ACTA UNIVERSITATIS AGRICULTURAE ET SILVICULTURAE MENDELIANAE BRUNENSIS
SBORNÍK MENDELOVY ZEMĚDĚLSKÉ A LESNICKÉ UNIVERZITY V BRNĚ

Ročník LV 11 Číslo 3, 2007

ROZŠÍŘENÍ ALGORITMU ODSTAVCOVÉHO ZLOMU V INVERZNÍM PARADIGMATU SAZBY

J. Přichystal, J. Rybička

Došlo: 15. prosince 2006

Abstract

PŘICHYSTAL, J., RYBIČKA, J.: *Line-breaking algorithm enhancement in inverse typesetting paradigma.* Acta univ. agric. et silvic. Mendel. Brun., 2007, LV, No. 3, pp. 117–122

High quality text preparing using computer desktop publishing systems usually uses line-breaking algorithm which cannot make provision for line heights and typeset paragraph accurately when composition width, page break, line index or other object appears. This article deals with enhancing of line-breaking algorithm based on optimum-fit algorithm. This algorithm is enhanced with calculation of immediate typesetting width and thus solves problem of forced change. Line-breaking algorithm enhancement causes expansion potentialities of high-quality typesetting in cases that have not been yet covered with present typesetting systems.

typesetting, typesetting paradigma, inverse typesetting paradigma, line-breaking algorithm, TEX, Adobe InDesign, printing line grid

Sazba jako proces vkládání proporcionálních symbolů do řádků je v současné době řešena prakticky ve všech případech počítačem. Programy pro počítačovou sazbu byly konstruovány při různých počátečních podmínkách a za různým účelem, z čehož automaticky plynou zcela rozdílné kvalitativní parametry. Profesionální programové systémy vycházejí částečně ze sazby kovovými literami.

Ruční (i strojní) sazba kovovými literami byla realizována algoritmem, z něhož některé systémy počítačové sazby přebírají určité základní prvky.

Principy práce počítačových systémů lze rozdělit na dvě kategorie:

1. Princip s prioritní funkcí textu – prvotním materiálem jsou textové bloky, jejichž seskládáním vznikají stránky. Pracovně tento princip označíme pojmem *přímé paradigma sazby*.
2. Princip s prioritní funkcí sazební plochy – prvotním materiálem jsou prázdné plochy a sestavené stránky, do nichž se vkládá textová masa. Pracovně označíme tento princip pojmem *inverzní paradigma sazby*.

První princip představuje model procesu sazby kovovými písmeny (Pop, 1989). Jeho zásadní výhodou je zpracování samotného textu, na jehož typografickou bezchybnost a preciznost sazby je zde kladen největší důraz.

Druhý princip více vychází z principu činnosti interaktivních grafických počítačových systémů a využívá především jejich grafických možností. Důraz je kladen na stránkové rozložení a na zpracování objektů představovaných jak textovými, tak i grafickými prvky.

Příkladem implementace priority textu je typografický systém TEX a všechny jeho nadstavby. Příkladem implementace priority sazební plochy je systém interaktivního návrhu cílového stránkového designu Adobe InDesign.

Jednoznačnou nevýhodou principu priority textu je obtížná definice stránkového designu, naopak priorita sazební plochy omezuje možnosti základní sazby.

Z uvedeného vyplývá, že žádný princip nesplňuje všechny požadavky, které vycházejí z praktických potřeb. To se projevuje zejména časovou náročností při překonávání nevýhodných vlastností a snížením

kvality výsledného dokumentu. Vzhledem ke složitosti systémů tohoto typu a ke vzájemné datové nekompatibilitě je pro většinu uživatelů (tvůrců dokumentů) nepřijatelné pracovat s oběma systémy a vybírat vždy ten, který je pro danou aplikaci v dokumentu optimální.

Zároveň je potřebné řešit i řadu běžných situací, které žádný z uvedených systémů beze zbytku nepodporuje. Jedná se především o kvalitní sazbu v situacích, kdy odstavec může být přerušen koncem strany (sloupce) a pokračuje v jiném horizontálním rozměru, dále situace, kdy odstavec obsahuje materiál měnící řádkový proklad a vyžadující sazbu na řádkový rejstřík. K těmto případům lze přidat i méně časté potřeby sazby do neobdélníkových tvarů použitých například při obtékání vložených objektů.

MATERIÁL A METODY

Používané algoritmy řádkového zlomu

Pro sazbu proporcionálního textu se používají algoritmy různé složitosti a kvality. Produkty pro pořizování strojopisů nebo textů soukromého charakteru používají algoritmy postavené na principu best-fit. Tyto algoritmy jsou schopny provádět separátní zarovnání pouze v jednotlivých řádcích odstavce. Nelze zde zabránit hrubým typografickým chybám (viz ON 88 2503, 1967), jako jsou posloupnosti stažených a světlých řádků, řeky apod. Těmito systémy, jejichž představitelem může být například Open Office Writer či Microsoft Word, se v tomto textu však nebudeme zabývat.

V kvalitních systémech pro počítačovou sazbu se používají algoritmy vycházející z algoritmu optimum-fit, zpracovávající materiál celého odstavce a realizující základní typografický požadavek na dosažení jednolité šedi sazby. Algoritmus zpracovává vstupní proud dat tvořených prvky tří typů, pro něž se užívají názvy box, glue a penalta. Prvek box představuje veškerý tisknoucí materiál, prvek glue označuje všechny výplňky. Prvek penalta je číselná hodnota ovlivňující hodnotu cenové funkce, sloužící pro výpočet vhodnosti řádkového zlomu v daném místě.

Prvek box je charakterizován především šířkou, výškou a hloubkou (tyto hodnoty jsou důležité pro algoritmus zlomu) a dále pozicí referenčního bodu, kterým se umisťuje na účaří sazby. Atributem prvku glue je šířka, stažitelnost a roztažitelnost. V implementacích algoritmu optimum-fit lze navíc pracovat s roztažitelností a stažitelností boxů — jedná se však o malé hodnoty, neboť tím dochází k deformaci znaků. Algoritmus zde má však větší možnosti pro výpočet optimálních bodů řádkového zlomu. Toto rozšíření je známo pod názvem hz-algoritmus (Zýka, 2004).

Algoritmus pracuje s materiálem celého odstavce. Vypočítává možné body řádkového zlomu, které ohodnocuje cenovou funkcí. Možné body zlomu jsou skládány do orientovaného grafu, jehož hrany jsou ohodnoceny vhodností příslušného zlomu. Vyhledáním nejkratší cesty v tomto grafu (nejmenšího součtu ohodnocení hran) se získá seznam optimálních bodů řádkového zlomu.

Matematický model sazby

Pro proces řádkového zlomu je v systému TeX využita metoda optimum-fit. Princip jejího fungování a její matematický model popisuje Knuth a Plass (1981). Základní aspekty tohto algoritmu rovněž popisuje Olšák (2000). Při návrhu našeho rošíření budeme z tohoto modelu vycházet.

Hodnocení vlastností algoritmu

Vstupními hodnotami jsou požadované délky všech řádků odstavce, algoritmus nijak nezohledňuje výšky sestavených řádků. Tato skutečnost má následující důsledky:

1. V místě stránkového zlomu je zbylá část odstavce vysazena do předem stanovených délek řádků, avšak před začátkem sazby odstavce není známo, kde ke stránkovému zlomu dojde.
2. Není zohledněn požadavek na sazbu do řádkového rejstříku, algoritmus vůbec nepoužívá výšky řádků.
3. Při výskytu boxů s výškou přesahující hodnotu řádkování dojde k posunu účaří, který může mít vliv na požadované délky následujících řádků, to však algoritmus nezohledňuje.

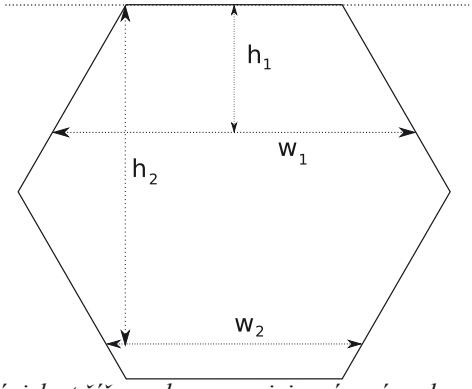
NÁVRH ŘEŠENÍ

Zásadním krokem řešení je změna algoritmu sazby spočívající především ve změně získávání vstupních požadovaných délek řádků.

Délka *i*-tého řádku není dána ještě před zahájením činnosti algoritmu pro zlom daného odstavce, ale je dána v průběhu práce voláním speciální funkce *R*.

Jedním z hlavních problému pri modifikaci paradigmatu algoritmu řádkového zlomu je určení délky jednotlivých řádků při sazbě do neobdélníkových tvarů odstavce. Představíme-li si běžnou sazbu do obdélníkového tvaru, má každý řádek odstavce stejnou délku. Určení této délky není samozřejmě velký problém. Ten ovšem nastává v případě sazby do tvarů odlišných. Jde především o to, že délka řádku v těchto podmínkách je závislá na konkrétní aktuálně vysazené výšce materiálu. Je tedy třeba určit ze znalosti výšky šířku tvaru, do kterého chceme text vsadit. Jak je patrné z příkladu

na obr. 1, první řádky odstavce budou sázeny na šířku menší než řádky přibližně uprostřed odstavce a nakonec se šířka opět zužuje. Protože předpokládáme, že tvar jednotlivých prvků sazby na stránce (tvaru, do kterého se bude sázet text) je zadán vektorovou formou, lze poměrně jednoduše určit i šířku řádku. Jako nejvhodnější formát pro uložení informací o vektorových objektech se jeví formát SVG. Podrobnější informace uvádí Eisenberg (2002).



1: *Závislost šíře sazby na pozici v sázeném obrazci*

Údaje o objektech v SVG formátu jsou uloženy různým způsobem v závislosti na typu vkládaného objektu. Naším úkolem je tedy definovat funkci, která bude schopna na základě informací o vkládaných objektech určit jejich vodorovnou šířku v jakémkoliv bodě, abychom mohli vysadit řádek odstavce s touto zjištěnou šířkou.

Tato funkce, kterou označíme *R,* určuje šířku objektu v dané výšce a bude se tedy lišit pro různé tvary objektu. Obecnou funkci by jistě nebylo obtížné definovat, avšak pro jednoduché útvary, jako je obdélník, by výpočet zabíral příliš mnoho strojového času, a přitom právě obdélník je patrně nejpoužívanější tvar odstavce v běžných dokumentech. Tvary, které budeme uvažovat jako nejčastější a nabízené možné tvary odstavce, jsou obdélník, polygon, kružnice a elipsa.

Budeme-li v této kapitole mluvit o výšce řádku, máme na mysli výšku *h* nejvyššího boxu řádku. Mluvíme-li o výšce vysazeného materiálu, máme na mysli soucet výšek *h* a hloubek *d* boxů tvořících vysazené řádky.

Pro obdélník je určení šířky sazby velmi snadné, neboť šířka řádku l_j je rovna straně obdélníka, čili v definici SVG se jedná o hodnotu parametru width.

Určení požadované šířky řádku l_j odstavce vyplňujícího kružnici lze realizovat na základě vztahu $l_j = 2 \cdot \sqrt{(h \cdot (2r - h))}$, kde *h* je výška aktuálně vysazeného materiálu zvýšená o výšku řádku a *r* je poloměr kružnice.

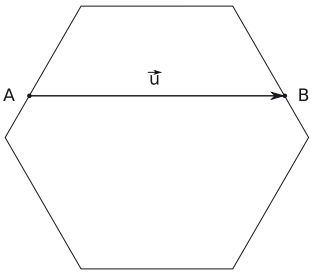
U elipsy je určení šíře řádku poněkud obtížnější. Šířku řádku určíme ze znalosti parametru elipsy a její rovnice:

$$l_j = 2 \cdot \left| \sqrt{(r_x^2 - \frac{r_x^2}{r_y^2}) \cdot h^2 - c_x} \right|,$$

kde r_x a r_y jsou délky poloos elipsy, c_x je *x*-ová souřadnice středu elipsy a *h* je výška vysazeného materiálu zvýšená o výšku řádku.

Polygon bývá definován množinou vrcholů, které jsou krajními body ohraničujících úseček. Díky tomu mužeme určovat šířku řádku jako vzdálenost dvou bodů A a B, které leží na těchto úsečkách a jejich kolmá výška od vrcholu tvaru odpovídá výšce sazby aktuálního řádku. Tyto body určíme jako průsečíky vodorovné přímky $y = kx + q$ reprezentující účaří a vedené ve výšce *h* s hranicemi zadaného útvaru. Protože tato přímka protínající polygon je vždy rovnoběžná s osou *x* (uvažujeme pouze sazbu odstavců s vodorovnými řádky), bude směrnice přímky $k = 0$. Hodnota *q* odpovídá hodnotě *y* souřadnice nejvyššího vrcholu polynomu zvýšené o součet výšky již vysazeného materiálu a výšky řádku. Vzdálenost dvou nalezených průsečíků, bodů A a B, pak lze určit jako velikost vektoru *u* umístěného do bodu A = $[a_0, a_1]$ a B = $[b_0, b_1]$ (obr. 2):

$l_j = \mathrm{d(A,B)} = |\mathrm{vector}(u)| = \sqrt{(q(b_0 - a_0)^2 + (b_1 - a_1)^2)}$.



2: *Určení aktuální šíře sazby z pozice v sázeném obrazci*

Modifikovaný algoritmus

Modifikace algoritmu řádkového zlomu navrhovaného řešení je založena na algoritmu používaném v TEXu. Algoritmus, stejně jako jeho předloha, očekává na vstupu horizontální seznam (boxy, glue, penalty) a na výstup produkuje seznam pozic vhodných pro řádkový zlom. Výsledné řádky jsou získány provedením zlomu v těchto místech. Algoritmus vytváří z možných míst zlomu graf a v něm vyhledává nejkratší cestu. Tak získá optimální zalomení odstavce.

Modifikace modelu

Pokusme se nyní formulovat modifikovaný algoritmus řádkového zlomu za pomocí matematických

konstrukcí. Předpokládejme v souhlase s modelem (Knuth a Plass, 1981), že text odstavce je posloupnost m prvků $x_1x_2...x_m$, kde každé x_i je buď box, glue nebo penalta. Box je specifikován hodnotami w_i a h_i, glue třemi hodnotami (w_i, y_i, z_i) a penalta hodnotami (w_i, p_i, f_i). Zjednodušeně tedy můžeme říci, že odstavec $x_1...x_m$ je specifikován sedmi posloupnostmi:

$t_1...t_m$, kde t_i je typ prvku x_i (box, glue, penalta),

$w_1...w_m$, kde w_i je šířka prvku x_i,

$h_1...h_m$, kde h_i je výška prvku xi,

$y_1...y_m$, kde y_i je roztažitelnost prvku x_i, pokud jde o glue (t_i = glue), jinak $y_i = 0$,

$z_1...z_m$, kde z_i je stažitelnost prvku x_i, pokud jde o glue (t_i = glue), jinak $z_i = 0$,

$p_1...p_m$, kde p_i je hodnota penalty prvku x_i, pokud (t_i = penalta), jinak $p_i = 0$,

$f_1...f_m$, kde $f_i = 1$, jestliže x_i je označená penalta, jinak $f_i = 0$.

Poslední hodnotou, která vstupuje do algoritmu řádkového zlomu, je požadovaný tvar odstavce s, kde s je celé číslo z intervalu ‹1, 4› . Tento parametr je důležitý pro výpočet délky řádku, avšak nespecifikuje konkrétní řádek. Hodnota s se v průběhu zlomu odstavce nemění. V závislosti na tvaru odstavce, definovaném pomocí grafického rozhraní, může nabývat hodnot:

1 – obdélníkový tvar odstavce,

2 – kruhový tvar odstavce,

3 – elipsový tvar odstavce,

4 – obecný, mnohoúhelný tvar odstavce (polygon).

Použité tvary odstavce vycházejí ze základních tvarů definovaných ve formátu SVG používaném pro popis designu stránky.

Mějme definován tvar odstavce získaný z grafického rozhraní. Z definovaného tvaru odstavce musíme určit délky jednotlivých řádků. Na první pohled je to velmi snadné. Jednoduché je to ovšem pouze pro odstavec, ve kterém se nevyskytují objekty narušující řádkový rejstřík, čili kde $h_1 = h_2 = ... = h_i = ... = h_m$. Pokud se však v odstavci objeví například matematické vzorce, může při jejich větší složitosti dojít k tomu, že některé řádky budou mít výšku odlišnou od řádků jiných, čili obecně $h_1 \neq h_2 \neq ... \neq h_i \neq ... \neq h_m$. V tuto chvíli je určení výšky jednotlivých řádků, a potažmo jejich šířky podstatně složitější.

Mějme funkci R, která na základě znalosti požadovaného tvaru odstavce a výšce již vysazeného materiálu odvodí požadovanou šířku řádku j označenou jako l_j. Funkce R získá hledanou šířku l_j na základě vstupní informace o tvaru odstavce (hodnota s) a výšce již vysazeného materiálu (hodnota H_{vysaz}). Šířku řádku v daném bodě lze vypočítat jako šířku tvaru odstavce s použitím matematických vzorců uvedených v části Návrh řešení.

Práce funkce R je v případě vyrovnaného řádkového rejstříku, čili odstavce bez vertikálně vyčnívajících prvků jednoduchá. Zde má každý řádek výšku h_v. V případě opačném lze říci, že při četnosti výskytu c jednotlivých rušivých prvků s výškou h_{p1} … h_{pm} může být výška výsledného odstavce H v rozmezí $(\max(h_{pi})) + (n - 1) \times h_v \leq H \leq (h_{p1} + h_{p2} + \ldots + h_{pm})$, kde $\max(h_{pi})$ udává výšku nejvyššího prvku v odstavci. Výsledná výška tedy závisí na tvaru odstavce a v případě výskytu objektů narušujících řádkový rejstřík i na okolnosti, jaké bude jejich uspořádání na výsledných řádcích. Možné varianty jsou:

1. Jednotlivé rušivé objekty se podaří umístit na jeden řádek, výška odstavce bude nejmenší.
2. Objekty budou rozmístěny v odstavci na řádcích neuspořádaně, někdy jeden, někdy více na jednom řádku.
3. Rušivé objekty se uspořádají tak, že každý bude na jiném řádku a výška odstavce tak bude největší.

Je potřebné rovněž uvažovat případ, kdy uživatel bez ohledu na objekty v odstavci zadá posloupnost h_1, …, h_m. Toho lze využít především při sazbě na řádkový rejstřík.

DISKUSE

Modifikovaný algoritmus zohledňující aktuální pozici účaří pro výpočet požadované délky řádku poněkud zvyšuje koeficient výpočetní složitosti (kvadratický charakter původního algoritmu však zůstává zachován), kterou platíme za zvýšení možností kvalitní sazby.

Pro usnadnění a zrychlení procesu výpočtu odstavcového zlomu lze vymezit některé zvláštní případy:

1. Sazba do běžného obdélníkového obrazce, se stránkovým (sloupcovým) zlomem neměnícím šíři sazby.
2. Sazba na řádkový rejstřík, kdy jsou pozice účaří známy dopředu a požadované délky řádků lze vypočítat před zahájením algoritmu odstavcového zlomu.

V prvním z uvedených případů se algoritmus může chovat stejně jako v původní nemodifikované podobě. Ve druhém případě lze vstupní hodnoty délek řádků dodat podobným způsobem, jako to řeší například příkaz \parshape systému TEX, avšak požadované délky nejsou konstantami zapsanými ve zdrojovém textu, ale výsledky funkce R aplikované na požadovaná místa účaří uvažovaného odstavce.

Implementaci modifikovaného algoritmu lze provést jako rozšíření funkce jádra sázecího stroje systému TEX. K tomuto procesu existují standardní nástroje — jazyk WEB popisující a dokumentující samotný algoritmus, programy konvertující tento obecný zápis

do implementačního jazyka (obvykle jazyk C) a také standardní testy umožňující objektivně zjistit úroveň zpětné kompatibility, která je pro správnou činnost sázecího algoritmu zcela nezbytná.

ZÁVĚR

Rozšíření možností kvalitní počítačové sazby vytváří základní předpoklady pro tvorbu dokumentů za podmínek, které nebyly doposud pokryty používanými typografickými systémy.

Předpoklad kvalitní sazby, tj. vysoce kvalitní algoritmus odstavcového zlomu, je implementován tak, že neumožňuje uživateli využít sazbu v inverzním paradigmatu. Současné systémy je tedy potřebné doplnit o možnost implementace sazebního algoritmu obohaceného o vstup aktuálních vertikálních rozměrů sazby a o okamžitý výpočet potřebné řádkové šíře.

SOUHRN

Kvalitní příprava textů pomocí počítačových publikačních systémů běžně používá algoritmus odstavcového zlomu, který neumožňuje plně zohlednit výšky jednotlivých řádků a přesně vysadit odstavec při změně šíře sazby vynucené předchozí sazbou, zlomem strany, řádkovým rejstříkem nebo objektem narušujícím řádkování.

Článek se zabývá definicí rozšíření možností algoritmu odstavcového zlomu na principu optimum-fit o výpočet okamžité požadované šíře sazby řešící problém vynucené změny z uvedených důvodů. Rozšíření algoritmu odstavcového zlomu má za následek rozšíření možností kvalitní sazby v případech, které současnými systémy nejsou zcela pokryty.

sazba, paradigma, inverzní paradigma sazby, algoritmus odstavcového zlomu, TEX, InDesign, řádkový rejstřík

LITERATURA

ČSN 88 0100 Základní a společné názvy – polygrafické názvosloví. 1977.

EISENBERG, J. D.: SVG Essentials, O'Reilly 2002, ISBN 0-596-00223-8.

FINE, J.: Instant Preview and the TEX daemon, EuroTEX 2001, Conference Proceedings, 49–58.

GOOSENS, M., RAHTZ, S., MITTELBACH, F.: The LATEX Graphics Companion. Addison Wesley 1994, ISBN 0201854694.

JELÍNEK, J.: Nadstavba typografického systému. Brno: MZLU, 2004, diplomová práce.

KNUTH, D.: Computer and Typesetting Series, Vol. A: The TEXBook. Addison Wesley 1984, ISBN 0-201-13448-9.

KNUTH, D.: Computer and Typesetting Series, Vol. B: TEX The Program. Addison Wesley 1986a, ISBN 0201-13437-3.

KNUTH, D.: Computer and Typesetting Series, Vol. C: The METAFONT Book. Addison Wesley 1986b, ISBN 0-201-13444-6.

KNUTH, DONALD E., PLASS, MICHAEL F.: Breaking Paragraphs Into Lines. Software – Practice and Experience. 11(11), pp. 1119–1184, November 1981.

OLŠÁK, P.: TEXbook naruby. Brno: Konvoj 2000. 468 s., ISBN 80-7302-007-6.

ON 88 2503 Základní pravidla sazby. Praha: ČSÚN, 1969.

PAZDZIORA, J.: Algoritmy řádkového a stránkového zlomu v počítačové sazbě. Brno: 1997. 56 s., diplomová práce.

POP, P., FLÉGER, P., POP, J.: Sazba I – Ruční sazba. Praha: SPN, 1989.

PŘICHYSTAL, J., RYBIČKA, J.: Rozšíření možností typografického systému TEX. In Firma a konkurenční prostředí 2004. IS/IT a konkurenceschopnost podniku. Brno: Konvoj, 2004, s. 143–153 ISBN 80-7302-079-3.

PŘICHYSTAL, J., RYBIČKA, J.: Moderní přístupy tvorby dokumentu obchodních podnikatelských subjektů. In Motyčka, A. Informatika XV /2004 – Sborník příspevků. 1. vyd. Brno: Konvoj, 2005, s. 89–95. ISBN 80-7302-067-X.

SKOUPÝ, K.: Proposal for Text and Graphics Unification in a TEX-Based Framework. In Proceedings of EuroBachoTEX'2002.

ZÝKA, V.: Používáme pdfTEX IV: mikrotypografické rozšíření. Zpravodaj CSTUG. Brno: Konvoj, 2004, 2, s. 47–53. ISSN 1211-6661.

Adresa

Ing. Jan Přichystal, Ph.D., doc. Ing. Jiří Rybička, Dr., Ústav informatiky, Mendelova zemědělská a lesnická univerzita v Brně, Zemědělská 1, 613 00 Brno, Česká republika