Dlaczego separować na RIPie? Dyskusja i przykłady

Radosław Tryc

TEX-em można zrobić wszystko

Podobnie jak PostScript, TEX jest językiem programowania, pozwalającym radzić sobie z bardzo różnorodnymi zadaniami. Możliwe okazuje się tworzenie baz danych, samodzielne przetwarzanie grafiki czy nawet interpretowanie innych języków. Miałem okazję oglądać seminarium, na którym autor prezentował tworzenie w TEX-u maszyny stosowej, co jest warunkiem wystarczającym do napisania interpretera języka Forth. Jeszcze krok i mielibyśmy RIPa PostScriptowego napisanego w TEX-u.

Oglądając taki pokaz komputerowej wirtuozerii miałem wiele wątpliwości – czy nie warto poszukać prostszego rozwiązania. Podoba mi się podstawowa idea systemów unixopodobnych, program musi być mały i robić tylko jedną rzecz – bezbłędnie. TEX genialnie składa strony pełne tekstu, matematyki i trudnych symboli. Czyż to mało ?

Inny warunek, zapożyczony ze współczesnych przygotowalni poligraficznych to OO (ang. *Output Once* – rób raz). Oczekuję, żeby PostScriptowy plik wynikowy, wyprodukowany przez dvips, mógł być wykorzystany do wydruku na drukarce kolorowej (np. po przetworzeniu GhostScriptem) i do naświetlenia rozseparowanych klisz do druku.

textBlack i inne czarne myśli

Zawartość przygotowywanego do druku materiału podzielimy na tekst i ilustracje (grafiki). Przyjmuję założenie, że grafiki są włączane w postaci plików EPS i TIFF z kolorami zapisanymi jako RGB bądź CMYK, natomiast TEX sam radzi sobie z tekstem. Fakt, że LATEX2ε sam potrafi składać nawet elektroniczne schematy ideowe, pomijam milczeniem. Taka grafika, z punktu widzenia traktowania kolorów, zachowuje się jak tekst.

Wiadomo, że wykorzystując opisane w dokumentacji do dvips-a instrukcje sterujące, da się „kolorować" tekst. Oglądając implementację *colordvi* nie sposób uniknąć wrażenia, że autor chciał jedynie uzyskać nowy sposób wyróżniania fragmentów tekstu, zamiast kursywą – na czerwono.

Dvips ma przykry zwyczaj generowania PostScriptu, w którym tekst domyślnie nie jest czarny, tylko ma 400% farby: 100% C, 100% M, 100% Y i na szczęście także 100% K. Takie coś drukuje się poprawnie na wszystkich drukarkach czarno-białych, nie sprawia kłopotu na kolorowych – atramentowych (niepotrzebnie podnosząc kilkakrotnie koszt wydruku) natomiast nie nadaje się dla drukarni. Drukarz nie ma praktycznie szansy idealnego dopasowania poszczególnych rozbarwień, więc wokół każdej litery otrzymamy barwną poświatę. Co gorsza, 400% przekracza wytrzymałość papieru i powoduje przesiąkanie, rozmazywanie i sklejanie się arkuszy.

Instrukcje kolorujące zawarte w pakiecie *colordvi* mogą nas ratować jeżeli mamy do czynienia z czysto tekstową pracą. Wystarczy raz napisać `\textBlack` i wszystko odtąd będzie idealnie czarne.

Grafiki z zewnątrz TEX-a bardzo często przełączają bieżącą barwę tekstu (czarną) na domyślną (400% normy). Jeżeli wykorzystujemy LATEX2ε i pakiet *graphicx*, mamy jeszcze gorszą sytuację. Automatyczne umieszczanie ilustracji powoduje, że to przełączenie może nastąpić w zupełnie nieoczekiwanym (z punktu widzenia treści pliku `*.tex`) momencie. Każdy kontekst graficzny (`figure`, `minipage` itp.) ma też swoją osobną barwę tekstu, co powoduje, że musi zostać pokolorowany na czarno. W praktyce `\textBlack` pojawia się w poprawnie przygotowanym pliku `*.tex` dosyć często.

Nie wiem, czy jest to planowane, ale jedynym ogólnym rozwiązaniem problemu koloru tekstu może być zmiana domyślnego zachowania dvipsa. W innym przypadku trudno mieć przekonanie czy w pliku umieściliśmy odpowiednią liczbę wywołań `\textBlack`.

Co robi RIP?

W zasadzie współczesne RIPy PostScriptowe można podzielić na takie, które separują przed lub po rastrowaniu. Te drugie mają mniejsze wymagania wobec procesora, ale większe wobec dysku.

Separacja przed rastrowaniem opiera się na pojęciu instrukcji rysujących (ang. *painting operators*). Wyobraźmy sobie najprostszy z możliwych plik PS:

```
0 0 0 1 setcmykcolor  % Ustawienie
        %% bieżącej barwy - czarnej
0 0 moveto % Początek rysowania ścieżki
72 0 lineto   72 72 lineto   0 72 lineto
```

```
closepath % Zamknięcie ścieżki
stroke % JEDYNA instrukcja rysująca
showpage
```

Rozważmy separację na RIPie w czasie interpretacji. Zmienia się stan bieżących parametrów graficznych (barwa i ścieżka), ale dopiero gdy wystąpi instrukcja rysująca, tworzona jest DL (*Display List* – format pośredni między PS a bitmapą).

Jeżeli RIP napotka instrukcję definiującą barwę do separacji (np. setcmykcolor, setrgbcolor lub sethsbcolor) dla każdej strony będzie chciał tworzyć cztery DL (CMYK). Jeżeli oprócz tego w pliku PS pojawi się instrukcja `\Separation` definiująca dodatkowy nieseparowalny kolor, RIP wygeneruje jeszcze jedną DL (piątą, szóstą...).

Plik PS, nawet zawierający dodatkowe kolory, jest poprawnie obsługiwany (pokazywany i drukowany) przez GhostScript dzięki temu, że definicja `\Separation` zawiera informację jak poradzić sobie, posiadając tylko CMYK-a.

Domyślnie RIPy PostScriptowe starają się separować na CMYKa (+dodatkowe). Bardzo często się zdarza, że grafiki włączane do TEX-a zostały przygotowane w innej reprezentacji (np. RGB). Niestety przeliczenie między RGB a CMYKiem nie jest jednoznaczne i zależy od bardzo wielu czynników. RIP zawiera *Color Rendering Dictionary*, gdzie znajdują się funkcje przejścia między różnymi przestrzeniami barwnymi. Domyślne wartości zaproponowane przez producenta mogą zupełnie nie pasować do tych, które odpowiadają naszej technice druku. Jeżeli nie stworzyliśmy własnego CRD, nie powinniśmy z niego korzystać.

Ta ostatnia uwaga dotyczy zdjęć i innych grafik mających odniesienia do rzeczywistości. Wszelkiego rodzaju diagramy, schematy i wykresy będą wyglądały nieźle, nawet jeśli zamienimy cyan z magentą – CRD jest dobre.

Jak zobaczyć, co zrobi RIP?

Program PostScriptowy o nazwie *aurora* stworzony przez Grahama Freemana może posłużyć jako ilustracja procesu separacji na RIPie. Stworzenie żółtej separacji z naszego kolorowego pliku PS wymaga jedynie dołożenie nagłówka z odpowiednimi parametrami. Odpowiednia komenda w UNIX-ie może wyglądać tak:

```
$ cat aurora 150.lpi yellow \
       mójplik.ps > mójżółty.ps
```

Aurora przedefiniowuje instrukcje rysujące tak, żeby działały tylko w zadanym kolorze. Podstawową różnicą między separacją na RIPie a aurorą jest czas interpretacji. Do stworzenia kompletu klisz wykonanie separacji poprzez dodawanie nagłówków zajmie co najmniej cztery razy więcej czasu.

Aurora nie obsługuje również bardziej zaawansowanych funkcji PostScriptu Level 2 takich jak wspomniane już `\Separation`. Pod kolorem dodatkowym zawsze będzie widać jego CMYK-ową emulację.

Znając ograniczenia programu, można się nim posługiwać do sprawdzania plików PS pod kątem przydatności do druku. Przed nadwietleniem da się na ekranie oglądać pracę klisza po kliszy.

Czy RIP zrobi dobrze?

Krótka odpowiedź na tak postawione pytanie brzmi – nie zawsze. Strona PS składa się z obiektów, które leżą na sobie w zadanej kolejności. RIP ma obowiązek przełożyć to na farby, które zawsze są nieco przezroczyste. Posługuje się przy tym jedynie informacjami z pliku PS.

Dla zilustrowania trudności rozważmy czarny tekst na zdjęciu. Pomijając zagadnienie czytelności, musimy od razu rozważyć trzy sytuacje:

- Duży stopień pisma, ciężkie litery – zdjęcie może być widoczne poprzez litery. Należy usunąć zdjęcie spod liter, zostawiając zalewkę. Czerń napisu powinna być wzmocniona jednolitą aplą w CMY (rys. 1).
- Niewielki stopień pisma – nadrukowujemy tekst na zdjęcie. Usuwanie zdjęcia pod literami jest błędem, może prowadzić do prześwitów (rys. 3).
- Średni stopień pisma, lekkie litery – wyjmujemy ze zdjęcia tyle, by naokoło liter zostawały białe obwódki (rys. 2).

Jak mieć pewność, że jest dobrze?

Dostępne oprogramowanie komercyjne (Luminous TrapWise, Ultimate Trapeeze) potrafi automatycznie wynajdywać i ustawiać odpowiednie parametry nadruku i zalewkowania w plikach PS. Użytkownik musi jedynie wpisać informacje na temat farby (kryjąca, przezroczysta, rozlewająca się) i precyzji pasowania maszyny drukarskiej (głębokość zalewek).

Jeżeli znane są parametry maszyny i farb już na etapie tworzenia dokumentu, można skorzystać ze wspaniałego pakietu cmyk-hax (Piotra Pianowskiego), który w pewnym zakresie pozwala ustawiać zalewki w samym TEX-u dla obiektów tworzonych przez niego. Mimo że jest to skuteczne dla obiektów tworzonych w TEX-u, wzdragam się przed stosowaniem tej metody. Zamiast jednego kolorowego pliku PS, otrzymujemy co najmniej cztery czarno-białe pliki. Żeby mieć OO (rób raz) trzeba by je scalić, co wcale nie jest łatwe.

Prościej?

„Zabawy z kolorami są trudne”. Wiele prac w kolorze nie wymaga jednak wytaczania najcięższych dział. Wystarczy żeby prawidłowo wykonane obiekty nie nakładały się na siebie i już nie ma kłopotu. Jeżeli w tekście zamiast kursywy zastosowaliśmy kolor, dołączyliśmy zdjęcia i wykresy wykonane gnuplotem, można zapomnieć o zalewkowaniu, parametrach zadruku i „wybijaniu” koloru.

Musimy zadbać o to, żeby poszczególne obiekty, które wkładamy na strony składane TEX-em były prawidłowo wykonane. Jest to znacznie łatwiejsze w małej skali, niż poszukiwanie Ogólnej Teorii Zalewkowania (OTZ). Kilka przykładów:

- **Wykresy**. Darmowy program gnuplot ma możliwość otaczania jednobarwnych pól lekko poszerzoną czarną kreską. Nawet przy dużym „rozjeździe” wynik będzie niezauważalny.
- **Tekst na zdjęciu**. Ze względu na czytelność, powinno się unikać takiej sytuacji. Bardzo dobrym narzędziem do ręcznego poradzenia sobie z taką sytuacją jest darmowy Mayura Draw (Windows 95).
- **Rysunki techniczne**. Dobrym rozwiązaniem jest otaczanie wszystkich, jednolitych płaszczyzn lekkim, czarnym konturem. Ewentualne niepasowania będą zakryte czarnym.
- **Znaki Firmowe**. Tutaj kontury nie wchodzą w rachubę. Pracując programem graficznym trzeba samemu rozpoznać trudne krawędzie i wstawić odpowiednie, dodatkowe elementy (np. wąski pasek pomarańczowy między czerwonym a żółtym polem).

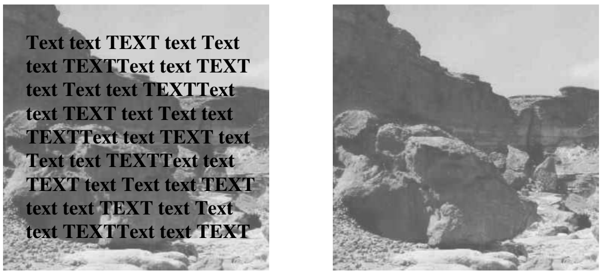
Wady i zalety

Do separacji na RIPie tworzymy pojedynczy plik PS zawierający całą informację dotyczącą prezentacji dokumentu. Może on posłużyć to stworzenia pliku

Tekst wydrukowany na zdjęciu, po lewej stronie widzimy efekt, po prawej – jak zmienione zostało zdjęcie.

Rys. 1. Czarny tekst na zdjęciu – przykład zalewki (ang. *trap*) i wzmocnienia czerni

Rys. 2. Ze zdjęcia została wybita (ang. *knockout* obwódka – dla podkreślenia kształtu



Rys. 3. Gdy tekst jest zbyt drobny, powinien być nadrukowany (ang. *overprint*). Nie ma zalewek ani wybijania

PDF, do obejrzenia go Ghostscriptem, do wydruku próbnego na drukarce barwnej. W celu „uszlachetnienia” pracy, jaką wykonaliśmy w TEX-u, da się użyć specjalistycznego oprogramowania komercyjnego (zalewki, układanie stron na arkuszu). Jeżeli tego chcemy, na samym RIPie można podejrzeć PostScript i dokonać ostatecznej korekty kolorów.

Cztery czarno-białe pliki PS, dają większą kontrolę nad procesem produkcji. Uniemożliwiają jakąkolwiek zmianę po wytworzeniu PS. Ponadto wydruki próbne z poziomu TEX-a pozwalają dostosować sposób tworzenia bitmapowych fontów do możliwości urządzeń.

◇ Radosław Tryc
radek@pi.com.pl