjuk össze a kép egyes területeit. A mintaillesztési módszereket hatékonyan alkalmazhatjuk arc, jármű detekcióra, így akár képek indexelésére is – affine transzformációkra invariáns megvalósításaik azonban nagy számításigénnyel rendelkeznek. Kézírás felismerésnél korlátozottan alkalmazható módszerek (karakter alapú felismerésre bizonyos kiterjesztésekkel alkalmas, de nem túl hatékony módszer).

2.3.2. Globális és lokális képleírók

Képleírók alkalmazása esetén nem közvetlenül a kép pixel alapú tartalmát használjuk az összehasonlításakor, hanem a kép egyes jellemző tulajdonságai alapján határozzuk meg a hasonlóság mértékét. Jellemzően a képekből kivonunk jellemző tulajdonságokat (globális leíró esetén az egész képből, lokális leíró esetén a jellemző pontokból külön-külön), így minden képre kaphatunk a képre jellemző tulajdonság leíró vektort (globális), vagy leíró vektor halmazt (lokális). Az összehasonlításnál a képleíró vektorokra, illetve vektor halmazokra definiált távolság metrika szerint állapítjuk meg két kép hasonlóságának mértékét.

A globális leírók a képből egy leíró vektort állítanak elő, ezek lehetnek például egy kép rögzített számú jellemző színei, azok előfordulási területével, vagy textúra jellemzési adatok a kép egyes területeiről. Kézírásos szavak felismerésére több esetben [9, 14] használnak globális leírókat. A globális leírók legtöbb esetben a kép úgynevezett skalár jellemzőit tartalmazzák, melyek a kép magassága, szélessége, képaránya, fentről, vagy lentől a legelső írás (tinta) pixel távolsága és az írás egyes becsült jellemzői, mint például a szótörzsből lefele, illetve felfele kilógó karakterek számának becsült értéke.

Használhatóak kézírás felismerésére az úgynevezett vetítési profilok [15] is – ezeket a tulajdonságokat a kép oszlopaiként egy-egy számmal reprezentálhatjuk:

1. vetítési profil (*projection profile*) – a pixel intenzitások összege oszloponként
2. felső szóprofil (*upper word profile*) – a kép felső széle, és az oszlopban található legfelső tinta pixel egységre normált távolsága
3. alsó szóprofil (*lower word profile*) – a kép alsó széle, és az oszlopban található legalsó tinta pixel egységre normált távolsága

A vetítési profilokat függvényként értelmezve vizsgálhatjuk azok hasonlóságát. Megfelelő összehasonlítási metrika esetén stabilak az eltolásra, azonban az elforgatásra általában erősen érzékenyek.

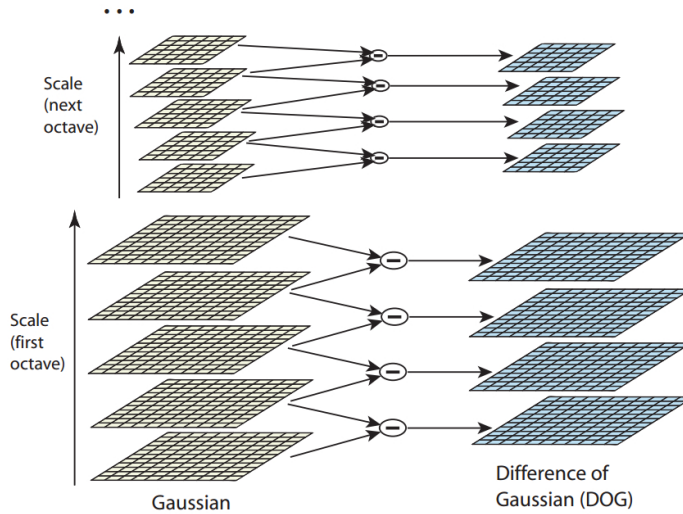
A lokális tulajdonság leírók alapvetően két folyamatból állnak össze – jellemző pontok megállapítása a képen, illetve a jellemző pontok tulajdonság leírása, azaz a pontok környezetének valamely tulajdonságok alapján való jellemzése, leíróvektor számítása. A jellemző pontok lehetnek akár élek, vagy sarkok (pl. Harris sarok detektor alkalmazása), vagy különböző textúra határok. A jellemző pontokat általában gradiens hisztogram alapú leírókkal reprezentálják.

Az egyik legstabilabb és legszélesebb körben alkalmazott lokális tulajdonság leíró algoritmust, a SITF-et (*Scale-invariant feature transform*) David Lowe publikálta 1999-ben [12] – azóta számos területen alkalmazták (pl. panoráma kép illesztés, objektum

felismerés, 3D modellezés), és több továbbfejlesztése (pl. color SIFT) is. Kézírás felismerő algoritmusunkhoz előnyös tulajdonságai és robusztussága miatt a SIFT leíró választottuk, a következő szakaszban ezért a leíró kiszámításához szükséges algoritmust részletesebben ismertetjük. Az algoritmus részletes leírását [13]-ben olvashatjuk.

SIFT leíró

1. A SIFT leíró kiszámításához első lépésben egy skála teret kell definiálnunk. A skála tér (*scale space* a kép $(1 + \frac{k}{n})\sigma$ paraméter szerinti Gauss elmosott változatait tartalmazza, ahol n bemeneti paraméter az egy oktávon elvárt skála szintek száma, $k \in \mathbb{Z}, 0 \leq k < 1$. A 2σ mértékben elmosott képet már információvesztés nélkül alul-mintavételezhetjük, méretét mind szélességében, mind magasságában felére csökkentve. Az így létrehozott következő oktávon ismét létrehozuk az elmosási szinteket. Így a kép egy Gauss piramissal reprezentált változatát kapjuk. A Gauss piramisból kiszámíthatjuk a Gauss különbségi képpiramist – a különbségi piramis, vagy *Difference of Gaussian (DoG)* egy szintje az azonos oktávon található, szomszédos skála szinteken reprezentált képek különbsége. A DoG piramis felépítését a 2.4 ábra szemlélteti.



2.4. ábra. A DoG piramis felépítése [13]

2. Az algoritmus a DoG térben lokális minimum és maximum pontokat keres. A lokális szélsőérték feltétele, hogy a pontnak a szomszédos 8 pixel értékénél, és a piramis szomszédos szintjein található 9-9 közvetlen szomszédjának értékénél is kisebbnek (minimum), illetve nagyobbak (maximum) kell lennie. A különböző skála szinteken detektált pontok koordinátáinak pontos meghatározásához interpoláció szükséges. Az így kapott kiugró pontok azonban még nem feltétlenül elég stabilak a megfelelő invariancia biztosításához, így a pontokat szűrési feltételeknek vetjük alá (pl. él miatti erős válaszok, alacsony kontrasztú jelöltek szűrése). A

szűrési feltételeknek megfelelő kiugró szélsőérték pontok lesznek a továbbiakban a SIFT kulcspontok.

3. Az elforgatás invariancia biztosításához relatív koordináta rendszert kell kijelölni. A relatív koordináta rendszert a kulcsponthoz tartozó fő gradiens irányhoz (a kulcspont orientáció) rögzítjük. Az orientáció kiszámításához a Gauss reprezentáció adott skálájának pixel értékeit használjuk. Előfordulhat, hogy egy kulcspont esetében több orientációs irány is jellemző lesz, ebben az esetben a második jellemző iránynak új kulcspontot hozunk létre, azonos skálán és pozícióban.
4. A kulcsponthoz tartozó leíró vektort, a 128 elemű gradiens hisztogramot a kulcspont orientációjához rögzített lokális koordináta rendszerben, a Gauss reprezentáció adott skáláján 16×16 -os pixel környezetéből számoljuk. A 16×16 -os területet 16 darab 4×4 -es alrégióra osztjuk. A gradiens értékeket alrégióként 8 irányban számoljuk ki. A megvilágítás változása okozta különbségek kiküszöbölésére a leíróvektort egységre normáljuk.

Egy SIFT pont tehát (x, y) (interpolált) koordinátákból, egy orientáció, egy skála értékből, illetve egy 128 elemű gradiens hisztogram leíró vektorból épül fel. A képen a SIFT-pontok száma nem korlátozott, a képtartalomtól, illetve a kép méretétől erősen függ. A SIFT invariáns a nagyításra, elforgatásra, eltolásra, illetve a megvilágítás változásaira, vagyis meglehetősen robusztus leírónak tekinthető, mely számos alkalmazási területe mellett alkalmas lehet kézírást tartalmazó képek leírására, és összehasonlítására.

2.4. Irodalmi áttekintés

A dokumentumfeldolgozás, ezen belül a kép alapú szövegfelismerés meglehetősen széles körű irodalommal rendelkezik a képfeldolgozás területén. Számos helyen, például a közigazgatásban, hivatali eljárásoknál, postai iratkezelésben napi szinten alkalmaznak karakter felismerő úgy nevezett OCR (Optical Character Recognition) szoftvereket, melyek nyomtatott karaktereket, vagy írott nyomtatott nagybetűket képesek nagy biztonsággal felismerni. Ezeknél a dokumentumoknál azonban a betűk meglehetősen jól szegmentáltak, legtöbbször előre meghatározott rublikákban találhatóak, és megjelenésük is szűk mintához igazodik. Dolgozatunk esetében a feldolgozandó dokumentumok korábban (2.2. pont) bemutatott sajátosságai miatt betű alapú felismerés, vagyis klasszikus OCR nem alkalmazható, így az irodalmi áttekintés esetében is a szókép alapú felismerés szűkebb területével foglalkozó legfontosabb cikkek bemutatására szorítkozunk.

Kézzel szegmentált szavak felismerését mutatja be [9]. A szavak képein a szegmentálás után több előfeldolgozási lépést végeztek, mint például a belógó részek eltávolítása, a szavak alapvonalainak azonosítása és pozicionálása, a ferdeség csökkentése, illetve a háttér normalizálása. A szavak azonosításához Rejtett Markov Modellt (HMM) használnak, ahol az állapot állapotátmeneti valószínűségeket a szavak bigram gyakorisága adja. Sokféle leírót használnak a modellépítéshez, a szöveg skaláris jellemzőitől (ereszkedő-, emelkedő szárok száma, szélesség, magasság stb.) az összetettebb jellemzőkig (vetületi

profil, alsó, felső szóprofil Diszkrét Fourier Transzformáció együtthatói). A módszerrel 65%-os találati eredményt sikerült elérni történelmi dokumentumoknál.

Archív dokumentumok szóképek alapú digitalizálására mutat módszert [14], mely során a szavak képeiből klasztereket alakítottak ki - két szóképek közötti hasonlóságot az ún. Dynamic Time Warping (DTW) módszerrel határozták meg. A sorok illetve a betűk ferdeségét korrigálták az előfeldolgozás során. A klaszterezés hasonlósági mértékét az alsó és felső profil összehasonlításából, és az előtér-háttér átmenetek számából kapták. Archív dokumentumok esetén a legjobb esetben 31.5%-os minimális hibaarányt értek el.

[17] szerzői két kép fontos lokális területeinek különbségére épülő összehasonlító módszert mutatnak be. A képeken Harris sarokdetektáló eljárás segítségével megkeresik a fontosnak ítélt területeket, majd ezeken a területeken kiszámítják a két kép differenciáját, SSD (Sum of Squared Differencies) módszerrel. Az eljárás nem a teljes képet használja fel közvetlen összehasonlításra, hanem annak bizonyos kulcspontjait, és a kulcspontok közvetlen környezetének képi tartalmát.

A kulcspontok környékén található képi tartalmat nem feltétlenül szükséges közvetlen formában felhasználni az összehasonlítás során, az azokhoz tartozó kis képterületekből lokális leírókat számíthatunk. A képeket ezután a hozzájuk tartozó kulcspontokkal, és azok leíróival jellemezhetjük. Lokális leírókat alkalmazó módszerre mutat példát [16], ahol grádiens hisztogramokat számítanak ki egy mozgó ablakban – az eljárás során nincs kulcspontkiválasztás, a kép minden pontjára kiszámítják a lokális leírókat, melyeket [9]-ben bemutatott módszerhez hasonlóan DTW és HMM segítségével hasonlítanak össze. Bár a hibázási ráta módszertől függően 12-30% körül alakul, azonban a módszert csak a teszt-dokumentumban legtöbbször előforduló 10 szó-osztályon tesztelték.

Speciálisan kínai karakterek felismerésére mutat be SIFT-hez hasonló leírókat alkalmazó metódust [10]. A cikk szerzői szerint az eredeti SIFT leíró rotáció és skála invarianciája gyengíti a leíró erejét karakterfelismerés esetén, így egy SIFT-hez hasonló grádiens-hisztogram alapú (skála, és elforgatás függő) leírókat számítanak ki néhány sajtósági pontban.

Gráf alapú karakter felismerő módszert alkalmaz történelmi dokumentumokon [4], mely a szavak binarizálása, a ferdeség és sormagasság normalizálása után a karakterek vázát gráfokkal reprezentálja (vonalvégek, elágazások, körkörös struktúrák). Részgráf alapú összehasonlító metodikát és HMM alapú szófelismerést alkalmazva a módszer jó eredményeket mutat a Parzival tesztadatbázison, mely 13. századi kézírásos szövegeket tartalmaz. (Fontos megjegyezni azonban, hogy ezek a szövegek úgy nevezett *fraktur* írásmóddal készültek, vagyis a betűk szépen szegmentáltak, nincsenek kötések. Az általunk feldolgozni kívánt dokumentumok folyó írással készültek – a betűk szegmentálása nehezebb, betű alapú felismerés nehezen lenne alkalmazható.)

A [18] módosított karakterleíró kód (Modified Character Shape Codes) [10] használatát javasolja szóalapú felismeréshez kézírásos angol nyelvű szövegeknél. A módosított algoritmus általánosabb tulajdonságok alapján definiálja a leíró kódot, így növeli a felismerés pontosságát, és csökkenti a feldolgozási időt azáltal, hogy az előfeldolgozási műveletek nagy része elhagyható.

A [11] komplex módszert mutat be szóalapú felismeréshez szóalapú modell (szintetizált szó) segítségével, megoldva a karakteralapú modell (graféma spektrum) tanítási

problémáit. Az alapmódszer karakteralapú tanítást használ, de mivel kézirásos szövegeknél az automatikus karakterszegmentálás nem megoldott, ezért grafémaalapú tanításra tér át. A grafémák egy vagy két karakter részeit tartalmazó alapegységek. A binarizált, vázósított képeken lévő szavakat grafémákra vágva, adott jellemzők (lokális minimumok és maximumok, elágazások) felhasználásával gépi tanulással megoldható a tanítómin-ták generálása. A módosított algoritmus segítségével növelhető a kézirásos történelmi dokumentumok feldolgozása.

Kétlépcsős módszert mutat be a szó alapú felismerésre a [8]. Módosított optikai áramlás módszerével elmozdulás kompenzációt végez, és kiszámolja minden pixelre az elmozdulás vektor értékét. A szimultán binarizáció mindkét képen (query és candidate) optimalizálja a különbségetektálást. A módszer 80% feletti felismerési pontosságot adott 18. századi fraktur goth germán dokumentumoknál.

SURF (Speeded-up Robust Features) leírót használ [20] kézirásos számjegyek felismerésére. Számjegyenként több kulcsponthoz azonosítanak 128 dimenziós vektor segítségével, melyeket 10 osztályba sorolnak. A SURF pontok pozícióját figyelmen kívül hagyják. A számjegyek felismerésénél a megtalált SURF pontok többségi szavazással döntenek a számjegy javasolt osztályáról. Több mint 90%-os felismerési rátát értek el a 10 numerikus karakter esetében.

2.5. Következtetések

Az irodalmi áttekintés után úgy találtuk, hogy minden lépésnek több különféle megközelítése lehet: előfeldolgozás (pl. különböző típusú normalizálás), jellemzők kivonása (pl. élhisztogram, alsó-felső profil, stb.), hasonlósági mérték (pl. euklideszi távolság, DTW), osztályozás (pl. k-NN, szavazásos módszerek, HMMs). Azt tapasztaltuk, hogy a legtöbb módszer előfeldolgozás segítségével megpróbálja elérni a skála és az elfogadás invarianciát, de tudjuk, hogy archív kézirásos dokumentumoknál ez nagyon bonyolult. Egyedül a [20] használtak skála és elforgatás invariáns leíró, elkerülve az előfeldolgozást. A szerzők megmutatták az is, hogy a legjobb eredményeket fix skála és orientációs beállítással érték el. Az alkalmazott SURF leírónál csupán a leíró pontok kiválasztása hozzájárulhatott a jó eredményhez. Fontos megjegyeznünk azonban, hogy a bemutatott teszt korlátozott képméretre (29x29), és mindössze a 10 számjegyre vonatkozott.

Szintén kiderült számunkra, hogy nagyon nehéz összehasonlítani a publikált módszerek teljesítményét több okból: adatbázisok különbözősége (pl. folyóírás, fraktur); különböző előfeltételek (pl. kézi tanítású karakterek vagy szavak); az osztályok számának eltérése (néhány esetben csak 10 osztályt különböztettek meg), és a különböző hibamértékek használata (találati arány, F-mérték, FRR, FAR, stb.). A lokális tulajdonság kivonó módszerek széles körben alkalmazhatóak a kézírás felismerés területén is.

A SIFT leíró használata az irodalmi adatok alapján jó alapot teremthet egy szó alapú kézírás felismerő rendszer tervezéséhez, melytől az irodalmi adatok alapján kb. 80%-os találati arány várható el.

3. Szókép alapú felismerő algoritmus

Mivel akár a sorok, szavak, akár a betűk erősen torzulhatnak forgatással, nagyítással vagy egyéb affine transzformációval, a szavak képeinek leírásához az ezekre nagy részben invariáns SIFT leíró használata megfelelőnek tűnt az általunk megvalósítandó feladatra. A SIFT leíró szókép alapú felismerés területén történő alkalmazására számunkra eddig ismert egyetlen korábbi munka szintén a kutatólaboratórium munkája [2]. Mivel a leíró alkalmazásával biztató eredményeket értünk archiv dokumentumok esetén, pontosítottuk, és gyorsítottuk algoritmusunkat, valamint kiterjesztettük kutatásunkat, hogy választ találjunk néhány fontos kérdésre:

- fontos a leíró skála invarianciája a felismerésben?
- fontos a leíró orientáció invarianciája a felismerésben?
- elég-e a szavak képi struktúrájának (pl. morfológiai vázának) ismerete a lokális leíró kinyerésénél a hatékony felismeréshez?

Mivel a szavak szegmentálása archiv kézírásos dokumentumok esetében önmagában szerteágazó és bonyolult problémaként kezelhető, a kutatásunk során ezzel a területtel nem foglalkoztunk, a szavak szegmentálását (a szoftverbe épített komponens esetén is) kézzel kell elvégezni. Az algoritmus bemeneteként szegmentált (query) szóképek, valamint egy tudásbázis (annotált szóképek – candidates) állnak rendelkezésre. Az algoritmus alkalmazásánál előnynek tekinthető, hogy a képeken a kivágástól eltekintve további elő-feldolgozásra nincs szükség.

3.1. Az algoritmus főbb lépései

1. A SIFT kulcspontok kiválasztása, valamint az azokhoz tartozó SIFT leíró vektorok generálása mind a query (Q), mind a candidate (C) szó-képek esetében. Kutatásunk során több SIFT implementációval kísérleteztünk – az eredeti Lowe féle implementáció mellett [12], használtuk az Open SIFT könyvtárat [5], illetve annak általunk módosított (skála fixált, orientáció fixált) változatait.

Az összehasonlítások alapjául szolgáló leírók esetünkben kulcspontként 132 dimenziós vektorok, melyek tartalmazzák a 128 dimenziós SIFT leíró 8 bites egész értékeit (h_i), a kulcspont fixikai pozícióját a képen (x, y), az orientáció (α) és skála értékeket (s), $q = [h_1, h_2, \dots, h_{128}, x, y, \alpha, s]$ formában.

2. Normalizáljuk a SIFT pont koordinátáit (x, y) , a szókép fizikai méretével - ezáltal a leíró vektorokban használt pozíciók relatív pozíciókká válnak.
3. Definiálunk egy R sugarú kört a query (Q) kép vizsgált kulcspontja körül – sz összehasonlítás során, csak a candidate kép (C) körön belül eső kulcspontjait vesszük figyelembe. A kör sugara (a normalizált távolság mértékek miatt) a query kép $R\%$ -ának tekinthető.
4. A körön belül eső candidate kulcspontokból a két minimális Euklideszi távolságút választjuk ki (a távolságot a query és az adott candidate egy-egy kulcspontja között, csak a 128 elemű leíró vektorra számítjuk):

$$D(q_i, c_j) = \sqrt{\sum_{k=1}^{128} (q_i(k) - c_j(k))^2} \quad (3.1)$$

$$c_{i,min} = \min_{c_j} D(q_i, c_j) \quad (3.2)$$

$$c_{i,min2} = \min_{c_j} D(q_i, c_j) \text{ s.t. } c_j \neq c_{i,min} \quad (3.3)$$

5. Küszöbértéket (T_α) alkalmazunk az orientáció különbségek ellenőrzésére:

$$|proj_\alpha(q_i) - proj_\alpha(c_{i,min})| < T_\alpha \quad (3.4)$$

Azt a kulcspont párt, mely nem teljesíti a feltételt, eldobjuk. Az adott q_i kulcsponthoz nem keresünk másik megoldást.

6. Küszöbértéket (T_D) alkalmazunk a $c_{i,min}$ kulcspont egyediségének ellenőrzésére:

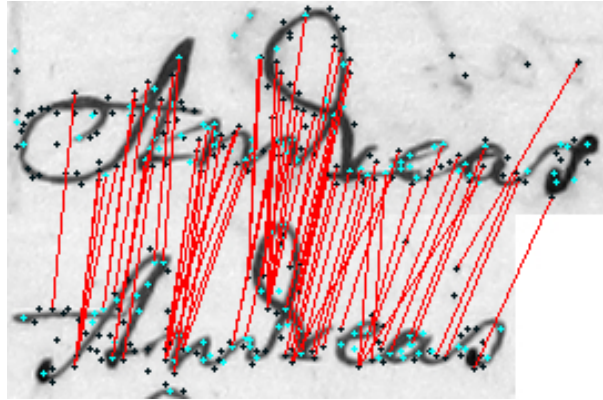
$$\frac{D(q_i, c_{i,min})}{D(q_i, c_{i,min2})} < T_D \quad (3.5)$$

Azokat a kulcspontokat melyek nem teljesítik a feltételt, a továbbiakban kulcspont párként nem vesszük figyelembe.

7. A feltételeket teljesítő pont párok elemei lesznek a két szóhoz (Q és C) tartozó kapcsolódó pontok halmazának: $M_{Q,C} = \{(q_i, c_{i,min}), i = 1, 2, \dots, N\}$, ahol N az összekapcsolt kulcspontok száma a Q queryben.
8. Kiszámítjuk a query és candidate szó közötti hasonlóság értéket, a kapcsolódó pontok felhasználásával:

$$S(Q, C) = \sum_{j=1}^N (\sqrt{255^2 \cdot 128} - D(q_j, c_j)_{(q_j, c_j) \in M_{Q,C}}) \quad (3.6)$$

9. A Q -hoz tartozó eredményeket csökkenő sorrendbe rendezzük.



3.1. ábra. Két kép között detektált kulcspont kapcsolatok

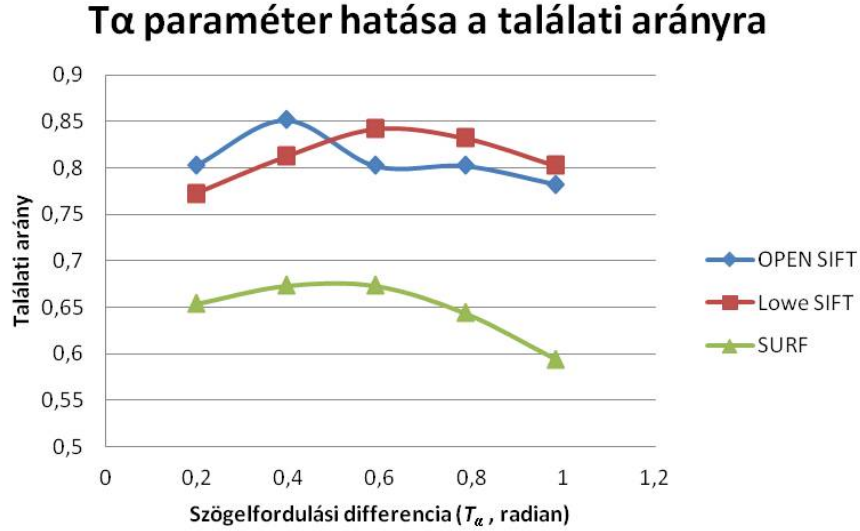
Két szóképek között detektált kulcspont párokat a 3.1. ábra szemlélteti. A bemutatott algoritmus a következő előnyökkel rendelkezik.

- a leírók skála és elforgatás invariánsak
- nincs szükség elő-feldolgozásra (például binarizáció, dőlésszög korrekció, vázásítás), illetve az apróbb szegmentálási hibákra is érzéketlen
- az adott kulcspont körüli keresési tér szimmetrikus, ellentétben a DTW-t alkalmazó módszerekkel, ahol visszafele keresés nem megengedett
- a SIFT leíró zajos környezetben is stabilan használható [12], mely jól illeszkedik az archív dokumentumok környezetére
- csak a magas információtartalmú, jellemző pontok kerülnek kiválasztásra és összehasonlításra

3.2. Kísérletek és eredmények

3.2.1. Tesztadatbázis

Kísérleteinkhez és algoritmusunk teszteléséhez a 2.2. pontban említett 1771-es összeírás első 22 oldalának kézzel szegmentált és annotált szavából felépülő adatbázist használtunk. A tesztadatbázis 177 név 1638 példányát tartalmazza. Egy szó átlagosan 9,25 példányban fordul elő az adatbázisban (1, illetve 111 szélsőértékekkel). Az adatbázis képein további elő-feldolgozás nem történt, a képek körülbelül 100-150 pixel magasak, különböző hosszúságúak, téglalap alakban szegmentálva – az esetleges belógó részek nem kerültek kitakarásra. A kísérletek során véletlenszerűen kiválasztott 101 szót választottunk querynek, melyekhez a maradék szópéldányok (candidates) közül kerestük a leghasonlóbbat.



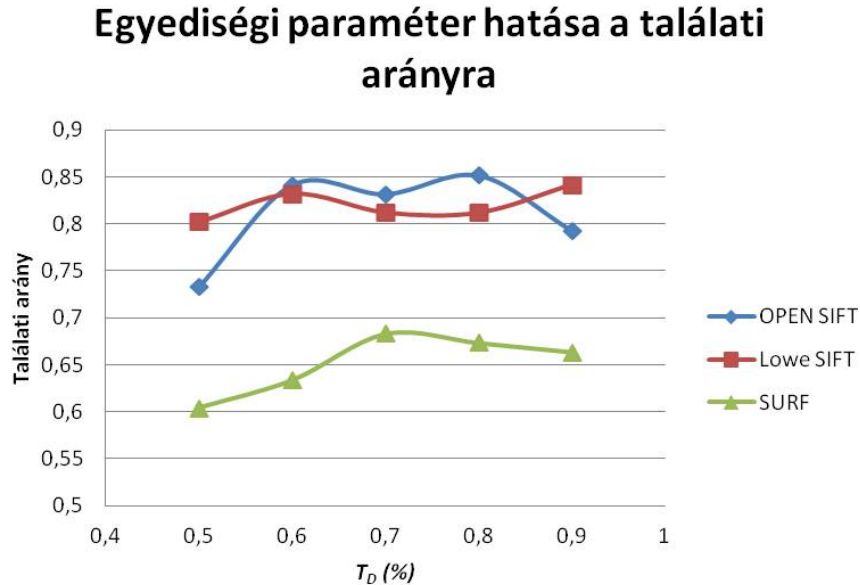
3.2. ábra. A kiértékelési eredmények a T_α küszöb érték függvényében

3.2.2. Lokális leírók összehasonlítása

Kísérleteinkben különböző SIFT implementációkat (Lowe-féle implementáció [12], OpenSIFT könyvtár [5]), illetve a SURF leírót használtuk a kulcspontok generálására. A leírókat különböző paraméter beállítások mellett teszteltük a távolság R , a szögelfordulás T_α és az egyediség T_D küszöb értékek esetében. A legjobb eredmény az első találatra 85% környékén alakult, az első tíz találatba 98%-os pontossággal került a megfelelő szó. Az első találatra vonatkozó eredményeket a különböző paraméter beállítások mellett a 3.4., 3.2., és 3.3. ábrán láthatjuk.

Az grafikonokról is leolvasható, hogy a legpontosabb eredmény az OpenSIFT könyvtár használatával, $R = 10\%$, $T_D = 0.8$, $T_\alpha = 23^\circ$ paraméter beállítások mellett kaptuk. A legjobb beállítások esetén véletlenszerűen kiválasztott query-kre az egy kísérletben előfordult tévesztéseket láthatjuk a 3.5. ábrán. Sárgával jelöltük azokat a szavakat, melyeknél ugyan az algoritmus nem megfelelő címkéjű szót adott vissza, de az eredmény alakilag nagyon hasonlít a lekérdezett szóhoz, annak rövidebb, vagy néhány betűvel kiegészített formája (például Catha - Cath, Joseph - Josephus), vagy az íráskép szemmel is alig megkülönböztethető betűket tartalmaz (pl. Jannes - Ivannes).

A SIFT leíró fő hátránya a nagy számításigény, mely új leíró fejlesztését motiválta. A SURF [1] az egyik leggyakrabban használt SIFT-hez hasonló, ám kisebb számításigényű, gyorsabb lokális tulajdonság leíró algoritmus. A kísérleteket SURF leíró alkalmazásával is elvégeztük (ez esetben a leíró vektorok $64 + 4$ dimenziósak voltak). A kísérletek eredményeiből leolvasható, hogy a SURF leíró alkalmazásával minden esetben jelentősen rosszabb felismerési teljesítményt értünk el – így a SURF leíró alkalmazását feladatunk esetében elvetettük.



3.3. ábra. A T_D küszöbérték hatása az eredményekre

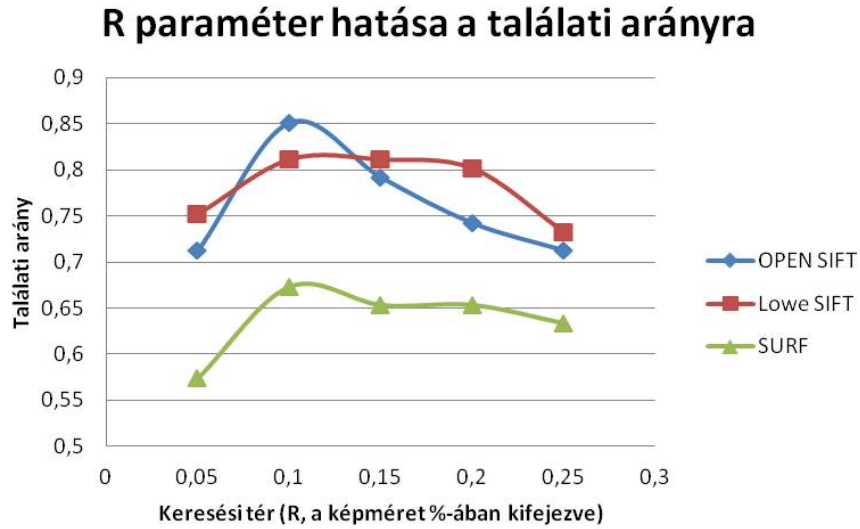
3.2.3. Skála és elforgatás invariancia vizsgálata

Több korábbi munkában ([16, 20, 21]) találtunk példát skála és rotáció függő lokális gradiens hisztogram alapú leírók alkalmazására karakter alapú kézírás felismerés esetén. Karakter alapú felismerés esetén az algoritmusok azonban viszonylag kevés osztályt különböztettek meg (pl. 10 arab számjegy), szó alapú felismerés esetén – munkánkban – az elkülönítendő osztályok száma jóval magasabb. Kísérleteink esetén minden esetben a fent említett optimális paraméter beállításokat alkalmaztuk.

Rögzített skála értékek

A 3.6. ábrán a SIFT skála értékek eloszlását ábrázoltuk, kézírásos dokumentumból generált több mint 300000 SIFT kulcspont esetében – az átlag 1,75, a szórás 1,33 körül alakult. Látható, hogy a skála értékek relatíve széles skálán helyezkednek el. Kérdés, hogy az összehasonlítás esetében mely skála információi a leginkább relevánsak.

A kérdés megválaszolásához nem elég az adott skálán található SIFT pontok szűrése a megadott skálára, hiszen a kulcspontok skála eloszlása kifejezetten az alsóbb régiókra koncentrált, valamint a kivont kulcspontok száma (utólagos szűrés esetében jelentősen változó érték) nagy mértékben befolyásolja az összehasonlítás eredményét. Módosítottuk tehát a SIFT leírót, oly módon, hogy a gradiens hisztogram kiszámítása előtt a skála értékeket egy-egy kijelölt értékre fixáltuk. A jellemző pontok keresése tehát változatlan maradt, azokat skála-tér teljes tartományában keressük, azonban a hozzájuk tartozó orientáció értékek, illetve gradiens hisztogram a képpiramis rögzített szintjén kerültek kiszámításra. A kivont kulcspontok számát a módosítás minimális mértékben



3.4. ábra. Az R paraméter hatása az eredményekre

változtatta meg (Az orientáció számításnál bekerülhetnek kettős orientációjú pontok, melyekhez külön leíró tartozik, vagyis ezeket külön kulcspontként reprezentáljuk.)

A különböző rögzített skála értékek esetén kapott eredményeket a 3.7. ábra szemlélteti. Látható, hogy bár az eredeti SIFT pont kivonás esetén az 5-ös skálán meglehetősen kevés kulcspont található (3.6. ábra), kizárólag ezt a skála szintet használva az összehasonlítás még is meglepően jó eredményt ad. Megfigyeléseink megegyeznek korábbi cikkek eredményeivel (pl. [20]). Magasabb skála értékek esetén (8-10) a kulcspont túl nagy képterületet „dolgoz fel”, mely a kézírásra és a karakterek jellegére jellemző részeket már csak kis részben azonosítja, így a felismerés pontossága (elvárt módon) romlik.

Rögzített orientáció értékek

Kísérleteket végeztünk az orientáció értékek esetében is. Vajon a SIFT leíró által biztosított elforgatás invariancia – melyet ugyan algoritmusunkban az elforgatás-különbség küszöb (T_α) segítségével kismértékben korlátozunk – szükséges-e a felismerés hatékonyságának biztosításához? A kérdés megválaszolásához ismételten a leíró módosítása szükséges. A SIFT leírót ez esetben az orientáció érték kiszámításánál változtattuk meg, az értéket egy-egy konstans értékre állítottuk, a gradiens hisztogramokat a módosított orientációnak megfelelően számítottuk ki. A skála értékeket ez esetben az eredeti módszernek megfelelő értéken hagytuk. A változtatás az egy képhez tartozó kulcspontok számában releváns változást ez esetben sem eredményezett.

Az eredmények rögzített orientáció értékek esetében a 3.8. ábrán látható módon alakultak. Az egyenletes módon jelentősen gyengébb felismerési teljesítményt tekintve kijelenthető, hogy a SIFT leíró orientáció invarianciája (bizonyos korlátozásokkal) különösen fontos tulajdonság a szókép alapú kézírás felismerésben.

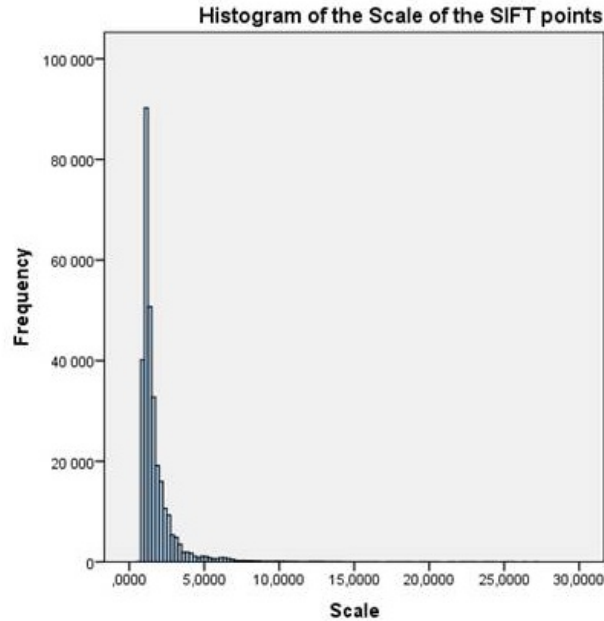
Ground truth	Wrong recognition	Ground truth	Wrong recognition
Andreas	Ivannes		
Anto	Anna		
Catha	Cath		
Cathar	Catharina		
Eva	Anna		
Filius	Viduus		
Jannes	Ivannes		
Joseph	Josephus		
Julia	Julian		
Math	Mich		
Paulus	Paul		
Sebastianus	Szephanus		
Susana	Susanna		
Vidua	Adam		
Viduus	Filius		

3.5. ábra. A tévesztett szavak listája 101 véletlen query esetén – a sárgával jelölt szavak bár nem a helyes címkét kapták, alakilag nagyon hasonlóak

A SURF leíró SIFT-el való összehasonlítását vizsgálja [6] – megállapításai szerint „A SURF gyors, és a SIFT-hez hasonlóan jó teljesítményű leíró, de érzékenyebb az elforgatásra, és a megvilágítás változásaira”. Az elforgatás invariancia fontossága ilyen típusú összehasonlítási feladatok esetében okozhatja a SURF 3.2.2. pontban tárgyalt (SIFT-hez képest) jelentősen rosszabb teljesítményét.

3.2.4. További kísérletek

Mivel több, az irodalmi áttekintésben bemutatott cikkben a tulajdonság kivonás előtt a képeket elő-feldolgozásnak vetették alá, megvizsgáltuk az elő-feldolgozási lépések (binarizálás, vázasítás) hatását SIFT alapú módszerünk esetében is. Az elő-feldolgozási lépések szükségességét általában azzal indokolják, hogy a felismerésben a szavak, betűk alakja fontos, nem a toll vastagság, vagy az aktuális szín. A SIFT leíró alkalmazásához a bináris képekből készített vázon enyhe elmosást alkalmaztunk. Megfigyeléseink sze-

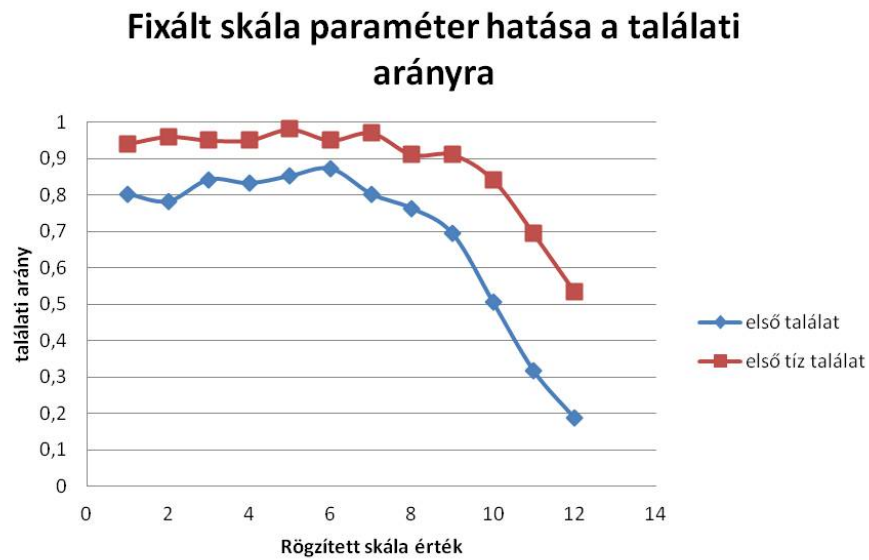


3.6. ábra. A skála értékek hisztogramja kézírásos dokumentumból kivont 301120 SIFT pont esetén

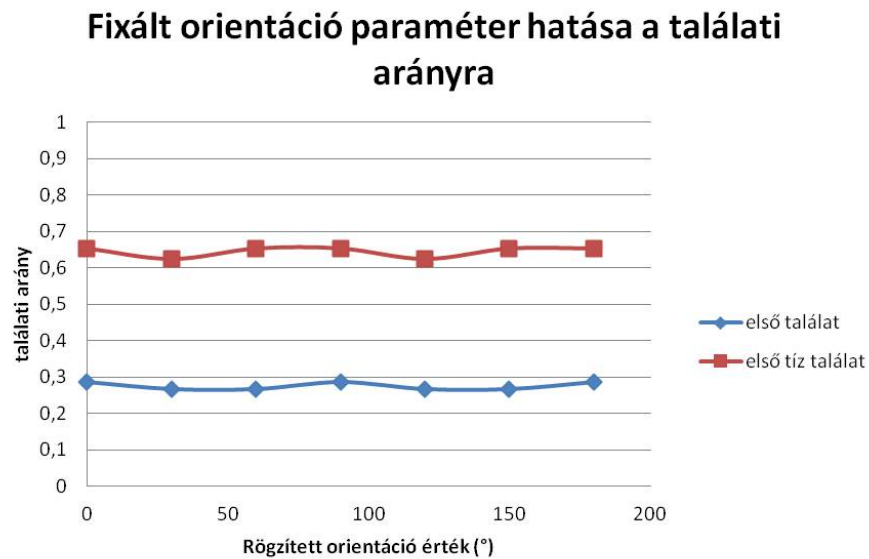
rint mind a vázasítás, mind a binarizálás növelte a módszer bizonytalanságát, értékes információt veszítettünk, a felismerési pontosság 5-7%-al visszaesett.

3.2.5. Eredmények értékelése

A kézírásos dokumentumok sok esetben zajjal erősen terheltek, nehezítve a precíz szegmentálást. Kézírásos dokumentumok esetén tehát zajra minél inkább érzéketlen tulajdonság kivonó és összehasonlító algoritmust kell alkalmazni. Tesztjeink alapján a SIFT lokális tulajdonság leíró jól használható kézírásos dokumentumok felismerésére. A SIFT vizsgálata során megállapítottuk, hogy a SIFT leíró elforgatás invarianciája különösen fontos tulajdonság a kézírás felismerő alkalmazások esetén. Az általunk kidolgozott SIFT leíró alapú összehasonlító metrika tesztjeink során 85%-os felismerési teljesítményt ért el az első találatnál, 90% feletti teljesítményt az első 10 találatba esés esetében, körülbelül 100 szó osztályt tartalmazó archív kézírásos dokumentum esetén. Meg kell említenünk, hogy mivel a keresés és összehasonlítás során az adott query az összes candidate példánnyal összehasonlításra kerül – az algoritmus futásideje a tudásbázis növekedésével lineárisan növekszik. Az algoritmus tehát nagy tömegű lekérdezés és nagyméretű tudásbázis esetén nem alkalmas azonnali, real-time válaszadásra. Mivel a kézírásos dokumentumok feldolgozása jelenlegi kézi feldolgozás esetén is meglehetősen hosszú folyamat, így a feldolgozási munkához nem feltétlenül szükséges az azonnali válasz biztosítása. Tesztjeink alapján elmondható, hogy algoritmusunk alkalmas a levéltári munka szoftveres támogatására – archív dokumentumok fél automatikus digitális feldolgozására.



3.7. ábra. A felismerési eredmények a rögzített skála értékek esetében



3.8. ábra. A felismerési hatékonyság a rögzített orientáció értékek függvényében

4. Archív kézirásos dokumentum feldolgozó szoftver

A 3.1. pontban bemutatott algoritmusunk minimális korlátozásokkal (futásidő) alkalmasnak mutatkozik a levéltári munka szoftveres támogatására. Korábbi tesztjeink (Onlineocr.net, Abby Finereader, SimpleOCR)[2] alapján elmondható, hogy a feladatban szereplő archív kézirásos dokumentumok feldolgozásának támogatására jelenleg alkalmazható rendszer nem áll rendelkezésre. Az algoritmushoz tehát elkészült egy kézirásos dokumentumok digitális feldolgozását elősegítő szoftver környezet. A feldolgozás során az archív dokumentumok szkennelt, vagy fotózott, jobb esetben könyvszkennerral digitalizált változatai állnak bemenetként rendelkezésre. Az elvárt kimenet a fél automatikus feldolgozás során a dokumentum oldalain található szavak listája, esetleg azok az eredeti dokumentumhoz hasonló táblázatos elrendezése. Az adatok ilyen formában való reprezentációja már lehetővé teszi a kutatóknak a dokumentumokban való keresés, illetve statisztikai elemzések készítésének lehetőségét.

4.1. Igények felmérése

A kutatók, levéltárosok által könnyen használható szoftver elkészítéséhez szükséges dokumentum feldolgozásnál felmerülő igények felmérése. Megbeszélést kezdeményeztünk tehát a levéltár munkatársaival, amin a Képfeldolgozás Kutatólaboratórium munkatársai, illetve levéltári dolgozók, a Veszprém Megyei Levéltár igazgatója és munkatársai is részt vettek. A megbeszélésen bemutattuk a fejlesztett program vázát, működését, illetve információkat gyűjtöttünk a felhasználói igényekről, funkcionális és nem funkcionális követelményekről, a felhasználás körülményeiről, valamint a dokumentum feldolgozás jelenleg alkalmazott menetéről.

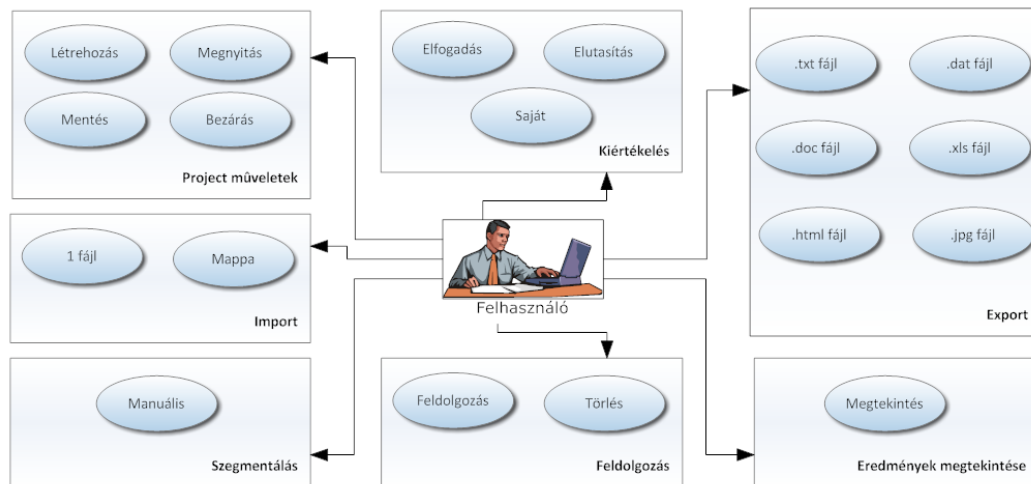
Az igényfelmérés során szerzett legfontosabb tapasztalatok:

- a feldolgozandó dokumentumok változatos hosszúságúak, 40 oldaltól kb. 1000 oldalig, az átlagos terjedelem 1000 oldal.
- a közigazgatásban keletkezett dokumentumok nagy részét aránylag szép, hivatalos dokumentumokban használatos írásképpel írták, azonban az írásminták az író egyéni sajátosságait tartalmazzák
- a dokumentumokban gyakran alkalmaznak rövidítéseket

- egy 70-100 oldalas dokumentum kézi feldolgozása jelenleg 4-5 hétig, akár több hónapig is tarthat
- a szó felismerésében hasonló alakú szavak, illetve a szöveggörnyezet segíthetik a levéltáros munkáját
- a dokumentumok egy csoportja táblázatos formában íródott, a különböző oszlopok azonosítása után az azokban az oszlopokban alkalmazott szókészlet azonosítható, és jóval szűkebb a teljes dokumentum szókészleténél

Feldolgozandó dokumentumok nagy tömegben, (levéltári megfogalmazás szerint a kapacitásokhoz képest végtelen számban) állnak rendelkezésre. Az általunk készített szoftver és az algoritmus tesztekben elért 85%-os teljesítménye a levéltári dolgozók szerint fél automatikus módban alkalmas a feldolgozási munka segítségére, illetve gyorsítására. A visszakeresés esetleges lassú válaszüzeje (tesztjeinkben nagy mennyiségű lekérdezés esetén néhány óra) a teljes feldolgozás összes idejéhez képest elenyésző, vagyis az elkészült rendszer használható, amennyiben a visszakeresési rész önállóan, felhasználói felügyelet nélkül végrehajtható – a keresés eredményét azonban a keresési fázis után kontrollálni kell.

4.2. Használati esetek



4.1. ábra. A *Handwritten Document Processor* használati esetei.

A felhasználói igények alapján a kézírási dokumentum feldolgozó (*Handwritten Document Processor*) szoftver esetében a következő alapvető használati esetéket azonosítottuk (4.1. ábra). A dokumentum feldolgozó szoftver alapvetően a feldolgozó szakember (levéltáros, vagy kutató) használja, vagyis egy felhasználó aktort azonosítottunk. A szoftver a feldolgozási munkát a könnyebb kezelhetőség, megszakíthatóság biztosítása érdekében projektekbe szerveztük – *projekt műveletekkel* a munka aktuális állását

menthetjük, azt újra betölthetjük, vagy új projektet nyithatunk. Minden projekt saját tudásbázist (korábban annotált képek) használhat. Az *importálás* során a bemeneti képeket adhatjuk meg a programnak (digitalizált kézírásos oldalak). A *szegmentálás* során a dokumentum teljes oldalából a feldolgozásra, felismerésre szánt szavakat jelölhetjük ki. A *feldolgozás* során az előbbieken kijelölt szavakból kivonjuk a SIFT leírókat, majd az általunk kidolgozott visszakereső modul segítségével alakilag hasonló annotált képet keresünk a tudásbázisból. A *kiértékelés* használati esetek segítségével az algoritmus találatait tekinthetjük meg, illetve értékelhetjük. A találatokat elfogadhatjuk, vagy elutasíthatjuk és a szónak saját annotációt adhatunk. Az *exportálás* és *eredmény-megtekintés* használati esetek a dokumentum feldolgozott formájának különböző formában való megjelenítésére és mentésére szolgálnak.



4.2. ábra. A szoftver által megvalósított feldolgozási szekvencia

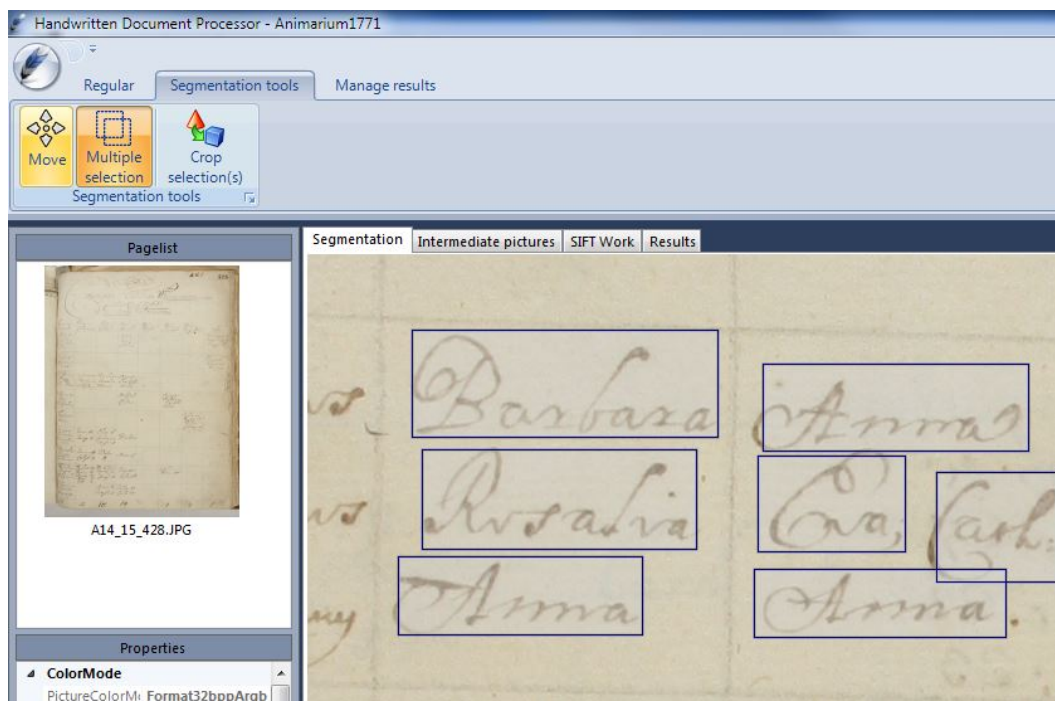
Látható, hogy a használati esetek a szoftver valós felhasználása során nagy részben egy feldolgozási szekvencia (4.2. ábra) részei. Az alapvető feldolgozási lépések a következők:

1. digitalizált oldalkép betöltése
2. szavak szegmentálása
3. SIFT pontok kivonása a szavak képeiből
4. összehasonlítás a tudásbázis képeivel, a leghasonlóbb kiválasztása
5. a keresési eredmény megjelenítése, ellenőrzése
6. feldolgozott dokumentum oldalak megjelenítése

A szoftver használata során az oldal teljes digitalizált képéből a szegmentálás során kis szóképeket alakítunk ki. A kis szóképekre kiszámítjuk a SIFT leírókat, majd ezek alapján hasonlósági értékeket állapítunk meg a tudásbázisban levő képekhez. A legmagasabb hasonlósági értéket kapott kép címkéjét adjuk alapértelmezett értéknek az új kérés (query szó) címkéjéhez (annotáció = jelentés). A címkézés a felhasználó beavatkozásával történik, a címkéjét lecserélheti az alacsonyabb hasonlósági értéket kapott címkékre, vagy egyéni címkét adhat meg. Az elfogadott címkéjű szavak bekerülnek a tudásbázisba.

4.3. A folyamatokhoz biztosított szoftvereszközök

4.3.1. Szegmentálás



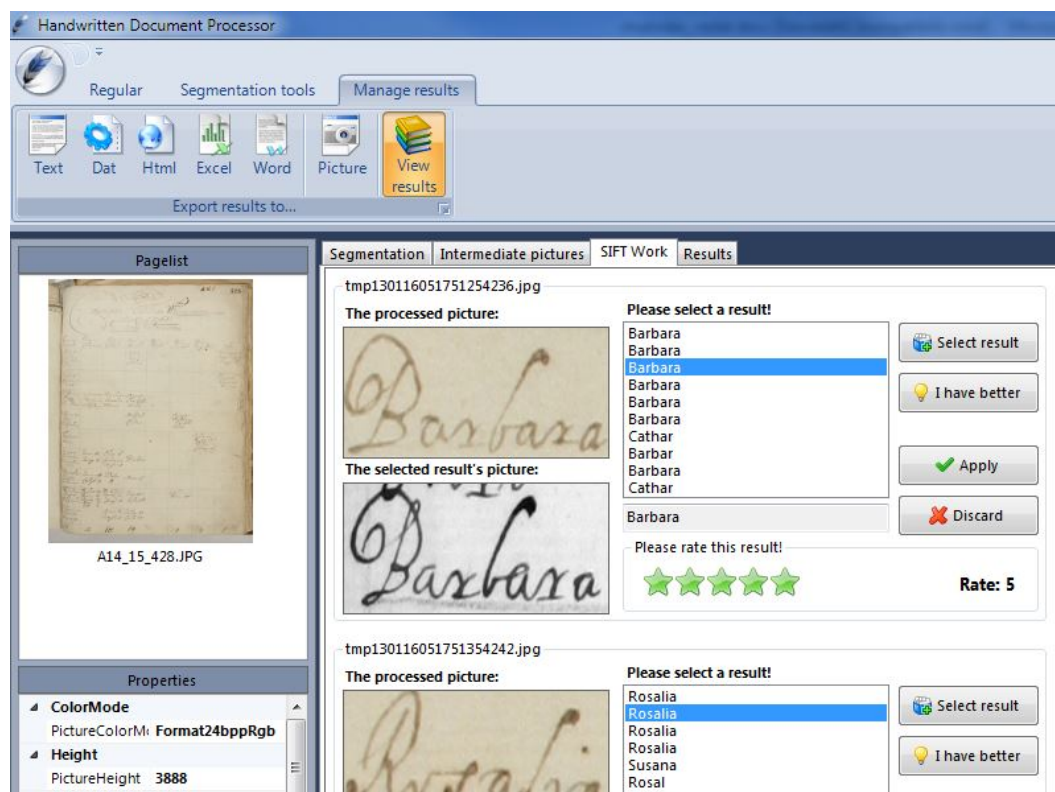
4.3. ábra. A *Handwritten Document Processor* szegmentáló felülete

A szegmentáláshoz a betöltött digitalizált dokumentum egy oldalát lehet betölteni a szegmentáló mezőbe. Az automatikus szegmentálás, vagy fél automatikus szegmentálás (előszegmentálás) ugyan nagyban segítené a feldolgozást, illetve gyorsítaná annak sebességét, azonban a zajos, átlapolódó, nem sorfolytonos dokumentumok szegmentálása önálló, nehéz problémának tekinthető. Az automatikus szegmentálás problémakörének kutatását és megvalósítását további munkánkban tervezzük, a szoftver jelenlegi fázisában a szavak szegmentálását (téglalap alakú kivágását) a felhasználónak manuálisan kell elvégezni. A feladat elvégzéséhez egy többszörös kijelölést támogató kijelölő eszköz áll a felhasználó rendelkezésére, mellyel az oldalon téglalap alakú területeket tud feldolgozásra kijelölni. Megjegyezzük, hogy a SIFT alapú algoritmus (3.1. pont) egyik jelentős előnyének tekinthető, hogy viszonylag érzéketlen a szegmentálás pontatlanságaira (belógó betűk, nem alapvonallal párhuzamos kivágás) – így elégséges téglalap alakú kijelölés alkalmazása, szabálytalan alakú kijelölésre, vagy a kijelölés elforgatására nincs szükség. A szegmentáláshoz rendelkezésre álló felhasználói felületet a 4.3. ábra szemlélteti. A szegmentálás után a program a kijelölt területeket kivágja és külön képként menti.

A szegmentált, kimentett, kiválasztott képekre indítható a feldolgozás – mely során a szoftver megkeresi a kulcspontokat, kivonja a tulajdonság leírókat, majd elindítja a

visszakereső modult. A keresési folyamat nagy mennyiségű lekérdezés esetén hosszabb időt is igénybe vehet, ezalatt felhasználói felügyeletet a szoftver nem igényel. A keresőmotor minden egyes kéréshez (szegmentált szóképhez) egy tíz elemből álló hasonlósági listát ad vissza, a tudásbázis alakilag leghasonlóbb elemeiről. A keresőmotor által adott eredmény ezután az eredmények értékelése fülön értékelhető.

4.3.2. Eredmények értékelése

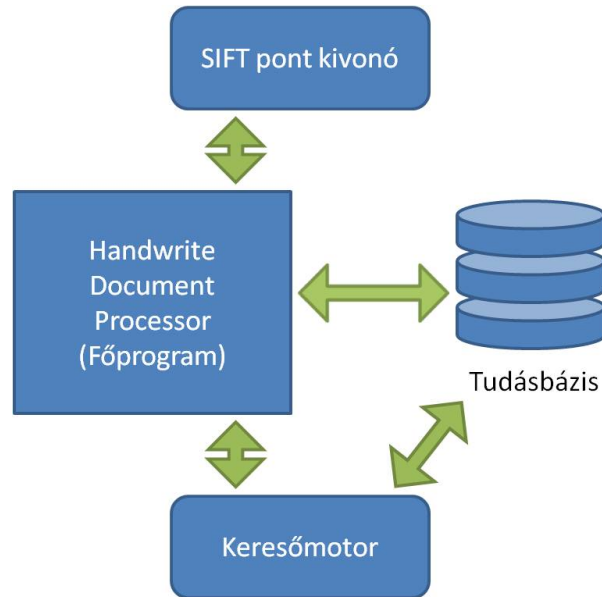


4.4. ábra. A szoftver eredmény értékelő felülete

Az eredmények értékelése során a felhasználó számára kijelzésre kerülnek az adott kérések (szegmentált képek), illetve a hozzájuk tartozó hasonlósági listák, valamint a kiválasztott (alapértelmezettként első találat) találat képe is segítségként megjelenítésre került. A megjelenített képek alapján a felhasználó dönthet az alapértelmezett találat címkéként való elfogadásáról, vagy másik találat választásáról a tízelemű találati listából (a kiválasztással egy időben az adott találathoz található kép megjelenik, ezáltal a két képalak összehasonlítható). Amennyiben a listában található egyetlen kép illetve címke sem felel meg az adott kérésnek a (szakértő) felhasználó szerint, saját címkét adhat meg a képnek. A levéltárosok ajánlásai alapján kifejlesztett eredmény értékelő felületet a 4.4. ábrán szemléltetjük.

4.4. Technikai megvalósítás

A szoftver technikai megvalósítása során alapvetően négy blokkot azonosíthatunk. A főprogram (Handwritten Document Processor) működését jelenleg megvalósított formájában két szoftver komponens segíti (SIFT tulajdonság kivonó modul, illetve SIFT alapú keresőmotor), valamint egy tudásbázis egészíti ki (4.5. ábra).



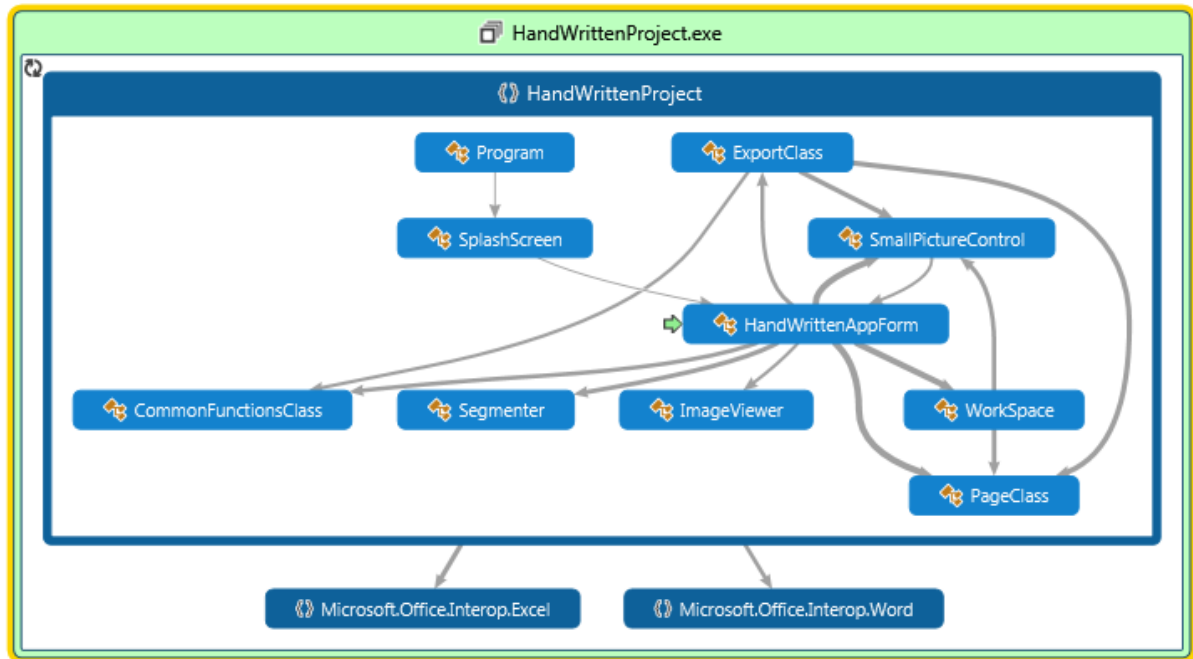
4.5. ábra. A Handwritten Document Processor blokkábrája

4.4.1. Handwritten Document Processor

A program jövőbeli felhasználói döntő többségben Microsoft operációs rendszert használnak (levéltári gyakorlat). A szoftvert így Microsoft operációs rendszerrel .NET környezet felhasználásával C# nyelven fejlesztettük. A program moduláris felépítésű, az egyes használati eset csoportonként felelős részletek külön modulba kerültek, az egyes modulok önállóan lettek implementálva. A főprogram és a külső komponensek (kulcs-pont kivonó, keresőmotor, tudásbázis) között alapvetően külső eljárashívási és fájlkezelési eszközökkel teremtünk kapcsolatot. A főprogram fontosabb funkcionális osztályairól a 4.6. ábra ad áttekintést.

4.4.2. SIFT pont kivonó szoftver

A SIFT pont kivonó komponens kialakításakor a tesztjeinkben (3.2.2. pont) legjobb eredményt adott, nyílt forráskódú (alapvetően Linuxos felhasználásra szánt) OpenSIFT könyvtár[5] C++ nyelven, Microsoft környezetre újrainplementált változatát használtuk. A komponens a bemeneti képekre megkeresi a képre jellemző kulcspontokat, illetve az azokhoz tartozó skála, orientáció értékeket és a leíró gradiens hisztogramot határozza



4.6. ábra. A főprogram fontosabb osztályai

meg. A kiszámított adatokat a komponens által előállított, a bemeneti képpel azonos nevű fájl tartalmazza. A komponens felhasználja az OpenCV képfeldolgozási célkönyvtár függvényeit.

4.4.3. Keresőmotor

Az általunk fejlesztett SIFT alapú visszakereső algoritmust (3.1. pont) MS Visual Studio 2008 környezetben C++ nyelven implementáltuk. Az implementáció során szintén felhasználásra kerültek OpenCV könyvtári függvények. A komponens bemenetként a query képet és a hozzá tartozó kulcsfájlt, illetve a tudásbázis elérhetőségét kapja meg. A tudásbázisban található példányokkal való összehasonlítás után kimenetként a tíz alakilag leghasonlóbb kép elérési útját, illetve a hasonlósági értékeket adja vissza. A keresőmotor implementálása során a korábban bemutatott tesztelés során megállapított optimális küszöb paraméterek kerültek felhasználásra.

4.4.4. Tudásbázis

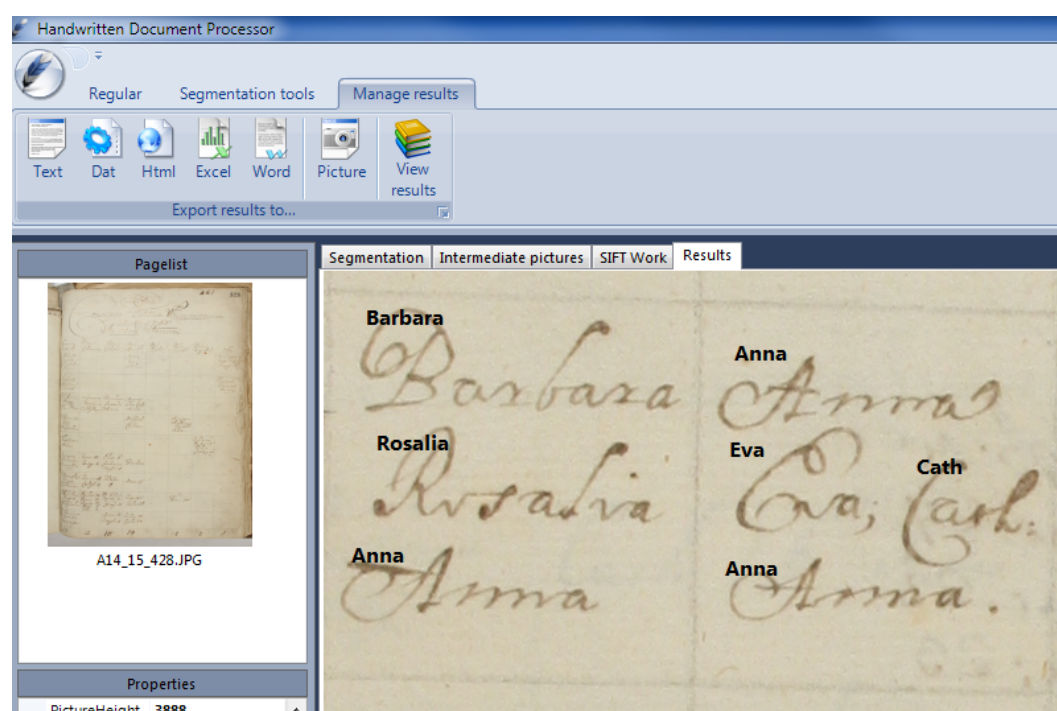
A tudásbázis gyakorlati megvalósítása egy mappa, mely a szóképeket, illetve a hozzájuk tartozó kulcsfájlokat tartalmazza. Egy projekt létrehozásánál a tudásbázist reprezentáló mappa paraméterként adható meg. Az egész projekt során ezt a tudásbázist alkalmazzuk a lekérdezésekre.

A tudásbázis természetesen adatbázis, és adatbázis kezelési eszközök segítségével is megvalósítható lenne, a fájlkezelés szintű megvalósítást a telepítés megkönnyítése,

illetve a felületes informatikai tudással rendelkező humán beállítottságú felhasználók számára is könnyű kezelhetőség miatt választottuk. A levéltári alkalmazottak így a megszokott eszközeikkel (fájlkezelés) létrehozhatnak akár tematikus tudásbázisokat egy-egy írásképre, szövegtípusra, vagy korszaki írásképre.

A szoftverhez külön tanító komponens nem készült, a szoftver használatát (különböző írásmintákat tartalmazó dokumentumonként) először üres tudásbázissal kell kezdeni – ez esetben a keresőmotor üres listákkal tér vissza a feldolgozás után, az eredmények értékelésénél a szavak címkéit kézzel kell megadni. A címkék elfogadása után az új szavak már a tudásbázis részét képezik. Az szoftver használata során tehát az első pár oldal szavait szegmentálás után kézzel megadva felépítjük az alap tudásbázist, majd a munka a további oldalakkal már a keresőmotor valós találatával folytatható.

4.5. A dokumentum feldolgozó szoftver kimenete és értékelése



4.7. ábra. A szoftverben megtekinthető kimenet

A feldolgozás eredményének megtekintésére a szoftver biztosít lehetőséget, illetve a feldolgozás eredmény exportálható különböző formátumokba. A szoftver kimenete tehát a feldolgozott dokumentum oldalak különböző formátumú leírása. A formátumokat a további (levéltári) kutatások elősegítése, illetve a korábban feldolgozott dokumentumok kézzel előállított eredményei alapján állapítottuk meg. A txt, illetve MS Word alapú kimenetben mentésre kerül az oldalon található szavak listája, azok pozíciója. Az MS

Excel alapú kimenetben az eredeti dokumentum oldalhoz hasonló elrendezést láthatunk, a HTML alapú kimenet az eredeti kézírásos oldalt és a feldolgozott szöveget egymásra lapolva képes megjeleníteni (4.7. ábra) – ez a formátum képként is menthető.

A bemutatott szoftver képes archív kézírásos dokumentumok digitális feldolgozásának szoftveres támogatására, a levéltárosok munkájának könnyítésére, segítésére. A módszer az automatikus szegmentáló modul kifejlesztésével és beépítésével válna teljessé, de algoritmusunk teszteken elért teljesítménye, illetve a hozzá kialakított felhasználó és munkafolyamat barát szoftverkörnyezet-kialakításnak köszönhetően jelen formájában is hatékony, és felhasználható valós feladatokhoz.

5. Összefoglalás

A dolgozatban bemutatott projekt keretében egy célszoftvert készítettünk az archív történelmi dokumentumok egy csoportja (például népszámlálások, összeírások, anyakönyvek) digitális feldolgozásának elősegítésére. Áttekintettük a témához kapcsolódó irodalmat, a karakter, illetve szókép alapú, főként lokális leírókat tartalmazó módszereket. Kidolgoztunk egy szókép alapú felismerő algoritmust, mely a kép lokális jellemző pontjainak kivonásával és összehasonlításával megadja a szóhoz leginkább hasonlító – korábban már látott - szóképet. Az algoritmus kialakításánál vizsgáltuk különböző lokális leírók hatékonyságát, illetve a választott és hatékonynak bizonyuló SIFT leíró esetében az elforgatás és skála invariancia szükségességét. Megállapítottuk, hogy a SIFT leíró orientáció invarianciája (bizonyos korlátozásokkal) különösen fontos tulajdonság az algoritmus kézírás-felismerés területén való hatékony működéséhez. 1638 valós archív szövegből annotált szó példányt tartalmazó teszt adatbázisunkra algoritmusunkkal 85%-os felismerési teljesítményt értünk el, az első tíz találatba 90% feletti valószínűséggel esik helyes találat.

Az elkészült algoritmust egy a levéltári munkában, nem informatikusok számára is könnyen kezelhető szoftver környezetbe foglaltuk, mely a dokumentum digitális feldolgozásának munkafolyamatát támogatja. A levéltári igényekre szabott szoftver fél automatikus módon támogatja a dokumentum feldolgozás lépéseit, az előállított kimenet alkalmas és fontos információkat tartalmaz a levéltári, illetve a történész szakma kutatásai számára. Az elkészült szoftvert a levéltár munkatársai hasznosnak tartják, véleményük szerint munkájukat nagy mértékben segítheti – reméljük hamarosan megkezdik a szoftver használatát valós kutatási feladatokon.

Irodalomjegyzék

- [1] Herbert Bay – Andreas Ess – Tinne Tuytelaars – Luc Van Gool: Speeded-Up Robust Features (SURF). *Comput. Vis. Image Underst.*, 110. évf. (2008) 3. sz., 346–359. o.
- [2] László Czúni – Tamás Szőke – Mónika Gál: Recognition of Hand-written Archive Text Documents. In *Proceedings of the 2012 International Conference on Computer Vision and Graphics (ICCVG'12)* (konferenciaanyag). 2012, 337–344. o.
- [3] A. Fischer – A. Keller – V. Frinken – H. Bunke: Hmm-based word spotting in handwritten documents using subword models. In *Pattern Recognition (ICPR), 2010 20th International Conference on* (konferenciaanyag). 2010, 3416–3419. o. ISSN 1051-4651.
- [4] A. Fischer – K. Riesen – H. Bunke: Graph Similarity Features for HMM-Based Handwriting Recognition in Historical Documents. In *Frontiers in Handwriting Recognition (ICFHR), 2010 International Conference* (konferenciaanyag).
- [5] Rob Hess: An open-source SIFTLibrary. In *Proceedings of the International Conference on Multimedia (MM '10)*. ACM (konferenciaanyag). 2010, 1493–1496. o.
- [6] Luo Juan – Oubong Gwun: A comparison of SIFT, PCA-SIFT and SURF. *International Journal of Image Processing (IJIP)*, 2009., 143–152. o.
- [7] Orosz Katalin – Rácz György – Reisz T. Csaba – Vajk Ádám – Véber János: Középkori oklevelek tömeges digitalizálása, 2008.
- [8] Vladimir Kluzner – Asaf Tzadok – Yuval Shimony – Eugene Walach – Apostolos Antonacopoulos:.
- [9] V. Lavrenko – T. M. Rath – R. Manmatha: Holistic Word Recognition for Handwritten Historical Documents. In *International Workshop on Document Image Analysis for Libraries* (konferenciaanyag). 2004, 278–287. o.
- [10] A. Lawrence Spitz: Using Character Shape Code for word spotting in document images, 1995.
- [11] Y. Liang – M.C. Fairhurst – R.M. Guest: A Synthesised Word Approach to Word Retrieval in Handwritten Documents. *Pattern Recognition*, 45. évf. (2012), 4225–4236. o.

- [12] D. G. Lowe: Object Recognition from Local Scale-invariant Features. In *Proceedings of the International Conference on Computer Vision 2* (konferenciaanyag). 1999, 1150–1157. o.
- [13] David G. Lowe: Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60. évf. (2004) 2. sz., 91–110. o.
- [14] T. M. Rath–R. Manmatha: Word Spotting for Historical Documents. *Int. Journal on Document Analysis and Recognition*, 2007., 139–152. o.
- [15] Toni M. Rath–R. Manmatha: Word image matching using dynamic time warping, 2002.
- [16] José A. Rodríguez–Florent Perronnin: Local Gradient Histogram Features for Word Spotting in Unconstrained Handwritten Documents. In *Int. Conf. on Frontiers in Handwriting Recognition* (konferenciaanyag). 2008, 7–12. o.
- [17] Jamie L. Rothfeder–S. Feng–Toni M. Rath: Using Corner Feature Correspondences to Rank Word Images by Similarity. In *Computer Vision and Pattern Recognition Workshop, CVPRW '03*. (konferenciaanyag), 3. köt.
- [18] Sayantan Sarkar: Word Spotting in Cursive Handwritten Documents Using Modified Character Shape Codes. *Advances in Intelligent Systems and Computing*, 2013.
- [19] Szőke Tamás: Zajos, archív kézirásos dokumentumok vizsgálata, szavainak felismerése. *Diplomadolgozat*, Pannon Egyetem, Veszprém, 2012.
- [20] S. Uchida–M. Liwicki: Part-Based Recognition of Handwritten Characters. In *Frontiers in Handwriting Recognition (ICFHR), 2010 International Conference on* (konferenciaanyag). 2010, 545–550. o.
- [21] Zhiyi Zhang–Lianwen Jin–Kai Ding–Xue Gao: Character-SIFT: A Novel Feature for Offline Handwritten Chinese Character Recognition. In *Document Analysis and Recognition, ICDAR '09. 10th International Conference on* (konferenciaanyag). 2009, 763–767. o.