SKANNER - KÄEKIRJA JA TRÜKITEKSTI ANALÜÜS

Kirjutas Isahiir
Monday, 14 November 2005

KÄEKIRJA JA TRÜKITEKSTI ANALÜÜS

Sissejuhatus

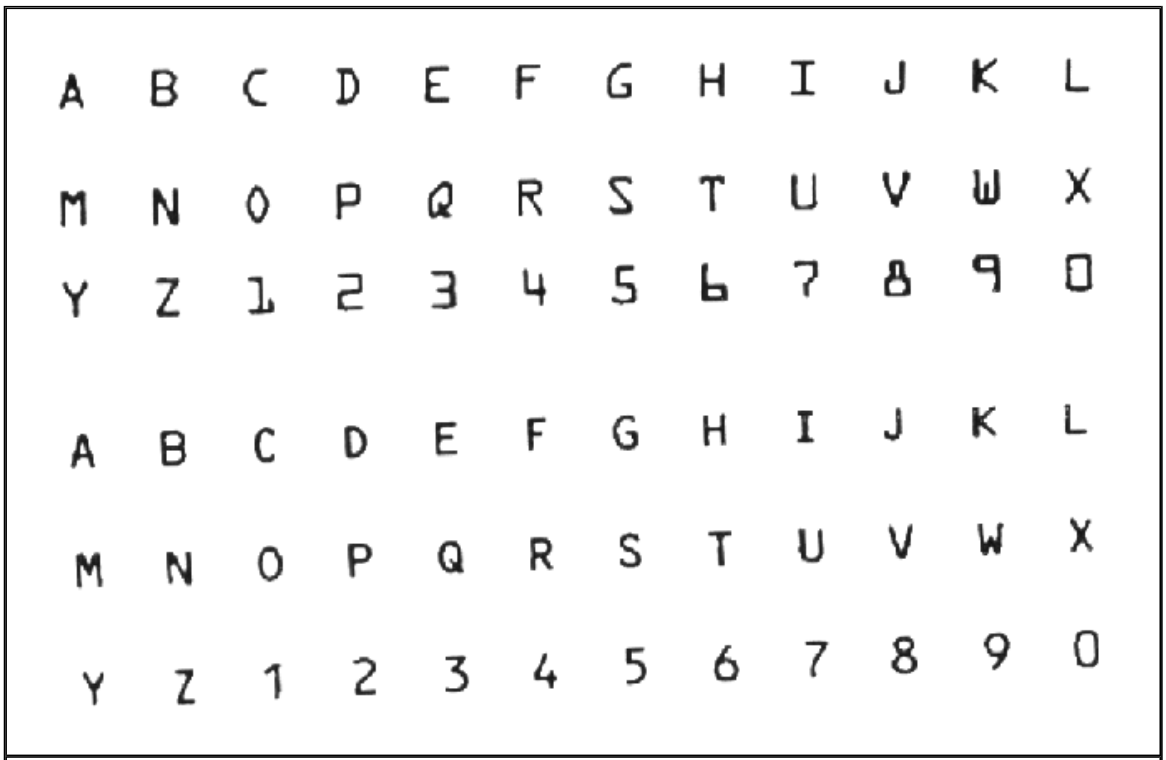
Üks keele põhivormidest on kirjakeel, mis esineb mitmel kujul: trükitekst (raamatud, ajalehed, ajakirjad), käsitsi kirjutatud tekst (kirjad, käsikirjad, tudengi konspektid, spikrid). Tekst võib esineda ka muudel kandjatel: kiri plangul, valgusreklaam, autonumber, vaguninumber. Seoses järjest kasvava vajadusega töödelda seda teksti arvutitega tekkib vajadus selle teisendamises elektronkujule. Vaatleme, millest koosneb kirjatekst, kuidas seda teisendada elektronkujule ja mõnede algoritmide põhimõtteid. Kirjateksti põhiomadused on järgmised:

1. Ta koosneb tasapinnalistest tehismärkidest;
2. Tema eesmärgiks on info edastamine;
3. Seda eesmärki saavutatakse (varem)kokkulepitud märkide ja keeleelementide vahelise vastavuse kaudu.

Vaatamata sellele, et kõnekeel on inimesele loomulikum, kui kirjakeel, ta ei säilu (ei oma mälu); väidetakse, et just kirjakeel andis võimaluse arendada inimkonna kultuuri ja teadust.

Erinevates keeltes eksisteerivad põhimõtteliselt erinevad üleskirjutamise viisid. Alfabeetilistes süsteemides (põhinäideteks on Ladina, Kreeka, Kirillitsa, Araabia, Devanagari jt.) sümbolid vastavad üksikutele häälikutele või silpidele, millest koostatakse keele suuremaid ühikuid. Paljud keeled võivad kasutada sama alfabeeti, mõned aga kasutavad omi (Gruusia, Armeenia, Kreeka, Heebrea jne). Ideograafilistes süsteemides (nt. Hiina) sümbolid tähendavad tervikut mõistet ja ei ole seotud häälduse ga. Mõned keeled kasutavad mõlemat süsteemi (nt. Jaapani keel). Praegu kasutatakse ligi 25 põhimõtteliselt erinevat alfabeeti (kui arvestada, et nt. Eesti ja Poola keeled kasutavad sama Ladina alfabeeti). Peale selle, igas alfabeedis on oma reeglid, kuidas käsitsi kirjutatud tähti ühendatakse sõnadeks. Peale tavalisi tähti igas alfabeedis on ettenähtud numbrite üleskirjutamise viis või sümbolid (Araabia-India numbrid).

Inimtegevuste teostamine seadmete abil on iidne unelm. Esimeseks eksperimendiks sellel alal võib lugeda 1870-L aastal Bostonis C.R.Carey poolt läbi viidud katset isetehtud maatriksskanneriga (fotoelementidel). 20 aastat hiljem poolakas P.Nipkow leiutas järjestikskanneri, mis sai alguseks tänapäeva telerite laotuse ideele. Kuid tõsised väljatöötlused sellel alal said tekkida alles peale elektronarvutite leiutamist 1940tel aastatel. Esimeseks tõeliseks teksti äratundjaks loetakse 1954 aastal loodud seadet, mis võimaldas kirjutusmasinal trükitud pangaarveid perforeerida kaartidele, et neid sisestada arvutisse.



Pilt-1

60-ndatel aastatel USA-s oli loodud eristandart tähtede ja numbrite kirjutamiseks, mis sai nimeks OCR-A [PILT-1] ja mis oli ettenähtud inimestele, et teha ennast arusaadavaks masinatele. Mõne aasta pärast Euroopas oli loodud sarnane standart OCR-B, mis oli küll ilusam , kuid probleem jäi ikka püsti, kuna see nõudis inimestelt kirjutamist nendele harjumata viisil (kõik vist mäletavad kirjaümbrike indeksite kirjutamise trafarettidega). Kohe tekkisid süsteemid, mis tuvastasid mõlemat fonti, sellele hiljem lisandus trükimasina font ja ka paljud teised, kuid selle printsiibi pidev kasutamine osutus võimatuks. Tekkis vajadus teaduslikus lähenemises, mis võimaldaks võtta maha piiranguid.

Tänaseks päevaks on loodud palju kaasaegseid tekstituvastamise süsteeme väga erinevate alade jaoks (teksti pöördformateerimissüsteemid, tekstituvastamise APId ja SDKd, süsteemid teksti lugemiseks pimedate jaoks ja rida teisi).

Dokumendi laotuse analüüs (DLA)

Teksti äratundmise protsessiga on vahekorras dokumendi laotuse analüüsi protsess. Selle juures on raske öelda, kumb neist on primaarne: DLA ülesandeks on dokumendi sisu füüsilise (ruumilise) ja loogilise struktuuri määramine. See struktuur võib varieeruda laiades piirides (ajalehed, ajakirjad, raamatud, kirjad, jne). Näiteks, ajalehe puhul kogu väli jagatakse üksikuteks ristküülikuteks, nagu: Pealkiri, Pilt, Tekst, Pilt, Teksti jätkamine, jne. Esimesel pilgul võib tunduda, et DLA on tekstianalüüsi metatasemeks, kuid võib esineda ka vastupidist, näiteks, kui mingi tekstilõik lõpeb sõnadega "järgneb leheküljel nr.5", siis hästi projekteeritud süsteemis tekstianalüsaator hakkab juhtima DLA protsessi. DLA põhilisteks kasutusaladeks on: tekstidokumendid, täidetavad vormid, postiaadressid ja joonised. Vaatleme neid põhjalikumalt:

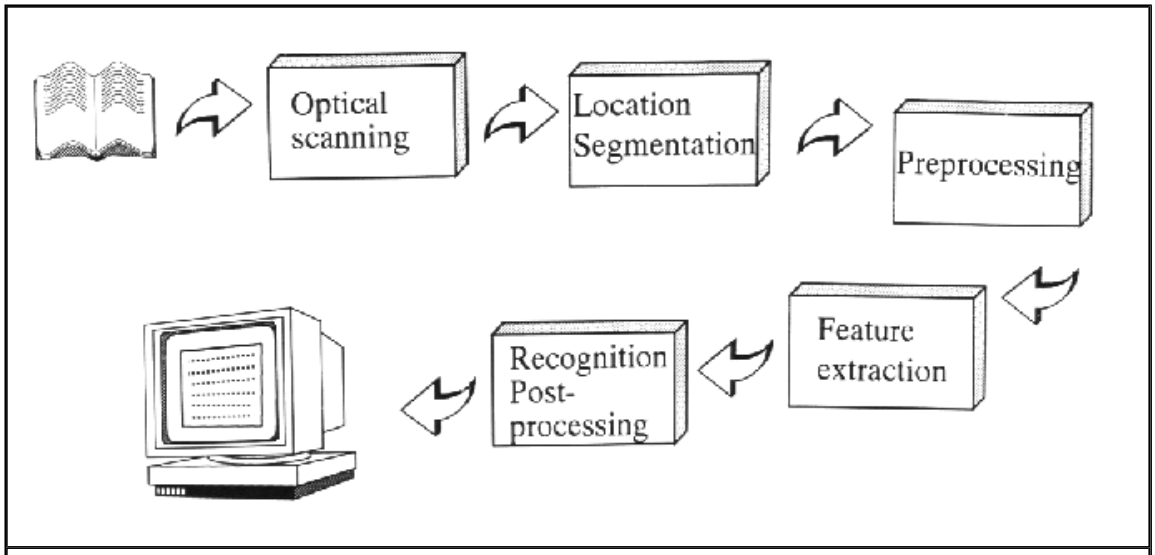
Teksti DLA lõplikuks eesmärgiks on Gutenbergi tsükli täitmine ehk pöördformateerimise ülesanne, mis tähendab, et skaneeritud dokument teisendatakse mingi dokumendi kirjelduse formaati (nt. .DOC või .html), millest teda võib edasi töödelda (ja/või välja trükkida). Olemasolevad DLA süsteemid võimaldavad piiratud pöördformateerimist, äratundes ja teisendades selliseid lihtsamaid struktuure nagu pealkirjad, eraldusjooned, tekstilõigud, veerud ja tabelid. Nad on samuti võimelised ära tundma, kuid mitte teisendama halli värvi toone ja jooniseid (93manda aasta seisuga). Mõningatel juhtudel ei vajatagi dokumendistruktuuri totaalset äratundmist. Näiteks, võrgu kaudu artiklite saatmise automaatsüsteemile vajaliku artikli leidmiseks piisab ajakirja nimest, aastakäigust, numbrist, artikli pealkirjast, autori(te)st ja võtmesõnadest. Artikli tekst aga hoitakse ja saadetakse tellijale maatrikskujul (pöördformateerimist ei tehta).

Täidetavad vormid - on väga tihedalt seotud andmebaasidega - tüüpiline täidetav vorm koosneb n paarist kujul: (andme nimetus, andme väärtus). Tavaliselt täidetava vormi lahterdus on ette teada, seega on teada iga lahtri kohta selle paigutus ja tüüp, millest genereeritakse piirangud. Seda kõike juba kasutab äratundmise algoritm. On võimalus ka analüüsida täidetavaid vorme, mille lahtrite paigutus ja tüüp pole ette teada; see protseduur on loomulikult keerulisem ja vähem usaldatav, kuid samuti teostatav.
Posti Aadressid. Siin on tegemist hästi määratud loogilise formaadiga, kuid väga vaba füüsilise paigutusega. Andmete suur liiasus võimaldab luua ranged piirangud andmetele. USA's lähemas tulevikus planeeritakse võtta kasutusele sorteerimismasinaid, mis võimaldaksid planeerida üksikute postikandjate mar? ruute.
Joonised. Suurem osa tegevusest sellel alal on pühendatud suurte jooniste sisestamisele CAD/CAM süsteemidesse. Arvatakse, et suurte integraalskeemide sisestamine arvutisse joonise skaneerimise teel on odavam, kui selle skeemi käsitsi sisestamine, kuna ekraanid on väikesed ja puudub ülevaade kogu skeemi kohta. On olemas tööstuslik programm, mis lahendab seda ülesannet. Teiseks suunaks jooniste sisestamisel on ruumilise kujundi kirjelduse genereerimine kolme skaneeritud projektsiooni järgi. Häid tulemusi on saavutatud ka topograafiliste kaartide sisestamisel.

Sümbolite tuvastamise protsess

Ülesanne seisneb graafiliste märkide ruumilise kuju teisendamises sellele vastavateks sümboliteks. Kuna me lahendamiseks kasutame arvutit, siis sümboliks on meil indeks mingis teadaolevas sümbolite tabelis (nt. ASCII, EBCDIC, WideChar või UNICODE).

Sümbolite tuvastamise protsess koosneb järgmistest etappidest:



Pilt-2

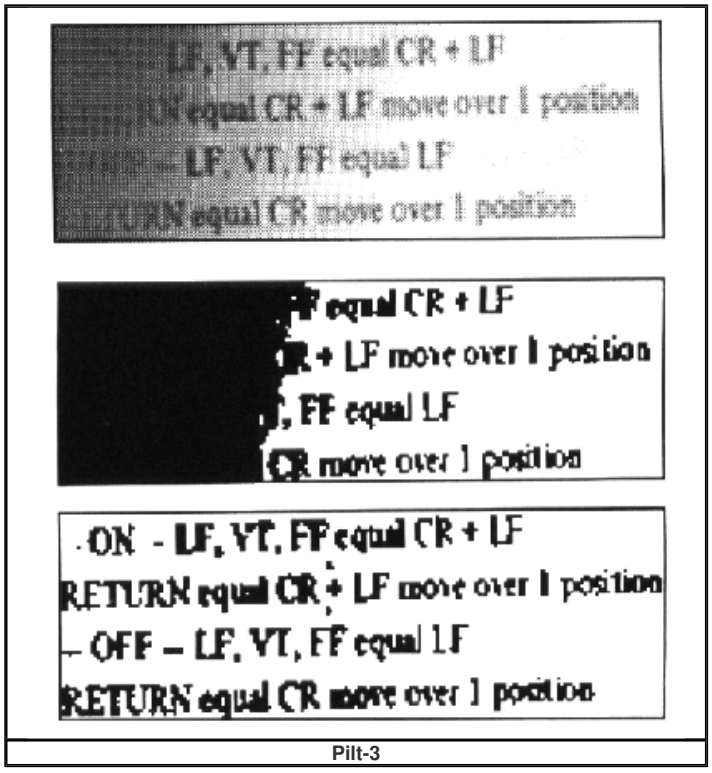
Teisendamine elektroonkujusse

Enamus dokumente kujutavad endast kontrastset tumedat trükiteksti heledal taustal. Selleks, et seda teksti arvutisse sisestada on vaja analoogpildi teisendamist digitaalkujju. Selleks kasutatakse mitmeid mooduseid, tavaliselt maatriks-skaneerimist. Näiteks, A4 lehakülje skaneerimisel (tihedusega 300punkti tollile) saame 8.4Mb pooltoonilise maatriksi, mis tavaliselt on juba sobiv edasiseks töötlemiseks. Eraldamisvõime/tihedus (dpi) sõltub minimaalse teksti suurusest, mis vajab kindlat äratundmist ja maksimaalsest edastamise kiirusest, mis on piiratud kanali võimalustega. Näiteks, standardne faksi pilt skaneeritakse tihedusega 200punkti tollile piki skaneerimisjoont ja 100punkti tollile lehe liikumise suunas. Tekst võib esineda samuti ka käsikirja kujul, kas paberil või elektroonsel tundlikul pinnal. Eristatakse kahte lähenemist: valmisteksti hilisem töötlemine (off_line) või teksti töötlemine kirjutamise ajal (on_line). Off-line juht on kõige tavalisem - meil on paberil tekst (kas trükitud või käsitsi kirjutatud) ja on vaja seda arvutisse viia. Selleks kasutame maatriksskanneri ja saame suurepärase maatriksi (peaaegu) suvalise tihedusega, kuid see on kahjuks ka kogu informatsioon. Palju parem on on_line juhtum, kus meil on kättesaadav kogu kirjutamise protsess kvanteeritud ajafunktsioonina (pärast näeme, mis sellest kasu on). Selle sisestamiseks on väga palju variante: videokaameraga jälgimisest (kõige kallim) tundlikul pinnal kirjutamiseni (paljud kaasaegsed NewTon'id kasutavad tundliku ekraani oma ainsa interfeisina). Lisaks sellele, et on_line juht on informatiivsem, tema väljund on isegi kompaktsem: näiteks, ühe käsitsi kirjutatud sõna hoidmiseks on_line juhul vajame 230 baiti (ajakvanteerimise tihedusega 100 korda sekundis) ja sama sõna korral off_line juhul peame hoidma 80 Kbaiti (skaneerimisetihedusega 300 punkti tollile). Peale selle, tuvastamise kvaliteet (%des) on on_line juhul tunduvalt parem, kui off_line juhul.

Kahenivooline kvanteerimine

Off-line juhul skaneerimine on tavaliselt kas värviline või mitmetooniline must-valge. Kuid edasine protsess vajab ranget tähe ja tausta eristamist (seega ta nõuab kahte värvi). Seda saavutatakse kahenivoolise kvanteerimise teel. Kõige lihtsam moodus oleks määrata piirnivoo, millest heledamad värvid teisenduksid valgeks ja tumedamad - mustaks värviks (või vastupidi). Kuid see on tavaliselt

vastuvõtmata, sest võib tekkida olukordi, nagu on näidatud pildil [PILT-3]. Selle pärast praktikas kasutatakse nn. adaptiivset kvanteerimist, mis võimaldab paremini eristada tähti taustast.

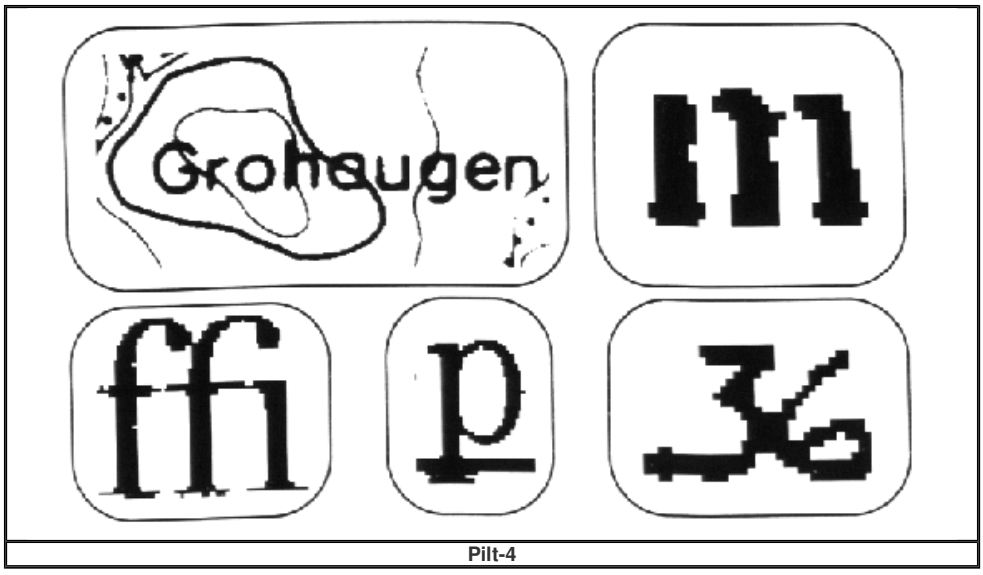


Pilt-3

Piirkondadeks segmenteerimine ja üksikute sümbolite "haaramine"

Segmenteerimine on protsess, mille käigus määratakse pildi üksikuid koostisosi. On tähtis eraldada trükiteksti piltidest, või isegi tähtistekst ebatähtsast, näiteks kirjaümbrikute sorteerimisel on tähtis eraldada aadressi muust tekstist: templi peal olev tekst, postmargil olev tekst, ümbriku pildi all olev tekst, jne. Teksti segmenteerimise puhul tuleb eraldada üksikuid sümboleid või terveid sõnu. Tavaliselt tükeldatakse veelgi peenemaks, jagades kogu pildi seotud piirkondadeks - seda on kerge realiseerida, kuid tekivad järgmised 4 probleemi [PILT-4]:

1. kokkusulanud ja katkenud sümbolite eraldamine. Tumedate koopiate puhul tähed kippuvad sulama kokku, ja vastupidi - liiga heleda teksti puhul tähed kippuvad katkema osadeks, mida arvuti võib pidada üksikuteks sümboliteks.
2. müra eraldamine tekstist. Punktid ja rõhumärgid tähtede kohal tihti ajatakse segamini müraga ja vastupidi.
3. pildi segamini ajamine tekstiga.
4. teksti segamini ajamine pildiga.

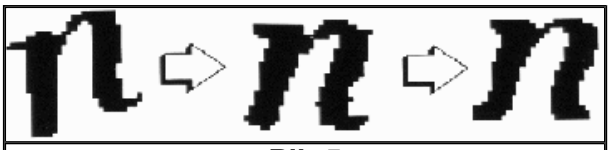


Pilt-4

Eeltöötlemine

Sõltuvalt skanneri eraldusvõimest ja binaarse kvanteerimise õnnestumist, on peale skaneerimist pildil tavaliselt mingis koguses müra, mida võib elimineerida enne tuvastamise protsessi algust. Silumine tähendab nii väikeste pilude täitmist, kui ka joonte peenemaks tegemist.

Peale silumist tavaliselt tehakse ka normaliseerimist [PILT-5].



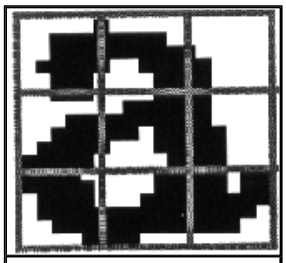
Pilt-5

Normaliseerimine on võrdsele suurusele, kalletele ja pööretele taandamine. Juhul kui meil on pööratud teksti lõik või isegi rida, on võimalik leida pöördenurk kasutades *Hough* teisendust ja pöörata ta tagasi. Kui aga nurga all on üksik täht, siis see on võimatu.

Omaduste eraldamine/väljatoomine

Kõige keerulisem probleem kogu Dokumendi Analüüsi protsessis on üksiku eraldatud ja eeltöödeldud sümboli tuvastamine. Tavaliselt selleks kasutatakse omaduste vektori mõistet. Omaduste vektor on arvude vektor, mis koosneb eelnevalt valitud omaduste väärtustest. Süsteem hoiab kõikide teadaolevate sümbolite vektoreid ja iga jooksva sümboli puhul võrdleb neid tema omaduste vektoriga. Tulemuseks valitakse sümbol, mis on lähim mingi teatava normi järgi. Vaatleme esiteks, milliseid omadusi selleks kasutatakse:

Otsene maatriksite võrdlemine. See meetod erineb teistest selle poolest, et siin mingeid omadusi ei eraldatagi. Selle asemel määratleva sümboli maatriks võrreldakse kõikide süsteemile teadaolevate sümbolite maatriksitega ja valitakse lähim. Seda on väga kerge realiseerida rauas, on kiire, kuid on häirete suhtes väga hell. Tegelikud omadused, mis põhinevad punktide statistilisel jaotusel: nad on tavaliselt tuimad moonutuste ja fontide varieerimiste suhtes.



Pilt-6

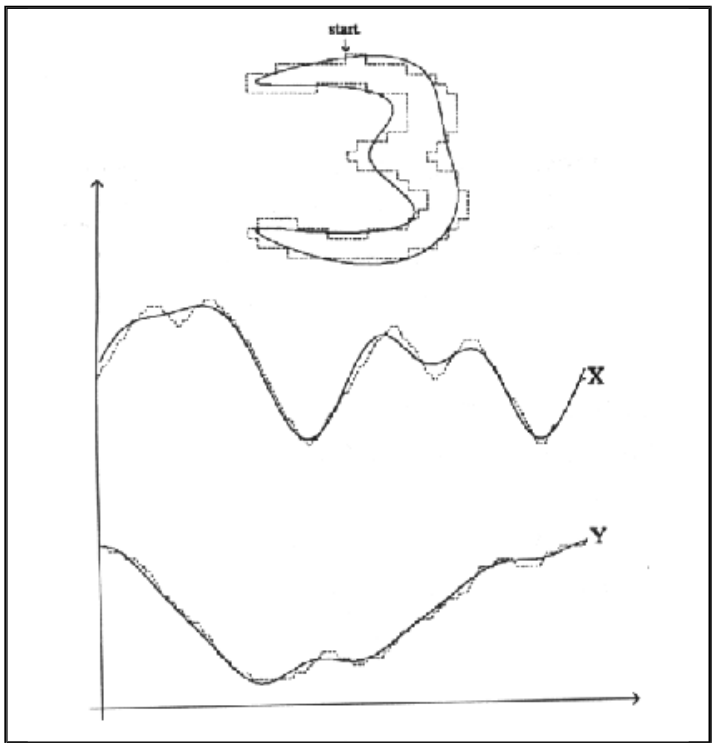
-Piirkondadeks tükeldamine: sümboli maatriks tükeldatakse kattuvateks või mittekattuvateks piirkondadeks ja mustade punktide tihedused piirkonniti võetakse omadusteks.

-**Statistilised momendid**: valitud punkti, telje või koordinaatide süsteemi suhtes võetud mustade punktide statistilised momendid võetakse omadusteks.

-**Lõikumised ja kaugused**: maatriksil valitakse mingi vektor ja arvutatakse sümboli ja selle vektori lõikumiste arvu. Seda omadust kasutatakse tihti, kuna ta on kiire ja lihtne. Kaugusteks võetakse nende lõikumiste vahelised kaugused.

-**Ahelad** on maatriksite võrdlemise erijuht, kus võrdlust teostatakse piki kindla kujuga joont maatriksis.

-**Karakteristlikud suunad** on lõikumiste omaduse erijuht, kus iga tausta punkti jaoks võetakse vertikaalne ja horisontaalne vektor ja omaduseks võetakse nende lõikumiste arv sümboli joontega.



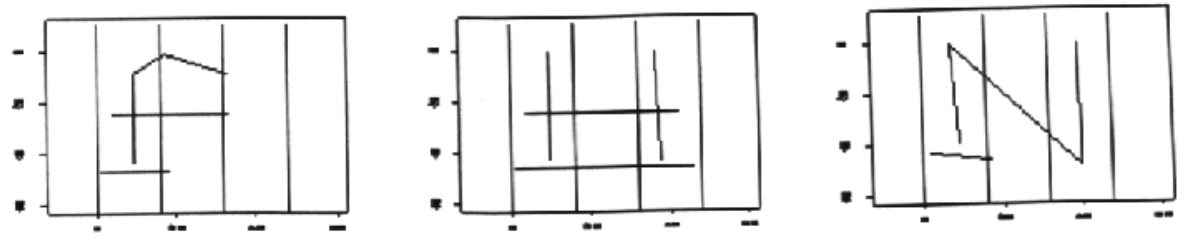
Pilt-7

Teisendused ja Interpolatsioon:

Need meetodid võimaldavad alandada omaduste vektori mõõtu ja teha neid omadusi invariantseteks globaalsete teisenduste (rööplükete ja pöörete) suhtes. Kasutatakse Fourier, Haar'i ja teisi teisendusi. Paljud neist baseeruvad kõveral, mis kirjeldab sümboli kontuuri [PILT-7]. Seega need omadused on väga tundlikud müra suhtes, mis moonutab seda kontuuri.

Struktuurne analüüs

See on kõige keerulisem, kuid ka kõige kindlam meetod. Siin eraldatakse omadusi, mis määravad sümboli geomeetrilise ja topoloogilise struktuuri. Siin analüüsi teostatakse kõrgemal tasemel, tekivad kõrgema taseme mõisted, nagu kaared, lõigud, lõikumised, seriffid [PILT-8].Võrreldes teiste meetoditega struktuurne analüüs on palju tolerantsem mürade ja erinevate fontide suhtes. Kuid ta on väga tundlik pöörete suhtes.



Pilt-8

Omaduste eraldamise meetodite head ja vead on toodud tabelis [TABEL-1], kus neid võrreldakse erinevate häirekindlususe ja praktilisuse kriteeriumite järgi.

Feature extraction technique	Robustness 1 2 3 4 5	Practical use 1 2 3
Template matching	● ● ○ ○ ○	○ ● ○
Transformations	○ ● ● ● ●	○ ○ ●
Distribution of points: Zoning	○ ● ○ ○ ●	● ● ○
Moments	● ● ○ ● ●	○ ● ○
n-tuple	● ○ ● ○ ●	● ● ●
Characteristic loci	○ ● ● ● ●	● ● ○
Crossings	○ ● ● ● ●	● ● ○
Structural features	○ ● ● ● ●	● ○ ●

● High or easy ● Medium ○ Low or difficult

Tabel-1

Klassifitseerimine: see on sümboli tuvastamise viimane etapp, mille tulemuseks on konkreetne sümbol. Selleks kasutatakse eelmisel etapil tuletatud omaduste vektorit, etalonide omaduste vektoreid ja mingit võrdlemisalgoritmi, nt. minimaalse normi/kauguse meetod, Tehisnärvivõrgud ja Varjatud Markovi mudelid.

Kasutatud materjalid:

www.cs.ut.ee/~roosmaa/KT98.html

KOMMENTAARID

Viimati uuendatud (Thursday, 24 November 2005)