Strojové čtení písma na počítačích třídy PC

Martin F i n k e

Motto: Pět fází vývoje projektu:
1. předprojektové nadšení
2. poprojektové vystřízlivění
3. hledání viníků
4. potrestání nevinných
5. odměnění nezúčastněných [1]

Zajisté znáte tuto anekdotu, která svého času s velkým úspěchem kolovala mezi odbornou veřejností. Existují i varianty s více fázemi. To ale není podstatné, protože pro nás budou zajímavé pouze první dvě, třebaže pointa je skryta v těch zbývajících. Časy se však změnily a málokterý odborník by byl (alespoň by neměl být) dnes ochoten připustit, že jeho práce má zrovna takový průběh. Už proto, že přece jenom dnes už záleží i na něm, jestli něco takového dovolí. Slyším výkřiky nesouhlasu a proto prosím o klid. O posledních třech krocích nehodlám polemizovat, jenom jsem lehce nadhodil svou představu o dalším vývoji v naší společnosti. Teď chci spíše ukázat na první dvě fáze. Všimněte si, že jejich platnost nepomine ani tehdy, když se vše vrátí do normálních kolejí a práce odborníků nebude zhodnocována pohledy politickými, ideovými či jinými nepatřičnými. A že neplatí jen při vývoji nějakého projektu, ale všude, kde nejprve o něčem nevíme nic, pak něco málo a nakonec něco. Jako v té pohádce, kde princ Jasoň zvolá: "Znám ji zatím pouze z vyprávění, a už hořím. Co teprve až ji spatřím...". A když ji pak opravdu spatří: "... vyprávěli mi, že má zlaté vlasy ... atd."[2]. V této pohádce vyvolená zavržena nebyla, třebaže příliš nesplňovala počáteční představu a tak se nakonec po přičinění všech zúčastněných vše v dobré obrátilo. Bohužel v běžném životě máme spíše sklon po zklamání (často i bezdůvodném) zatracovat do pátého kolena. A tak když se začínáme s něčím seznamovat, projdeme obvykle dvěma silnými emociálními stavy, kterými škodíme nejen sami sobě, ale obvykle i věci samé. Určitě vás něco podobného potkalo i při seznamování s novými programovými produkty. Oslavný článek, příznivý ohlas u přítele, letmé zahlédnutí obrazovky počítače se spuštěným programem na některé výstavě a už člověk hlásá: ten a žádný jiný. A pak je vám program dostane do rukou (opomineme raději způsob jakou cestou), a stačí několik nepodařených pokusů o oživení (bez příručky to obvykle jde špatně) a už je ochoten vzývat ďábla. V takovém okamžiku, zvláště jste-li pro své okolí určitou autoritou, lze do světa vypustit velice nepříznivé míně-

ní, které by člověk později vzal i rád zpátky. Jenomže džin je už z láhve venku a nebohý producent programu aby zachraňoval co se dá. Nakonec se stejně všeobecné mínění o programu ustálí přesně na té úrovni, jakou si zaslouží. Do té doby je však třeba přistupovat k jakékoliv extrémní informaci s určitou opatrností. Jenomže z čeho si má nepoučený potenciální uživatel vlastně vybrat, pokud nemá zájem zkoušet na vlastní kůži a za vlastní peníze, kudy cesta nevede.

A proto opusťme jalové teoretizování a věnujme se konkrétnímu problému: strojovému čtení tištěných textů (v zahraničí označované jako OCR - Optical Character Recognition). Pro většinu uživatelů počítačů třídy PC jde o horkou novinku, která u nás právě prochází fází "nadšení" a "vystřízlivění". Přesně tak, jak o tom hovoří úvod této přednášky. Jde však o novinku značně relativní. Jednak OCR je problém řešený odpradávna, zpočátku dokonce pomocí specializovaných automatů, a jednak na počítačích třídy PC se první programy objevily už v polovině osmdesátých let. Hlavní vlna nových produktů ale přišla teprve v posledních letech, kdy se ve velké míře rozšířila nová počítačová periferie - snímač obrazu neboli scanner. Scanner převádí snímanou předlohu do číselné formy, často ve formátech používaných obecně pro záznam grafických obrazů v počítači. Jeho využití pro čtení tištěných textů je funkce pouze odvozená, třebaže řada uživatelů dnes mylně zaměňuje scanner se zařízením určeným ke čtení textu. Proto hned na začátku zdůrazňuji: scanner pouze snímá libovolnou předlohu a získaný obraz předlohy v číselné podobě může být použit jako vstup programu, který dokáže rozpoznávat obrazy jednotlivých písmen a z nich vytvořit klasický textový soubor, v němž jsou znaky kódovány podle některého obvyklého kódu (např. ASCII). Jinak je možné sejmutý obraz využívat i jiným způsobem. To je ale již mimo rámec tohoto příspěvku.

Je zvláštní, že v představě mnoha uživatelů počítačů PC je problém strojového čtení písma problémem vyřešeným. Silně to podporují i samotní výrobci scannerů tím, že ve svých reklamních materiálech se o programech OCR zmiňují velice "cudně", asi tak ve smyslu: OCR program je samozřejmě možno přikoupit. Konec, tečka. Jenomže OCR není a ještě hned tak brzo nebude "vyřešený problém". Konečně, proč by měl být? Vždyť dnes se málokdo domnívá, že ve zpracování hromadných dat či textů bylo řečeno poslední slovo. A přesto představa mnohých zákazníků o OCR je neuvěřitelně naivní: vložím předlohu do scanneru, chvíli počkám a převedený text je v souboru. Když pak zjistí, že to není vždy tak jednoduché, jsou zklamáni. A přesto může OCR usnadnit mnoho práce, stačí být na nedostatky předem připraven.

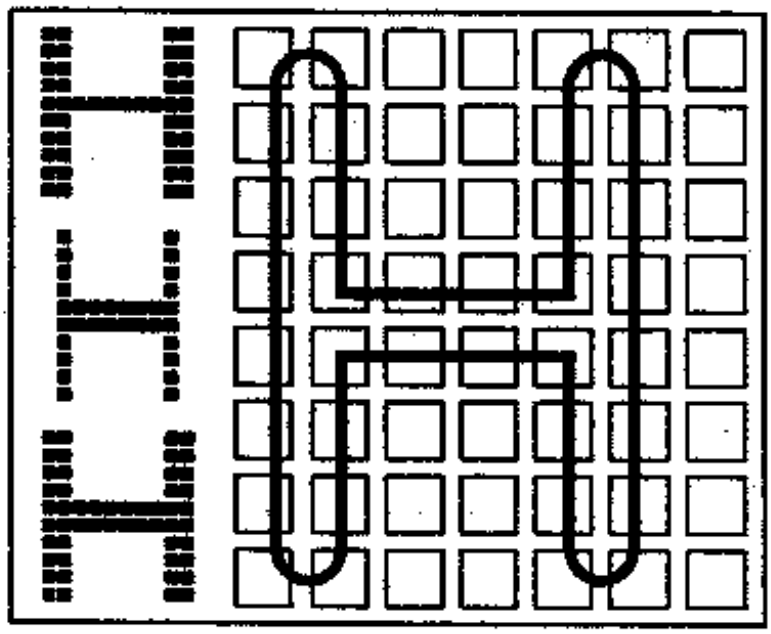
Rozpoznávání písma je součást vědního oboru rozpoznávání obrazu, který spolu s dalšími obory tvoří vědní disciplínu označovanou obvykle jako umělá inteligence. Rozpoznávání písma představuje samostatný oddíl už proto, že přichází s určitým zjednodušením vstupních podmínek (pracuje výhradně s dvourozměrným obrazem,

rozlišuje pouze dva stavy obrazových bodů apod.) Navíc se zde nebudeme zabývat komplexnějším zpracováním textového dokumentu[3] (do čehož připadá kromě čtení písma také zpracování obrázků, tabulek, vyhledávání jednotlivých částí textu apod.).

Rozpoznávání[4] je vlastně vyhledání určitého objektu v obraze a jeho zařazení do určité třídy objektů tak, jak to konkrétně naše aplikace vyžaduje. Jinými slovy zařazením objektu do třídy jsme tento rozpoznali. V rozpoznávání písma je objektem chápán konkrétní znak vytištěný na papíře, třídou pak množina objektů, kterým odpovídá určitý kód znaku. To znamená, že např. třída písmene 'a' (kód 61H v ASCII) by měla být tvořena všemi obrazy objektů, které toto písmeno představují. Jsou to jednak všechna malá písmena 'a' různých typů písma (Helvetica, Courier atd.), různé tloušťky, sklonu, velikosti apod., ale také všechny písmena s různými odchylkami, deformacemi a poškozeními. Třídu se lze představit jako množinu objektů splňujících určité podmínky, přičemž by bylo žádoucí, aby vzájemným průnikem všech množin jednotlivých tříd byla množina prázdná. A tady přichází první kámen úrazu - průnikem těchto množin množina prázdná není. Ale proč? Přece člověk, který čte určitý text průměrné kvality, nemá obvykle velké potíže jednotlivé písmena správně určovat. Problém však vyplývá z možností současné výpočetní techniky. Vzhledem k její (pro řešení dané problematiky) omezené rychlosti a paměťové kapacity je nutné zjednodušení popisu, který definuje jednotlivé prvky množiny objektů dané třídy. Tento popis je kompromisem mezi požadavkem obsáhnout co nejvíce možných variant určitého znaku a přitom do sebe zahrnovat i určité mezní varianty jiných, podobných znaků (např. písmena c a e, nebo l a I atd.). Zde přichází další potíž. Vzhledem k variabilitě písma nelze zcela definovat jednotlivé třídy předem a je žádoucí, aby samotný uživatel měl možnost jejich dodefinování neboli doučení. Zde je již velmi těžké zajistit, aby se množiny objektů jednotlivých tříd vzájemně neprostupovaly, neboť na jejich definici se podílí samotný uživatel, který nemá zkušenost ani potřebnou trpělivost vybírat reprezentativní znaky. Málokdo si totiž uvědomuje, že ani u kvalitních tisků nejsou jednotlivá písmena stejného vzhledu. Pokud bychom měli možnost písmena zvětšit a navzájem překrýt, překvapila by nás velká variabilita tvaru, odchylek a deformací. Kdybyste vzali např. větší množství obrazů písmen 'e' z nějakého textu a všechny je položili na sebe, dostanete jako výsledný obraz neurčitý chuchvalec, do kterého by se postupně vešla i další malá písmena jako c,o,a,s atd.

K dalším deformacím dochází při snímání tištěné předlohy, kdy je pořizován její obraz. Zde dochází k deformacím dvojího druhu: první případ se vztahuje k ručním scannerům, kdy je obraz deformován při prudších pohybech nebo při vychýlení scanneru ze správného směru snímání. U stolních scannerů tento problém odpadá. V druhém případě jde o deformace dané vlastní formou reprezentace obrazu, do

něhož je předloha převáděna. Jde o to, že obraz není spojitý, ale kvantován do jednotlivých obrazových bodů uspořádaných do dvourozměrné matice. Scanner si rozdělí předlohu do jakési sítě a jednotlivým políčkům podle zabarvení přiřadí určitou číselnou hodnotu (obvykle 0 nebo 1). Avšak vytištěná písmena předlohy mohou být vůči této síti různě posunutá, což vede k následujícím jevům:



To, o čem byla doposud řeč, představovalo z hlediska programů OCR jakýsi ideální stav: kvalita tisku zaručuje dobrou reprodukovatelnost jednotlivých písmen. Co ale má dělat takový ubohý program, jehož tvůrce nikdy ve svém životě nepřišel do styku s tiskem naší, tuzemské kvality?

Nechme tuto otázku otevřenou a pokračujme ve výkladu. Nyní máme obraz tištěné předlohy v počítači v podobě binární mapy, t.j. matice bitů odpovídajících jednotlivým bodům kvantovaného obrazu. Vzhledem k nárokům na paměť (obraz předlohy formátu A4 zabírá několik stovek kB) může dojít pro tuzemského zájemce k nepříjemnému omezení - řada programů OCR požaduje alespoň 2MB operační paměti. A ta u nás není příliš rozšířená.

Začněme nyní s vlastním převodem textu. Nejprve je třeba v obraze oddělit jednotlivé objekty (písmena) pro zatřídění. A znova se nám vrací kvalita tisku. Jestliže se jednotlivé znaky na předloze navzájem dotýkají, je často problém je správně rozdělit. Menší komplikace přináší tzv. neproporcionální písmo, kdy je známa konstantní šířka jednotlivých písmen, takže program ví, ve kterém místě by se zhruba měly znaky stýkat. Potíž nastává u písma proporcionálního, u kterého šířka nejširších písmen (jako je m,M,W apod.) může být i několikanásobkem šířky písmen nejužších (jako

jsou i,j,l atd.). Pokud se tedy u proporcionálního písma některé znaky spojí, je obtížné je od sebe oddělit. Řada programů OCR vychází při řešení takovýchto situací z následující úvahy: ke spojení písmen sice dochází, ale není to běžný jev (to bohužel neplatí u nás). Pokud se náhodou znaky spojí, je pravděpodobné, že jich nebude více než dva, přičemž jde často o stejné dvojice. Situace se řeší tím, že pro objekt představovaný dvojicí spojených znaků se zavede zvláštní třída. Kdykoliv se podaří nějaký objekt zařadit do takové třídy, budou do výstupního textového souboru vloženy dva znaky. Tento přístup v našich podmínkách přináší spoustu problémů: jednak dochází ke spojování více znaků než dvou, a jednak vzniká velké množství dvojic, což přináší řadu nepříznivých účinků (zpomalení, zvýšené nároky na paměť apod.). Existuje i další přístup k problematice, a to snaha spojené znaky od sebe oddělit. V takovém případě se postupuje při separaci objektu v součinnosti s jeho rozpoznáváním. Nedojde-li k zařazení objektu do žádné třídy, oddělí se část objektu (obvykle zprava) a zmenšený objekt se znovu zatřiďuje. To se pak opakuje do té doby, než dojde zařazení části objektu do určité třídy, nebo je splněna jiná, omezující podmínka. V prvém případě to znamená, že se podařilo oddělit první ze spojených znaků, v druhém případě pak bude celý objekt označen jako nerozpoznaný.

Na chybném rozpoznání se podílejí i vlastní rozpoznávací metody. Nebudeme se věnovat jejich popisu, řekneme si jenom, že jsou rozděleny do dvou skupin na metody příznakové a strukturální[4][5]. V praxi je vhodné obě metody kombinovat, neboť v případě strukturálních metod je nižší citlivost na různé deformace, zatímco prostřednictvím příznakových metod se můžeme dozvědět o konkrétních znacích i další doplňkové informace, jako např. o jaký typ písma se jedná. Bohužel použití strukturálních metod pro rozpoznávání písma je na počítačích PC nepříznivé z hlediska rychlosti a prakticky se nepoužívá.

Chtěl bych upozornit na ještě jeden problém specifický pro naše národní abecedy. Jsou jím diakritická znaménka. Ve slovenštině a češtině je spousta znaků, které se od sebe liší pouze diakritickými znaménky (mám na mysli znaky jako á a ä nebo é a ě apod.). Vlastní znaménko na těchto znacích tvoří jen malou část a tak v důsledku určité dovolené tolerance rozpoznávací metody dochází velice snadno k záměně takovýchto písmen. Řešením může být oddělené zpracování samotného diakritického znaménka.

Podařilo se naznačit jen ty nejvážnější problémy, se kterými se tvůrci programů pro strojové čtení písma musí potýkat a díky kterým nelze ještě dlouho tuto problematiku považovat za vyřešenou. Největší problém spočívá totiž v tom, že dílčí problémy, které byly popsány, lze poměrně snadno samostatně vyřešit, ale bohužel vždy na úkor jiného dílčího problému. Nezbývá proto nyní než hledat cestu, jak dílčí problémy navzájem skloubit, aby výsledek byl co nejlepší pro danou situaci.