GRUPIRANJE TEKSTA V SLIKAH NARAVNIH SCEN

Andrej Ikica, Peter Peer

Laboratorij za računalniški vid
Fakulteta za računalništvo in informatiko, Univerza v Ljubljani
E-pošta: {andrej.ikica, peter.peer}@fri.uni-lj.si
URL: http://www.fri.uni-lj.si/si/laboratoriji/lrv/

POVZETEK: *Avtomatična razpoznava teksta v slikah naravnih scen postaja v zadnjih letih zelo popularna – tako na raziskovalnem kot na aplikativnem področju. Da bi bila razpoznava teksta čimbolj natančna, je potrebno predhodno tekst v sliki pravilno detektirati ter ga ustrezno segmentirati v posamezne vrstice in besede. Pravilna segmentacija je namreč predpogoj za dobro delovanje same razpoznave teksta. V članku opisujemo metodo za grupiranje črk detektiranih v slikah naravnih scen v posamezne vrstice. Metoda temelji na izgradnji minimalnega vpetega drevesa, katerega vozlišča so posamezne detektirane črke, ter iskanju optimalnih poddreves, ki ustrezajo posameznim vrsticam v tekstu. Iskanje optimalnih poddreves je optimizacijski problem, ki temelji na minimizaciji skupne energije vseh poddreves. Metoda je evalvirana na zbirki slik teksta v naravnih scenah CVL OCR DB.*

1. UVOD

Z izrazom *slike teksta v naravnih scenah* označujemo slike raznovrstnih scen, v katerih se pojavlja poljuben tekst različnih velikosti, barv in oblik. Večinoma gre za slike vsakodnevnih scen, posnete z mobilnimi telefoni, fotoaparati in videokamerami, ki so tipično slabše kvalitete. Slika 1 prikazuje nekaj primerov takšnih slik.

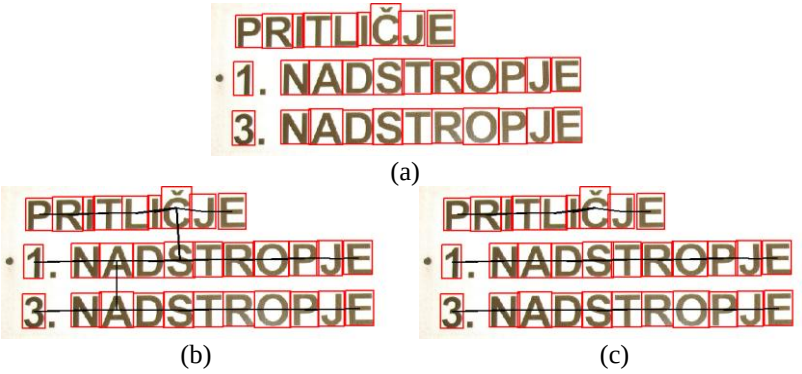
Slika 1: Primeri teksta v slikah naravnih scen

Raziskovalna skupnost se z razpoznavo teksta v slikah naravnih scen ukvarja že vrsto let, v zadnjem času pa se je zanimanje za to področje še dodatno povečalo, saj ima velik aplikativni potencial tudi na področju mobilnih naprav. Kot primer navedimo avtomatično prevajanje napisov posnetih z mobilnimi telefoni.

Slike naravnih scen vsebujejo zelo kompleksna ozadja, so slabše kvalitete, zajete so pod spremenljivimi svetlobnimi pogoji in so podvržene perspektivnim projekcijam, zato za razliko od klasične razpoznave teksta v standardnih črno-belih tiskanih dokumentih razpoznava teksta v slikah naravnih scen predstavlja zelo trd oreh. Pred samo fazo razpoznave je tekst potrebno natančno detektirati, tj. določiti lokacijo v sliki, kjer se nahaja, ter ga pravilno grupirati v posamezne vrstice oz. besede, ki se uporabijo kot vhod v klasifikator OCR (Optical Character Recognition).

V nadaljevanju podajamo opis metode, ki učinkovito grupira tekst v posamezne vrstice. Metoda iz posameznih črk v sliki generira minimalno vpeto drevo, pri čemer vozlišča drevesa ustrezajo črkam, povezave pa povezujejo pare sosednih črk. Segmentacija teksta na posamezne vrstice poteka z rezanjem odvečnih povezav, tj. povezav, ki povezujejo črke v različnih vrsticah. Rezanje povezav in generiranje optimalnih poddreves je optimizacijski problem, ki minimizira skupno energijo vseh poddreves.

Metodo smo evalvirali na zbirki slik teksta v naravnih scenah CVL OCR DB (Computer Vision Lab OCR DataBase) [1], ki trenutno vsebuje 341 raznovrstnih anotiranih slik teksta v naravnih scenah, zajetih pod različnimi svetlobnimi pogoji. Slike v zbirki so ročno anotirane, tako vsaki sliki v zbirki pripada datoteka XML, ki vsebuje informacijo o lokaciji teksta v pripadajoči sliki, informacijo o lokaciji posameznih črk ter izrezane sličice črk, ki služijo učenju in testiranju klasifikatorjev OCR za naravne scene.



Slika 2: (a) Detektirani okvirji črk, (b) minimalno vpeto drevo in (c) povezavi na sliki b, ki »napačno« povezujeta vrstici, sta odrezani

2. METODA GRUPIRANJA TEKSTA V VRSTICE

Metoda grupiranja teksta v vrstice je sestavljena iz treh korakov. V prvem koraku z metodo SWT (Stroke Width Transform) [2] detektiramo posamezne črke v sliki. Rezultat detekcije so okvirji (angl. bounding box), ki obdajajo črke (slika 2a). V drugem koraku na podlagi posebne metrike definiramo razdalje med okvirji črk [3] in generiramo minimalno vpeto drevo, katerega vozlišča so okvirji črk, povezave pa ustrezajo

povezavam med sosednimi okvirji (slika 2b). Ker minimalno vpeto drevo vsebuje tudi povezave med vrsticami, je potrebno te povezave identificirati in jih odrezati (slika 2c).

2.1 Detekcija črk v sliki

Posamezne črke v sliki detektiramo s pomočjo metode SWT [2], ki vsakemu slikovnemu elementu slike priredi debelino loka črke (angl. stroke), ki ji pripada. Metoda deluje tako, da iz vsakega slikovnega elementa v sliki potuje v smeri gradienta, dokler ne naleti na slikovni element s približno nasprotnim gradientom. Metoda dolžino preiskovalnega žarka zapiše v vse slikovne elemente, ki jih prečka na poti. Ko je slikovni element del črke, postopek iskanja najverjetneje naleti na nasprotni rob (zaradi karakteristike črk, ki so sestavljene iz dveh vzporednih kontur), v primeru, ko slikovni element ni del črke, pa preiskovalni žarek potuje iz slike. Slika 3 prikazuje rezultat transformacije slike z metodo SWT.

(a) (b)

Slika 3: Delovanje metode SWT: (a) originalna slika in (b) SWT transformacija – v primeru črk in črkam podobnih struktur se preiskovalni žarek ustavi na nasprotnem robu črke, v ostalih primerih pa potuje v neskončnost, kar prikazujejo osamele črte na sliki

Pa transformaciji slike z metodo SWT se slikovni elementi, ki se dotikajo in imajo podobno debelino loka črke, grupirajo v povezane komponente (angl. connected components). Komponente, ki ne ustrezajo določenim geometrijskim pravilom značilnim za črke (velikost, razmerje višine in širine ipd.), so izločene. Preostale komponente po vsej verjetnosti ustrezajo črkam v sliki, zato vstopajo v naslednjo fazo, tj. fazo generiranja minimalnega vpetega drevesa.

2.2 Izgradnja minimalnega vpetega drevesa

Ko na sliki detektiramo okvirje posameznih črk (slika 2a), jih povežemo v minimalno vpeto drevo. Povezovanje poteka na podlagi sosednosti med okvirji črk. Kljub temu, da je za človeka identifikacija sosednih črk sila enostavna, je ta postopek za računalnik izredno kompleksen. Ni nujno namreč, da sta sosedni črki ravno tisti, ki ležita evklidsko najbližje. Prav lahko se zgodi, da je evklidska razdalja do črke v sosedni vrstici krajša od

razdalje do sosedne črke v isti vrstici. Zato se pri izgradnji minimalnega vpetega drevesa ne uporablja klasična evklidska razdalja med črkami, temveč kompleksnejša metrika.

Podobno kot v [3] je metrika za razdaljo med parom črk C_i in C_j definirana kot linearna kombinacija obteženih značilk:

$$dist(C_i, C_j) = W_d \cdot F_{ij} \tag{1}$$

pri čemer je W_d vektor uteži, F_{ij} pa je vektor značilk posameznega para črk, ki ga sestavljajo razmerje med višinama obeh črk, razmerje med širinama obeh črk ter horizontalna in vertikalna razdalja med črkama. Vrednosti uteži določimo z metodo LMS (Least Mean Squares) [4]. Kot učno množico za učenje uteži smo izbrali podmnožico slik zbirke CVL OCR DB. Na vsaki izmed učnih slik smo določili pozitivne in negativne primere, tj. pare sosednih in nesosednih črk. Da bi se izognili kombinatorični eksploziji, ki jo prinaša število kombinacij parov črk v sliki, smo kot učne pare črk definirali le tiste pare črk, ki imajo skupne Voronoieve robove [5].

Gradnja minimalnega vpetega drevesa poteka na naslednji način. Najprej generiramo polni graf, kjer je vsaka črka povezana z vsemi ostalimi črkami v sliki, medsebojnim povezavam pa priredimo razdaljo $dist$ med pripadajočima črkama. S Kruskalovim algoritmom [6] na uteženem polnem grafu poiščemo minimalno vpeto drevo. Primer generiranega minimalnega vpetega drevesa je prikazan na sliki 2b.

2.3 Rezanje minimalnega vpetega drevesa

Kot lahko opazimo na sliki 2b, minimalno vpeto drevo ustrezno poveže sosedne črke. Kljub temu pa vsebuje tudi medvrstične povezave, ki jih je potrebno odstraniti. Z rezanjem teh povezav namreč dobimo poddrevesa, ki dejansko ustrezajo posameznim vrsticam.

Rezanja povezav se lotevamo na podoben način kot v [3]. Problem rezanja povezav prevedemo na problem minimizacije skupne energije vseh poddreves, tj. poddreves, ki jih dobimo po rezanju določenih povezav minimalnega vpetega drevesa. Z rezanjem različnih povezav dobimo različne kombinacije poddreves in pravilna kombinacija, tj. kombinacija, kjer poddrevesa ustrezajo dejanskim vrsticam, mora imeti minimalno energijo. Skupno energijo vseh poddreves označimo kot:

$$E = \sum_{i=1}^{N} W_l \cdot FL_i + \sum_{i=1}^{M} W_e \cdot FE_i \tag{2}$$

pri čemer je *N* število poddreves, *M* število odrezanih povezav, W_l vektor uteži za poddrevesa, FL_i vektor značilk *i*-tega poddrevesa, W_e vektor uteži odrezanih povezav ter FE_i vektor značilk *i*-te odrezane povezave.

Da bi koncept minimizacije energije resnično favoriziral razbitje na pravilna poddrevesa, uporabimo naslednje značilke, ki dobro povzemajo koncept vrstice – ta mora biti namreč ravna ter mora vsebovati bolj ali manj podobne črke. Vektor značilk poddrevesa sestavljajo:

- **Napaka linearne regresije** – za vsako poddrevo z linearno regresijo izračunamo premico, ki se najbolje prilega črkam poddrevesa ter izračunamo napako linearne regresije. Vrstice so v praksi ravne, kar pomeni, da morajo imeti nižjo napako, kot poljubno razpršene črke.
- **Višina vrstice** – višina vrstice ustreza razdalji med najbolj zgornjo in najbolj spodnjo črko v poddrevesu v ortogonalni smeri glede na smer premice dobljene z linearno regresijo. Vrednost normaliziramo z vertikalno razdaljo med zgornjo in spodnjo črko poddrevesa.
- **Število poddreves** – s številom poddreves omejimo pretirano razbitje na preveč poddreves.

Vektor značilk odrezanih povezav sestavljajo:

- **Razdalja med črkama odrezane povezave** – na podlagi metrike opisane v poglavju 2.2 izračunamo razdaljo med črkama, ki ju je odrezana povezava predhodno povezovala. Ta značilka favorizira rezanje povezav z večjimi razdaljami, saj le-te ustrezajo medvrstičnim povezavam.
- **Evklidska razdalja med črkama odrezane povezave** – izračuna se evklidska razdalja med središčema okvirjev predhodno povezanih črk.
- **Razdalja med okvirjema** – dodatno se izračuna tudi evklidska razdalja med okvirjema črk, ki ju je predhodno povezovala odrezana razdalja.

Število možnih razbitij poddrevesa je 2^P, pri čemer P ustreza številu povezav v drevesu. Pravilno razbitje na poddrevesa je le eno, to je tisto, ki ustreza dejanskim vrsticam. Ostalih $2^P - 1$ razbitij je nepravilnih. Ker visoka vrednost P povzroči kombinatorično eksplozijo, je učenje pravilnih uteži problematično. Vsak pravilen primer ima namreč $2^P - 1$ protiprimerov. Da bi se kompleksnosti problema izognili, uporabimo metodo učenja MCE (Minimum Classification Error) [7], kjer kot protiprimer uporabimo najboljšega izmed protiprimerov, tj. tistega, ki od pravilnega primera odstopa najmanj. Najboljši protiprimer dobimo z iskanjem v snopu (angl. beam search) [8], ki med preiskovanjem prostora stanj ohranja k najperspektivnejših rešitev.

V skladu z MCE izračun uteži poteka iterativno z metodo spusta [7], pri čemer se uteži v vsaki iteraciji ažurirajo po naslednji formuli:

$$W(t+1) = W(t) - \varepsilon(t) \cdot \xi \cdot l \cdot (1-l) \cdot (F_r - F_c)$$
$$l = \frac{1}{1+e^{-\xi \cdot d(r,c)}} \quad (3)$$
$$d(r,c) = E(r) - E(c)$$

pri čemer predstavlja c pravilno razbitje, r najboljši protiprimer, ε parameter hitrosti učenja, ξ stopnjo nelinearnosti sigmoidne funkcije, W, F in E pa ustrezajo vektorju uteži, vektorju značilk ter energiji iz enačbe 2.

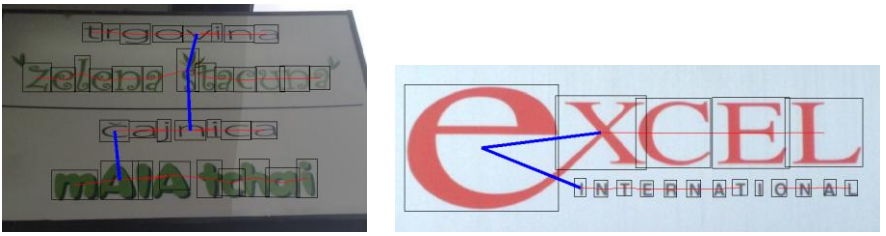
Na začetku poglavja smo omenili, da je problem razbitja minimalnega vpetega drevesa na poddrevesa optimizacijski problem, pri čemer iščemo takšno razbitje, ki minimizira skupno energijo vseh poddreves. Ko imamo izračunane ustrezne uteži, je razbijanje vhodnih dreves enostavno. Za iskanje optimalnega razbitja za razliko od [3] uporabljamo iskanje v snopu [8]. Iskanje v snopu teoretično ne najde vedno optimalne rešitve, vendar v primeru iskanja razbitja, ki minimizira skupno energijo poddreves, deluje povsem korektno.

3. REZULTATI

Za namene učenja uteži iz poglavij 2.2 in 2.3 smo uporabili prvih 40 slik iz kategorije *fog* v zbirki CVL OCR DB, metodo pa smo testirali na 148 slikah kategorije *normal*. Kategorija *normal* sicer vsebuje 158 slik, vendar smo jih 10 izpustili, ker so dvoumne in ima tudi človek probleme z interpretacijo vrstic v njih. Natančnost metode smo merili na dva načina: na nivoju celotne slike ter na nivoju vrstice. Na nivoju slik smo izmed vseh slik prešteli slike, katerih tekst je bil pravilno grupiran v vrstice, na nivoju vrstic pa smo izmed vseh vrstic, ki se pojavljajo v vseh slikah, prešteli pravilno določene vrstice. V primeru, ko je bila določena povezava nepravilno odrezana, smo pripadajočo vrstico šteli kot nepravilno detektirano. Prav tako, ko določena povezava ni bila odrezana, pa bi morala biti, smo pripadajočo vrstico šteli kot nepravilno detektirano. Rezultati metode so podani v tabeli 1.

Tabela 1: Rezultati grupiranja v vrstice na 148 slikah kategorije *normal* zbirke CVL OCR DB

	Slike	Vrstice
Število vseh primerov	148	313
Število pravilno detektiranih primerov	135	295
Natančnost grupiranja	91,22%	94,25%



(a) (b)

Slika 4: Rezultati grupiranja v vrstice – tanjše črte predstavljajo povezave minimalnega vpetega drevesa, debelejše črte pa odrezane povezave: (a) pravilno razbitje na vrstice in (b) nepravilno razbitje na vrstice

Slika 4a prikazuje primer pravilnega grupiranja na vrstice, medtem ko slika 4b prikazuje napačno razbitje. Črka »e« v besedi »eXCEL« je prevelika, hkrati pa krši pravilo linearne regresije, zato je odrezana od preostalega dela besede »XCEL«.

4. ZAKLJUČEK

Metoda grupiranja teksta na vrstice deluje zadovoljivo in je primerna za uporabo v praksi. Metoda se je izkazala za zelo uporabno prav v primeru slik naravnih scen, ki so v osnovi izredno problematične.

V nadaljevanju bomo metodo še izboljšali z vpeljavo dodatnih značilk, ki jih ponuja transformacija SWT [2]. Informacija o debelini loka črke je namreč zelo pomembna pri grupiranju teksta. Črke različnih debelin večinoma ne pripadajo istemu segmentu teksta (vrstica, beseda). Prav tako bomo v bližnji prihodnosti implementirali tudi razbitje vrstic na posamezne besede. Informacija o posameznih besedah v sliki je zelo pomembna, med drugim tudi v primeru evalvacije metod detekcije teksta na zbirki slik teksta v naravnih scenah ICDAR [9], saj je le-ta anotirana na nivoju besed.

ZAHVALA

Operacijo delno financira Evropska unija iz Evropskega socialnega sklada.

LITERATURA

1. A. Ikica, P. Peer (2011), CVL OCR DB, an annotated image database of text in natural scenes, and its usability, *Info. MIDEM*, let. 41, št. 2, str. 150-154.

2. B. Epshtein, E. Ofek, Y. Wexler (2010), Detecting text in natural scenes with stroke width transform, *CVPR 2010*, str. 2963-2970.

3. Y. F. Pan, X. Hou, C. L. Liu (2011), A hybrid approach to detect and localize texts in natural scene images, *IEEE Transactions on Image Processing*, let. 20, št. 3, str. 800-813.

4. S. M. Bishop (1995), *Neural Networks for Pattern Recognition*, Clarendon Press, Oxford.

5. F. Yi, C. L. Liu (2009), Handwritten Chinese text line segmentation by clustering with distance metric learning, *Pattern Recognition*, let. 42, št. 12, str. 3146-3157.

6. T. H. Cormen et al. (2001), *Introduction to Algorithms, Second Edition*, MIT Press and McGraw-Hill.

7. B. H. Juang, W. Chou, C. H. Lee (1997), Minimum classification error rate methods for speech recognition, *IEEE Transactions on Speech and Audio Processing*, let. 5, št. 3, str. 257-265.

8. W. Zhang (1999), *State-space search: Algorithms, complexity, extensions, and applications*, Springer, New York.

9. S. M. Lucas et al. (2003), ICDAR 2003 robust reading competitions, *ICDAR 2003*, str. 682-687.