Uniwersytet Warszawski
Wydział Matematyki, Informatyki i Mechaniki

Jakub Wilk
Nr albumu: 209508

Rozbudowa pakietu oprogramowania *DjVuLibre*

Praca magisterska
na kierunku INFORMATYKA

Praca wykonana pod kierunkiem
dra hab. Janusza Bienia, prof. UW
Katedra Lingwistyki Formalnej UW

Lipiec 2008

Oświadczenie kierującego pracą

Potwierdzam, że niniejsza praca została przygotowana pod moim kierunkiem i kwalifikuje się do przedstawienia jej w postępowaniu o nadanie tytułu zawodowego.

Data Podpis kierującego pracą

Oświadczenie autora (autorów) pracy

Świadom odpowiedzialności prawnej oświadczam, że niniejsza praca dyplomowa została napisana przeze mnie samodzielnie i nie zawiera treści uzyskanych w sposób niezgodny z obowiązującymi przepisami.

Oświadczam również, że przedstawiona praca nie była wcześniej przedmiotem procedur związanych z uzyskaniem tytułu zawodowego w wyższej uczelni.

Oświadczam ponadto, że niniejsza wersja pracy jest identyczna z załączoną wersją elektroniczną.

Data Podpis autora (autorów) pracy

Streszczenie

W pracy przedstawiono metodę kompresji obrazów i format dokumentu DjVu oraz pakiet swobodnego oprogramowania *DjVuLibre*, obsługujący dokumenty w tym formacie. Następnie zaprezentowano nowe, autorskie oprogramowanie poszerzające możliwości tego pakietu.

Słowa kluczowe

DjVu, kompresja obrazów, biblioteki cyfrowe, PDF, OCR

Dziedzina pracy (kody wg programu Socrates-Erasmus)

11.3 Informatyka

Klasyfikacja tematyczna

I Computing Methodologies
I.7 Document and Text Processing
I.7.4 Electronic Publishing

Tytuł pracy w języku angielskim

Enhancement of the *DjVuLibre* software package

Nota licencyjna

Licensing note

Spis treści

Wprowadzenie

Dostęp do tradycyjnych, wydanych na papierze książek, często nastręcza pewnych *fizycznych* trudności: odległość miejsca zamieszkania czy pracy do najbliższej biblioteki jest najczęściej niezerowa; odległość do biblioteki, w której znajduje się jakaś *konkretna* książka bywa większa; kwerenda biblioteczna jest zazwyczaj zadaniem nietrywialnym; wyszukanie pojedynczej informacji nawet w konkretnym tomie ma w pesymistycznym przypadku taką samą złożoność jak przeczytanie jego całości.

Między innymi z tych powodów oraz wobec powszechności dostępu do Internetu, popularność zdobywają *biblioteki cyfrowe*. Ale i użytkownik takiej biblioteki może napotkać *fizyczne* przeszkody: pobranie dokumentu (lub jego fragmentu), a następnie jego przetworzenie w celu wyświetlenia, zajmuje czas i pamięć operacyjną komputera. Formaty graficzne stosowane typowo w Internecie, tj. JPEG, GIF i PNG, nie nadają się do udostępniania skanowanych książek z powodu niedostatecznej kompresji, dużych wymagań pamięciowych i niskiej wydajności dekompresji, braku sposobu indeksowania treści; co więcej, każdy taki pliki reprezentuje tylko jeden obraz, a nie wszystkie strony dzieła.

Problemy te rozwiązuje, będący jednym ze standardów współczesnych bibliotek cyfrowych, format *DjVu*. Jest on *otwarty* w podwójnym sensie: po pierwsze, jego specyfikacja jest publicznie dostępna; po drugie, dostępne jest swobodne oprogramowanie (ang. *free software*), za pomocą którego można czytać i tworzyć dokumenty. Ów dostępny zestaw programów nie zasłużył jednak dotąd na miano *kompletnego*.

Pierwsza część pracy przedstawia główne idee kompresji zastosowane w DjVu, a także sam format, wraz z niektórymi niedostatecznie dotąd udokumentowanymi szczegółami. Równolegle został on porównany do innego formatu dokumentu — PDF. Wyjaśnione zostały również pewne nieścisłości w specyfikacji DjVu.

W drugiej części pracy opisane zostały możliwości — ale i niedostatki — pakietu swobodnego oprogramowania *DjVuLibre*, obsługującego dokumenty w tym formacie. Pobieżnie przedstawione zostały również niektóre inne swobodne programy obsługujące DjVu.

Nowemu, powstałemu w ramach niniejszej pracy, oprogramowaniu do obsługi formatu DjVu, poświęcona jest część ostatnia. Każdy z programów został opisany pod względem innowacyjności, dostępnych funkcji, interfejsu, wybranych aspektów implementacji i możliwości rozwoju. Kody źródłowy programów i binarne pakiety debianowe został zamieszczone na dołączonej do pracy płyty CD.

Skutkiem ubocznym prac implementacyjnych było odnalezienie w *DjVuLibre* pewnej ilości błędów. Do pracy, jako dodatek, dołączony został ich wykaz.

Rozdział 1

Format DjVu

1.1. Wprowadzenie

DjVu to stosunkowo nowa technika kompresji, zoptymalizowana do celów dygitalizacji skanów, w tym również skanów kolorowych, ale z powodzeniem dająca się zastosować również do dokumentów elektronicznych. Nazwa ta oznacza również format pliku, w którym zastosowano wspomniane techniki, opracowany specjalnie na potrzeby udostępniania dokumentów w Internecie.

1.1.1. Historia

Przedsięwzięcie pod nazwą DjVu zostało rozpoczęte w 1996 przez Yanna LeCuna w laboratoriach AT&T. W badaniach, prócz LeCuna, brali udział m.in.: Léon Bottou, Patrick Haffner, Paul Howard, Pascal Vincent, Yoshua Bengio i Bill Riemers.

W 1999 roku została opublikowana specyfikacja wersji 2 formatu DjVu — zobacz [1].

W 2000 roku technologia DjVu została nabyta przez firmę LizardTech[1] , która obecnie *komercjalizuje* DjVu. Jeszcze w tym samym roku firma ta udostępnia na swobodnej licencji *bibliotekę referencyjną* (zobacz 2.1.1).

W 2005 roku LizardTech opublikował specyfikację wersji 3 formatu DjVu — zobacz [2]; zawiera ona również opis różnic między pomniejszymi jego wersjami (zobacz [2, s. 25]).

Faktyczne i proponowane zmiany formatu w stosunku do [2] znajdują się w [3].

1.2. Obrazy rastrowe

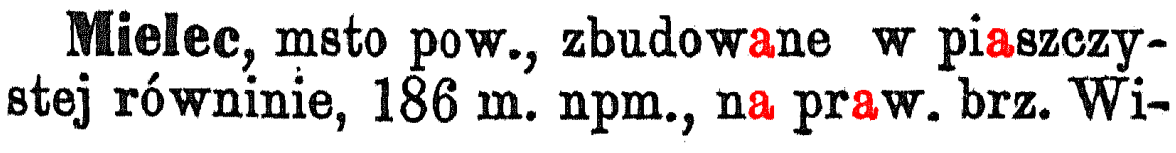
1.2.1. JB2

Fundamentem formatu DjVu jest technika kompresji obrazów czarno-białych o nazwie *JB2* (znana też jako *DjVuBitonal*), przeznaczona zwłaszcza do tekstu i prostej grafiki. Wysoki współczynnik kompresji jest uzyskany poprzez wykorzystanie faktu, że tego typu obrazy składają się z wielu *podobnych* kształtów (np. liter — zobacz przykładowe rysunki 1.1 i 1.2).

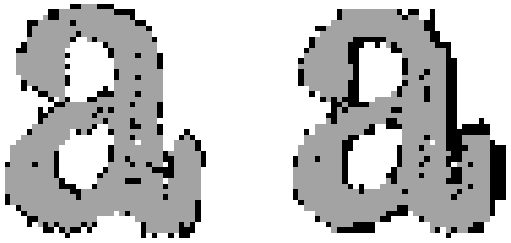
Kompresja, koncepcyjnie, polega na:

- Podziale obrazu na rozłączne jednobarwne kształty. W najprostszym przypadku stosuje się tu analizę spójnych składowych, ale możliwe są bardziej wyrafinowane techniki (np. takie jak opisane w [4]).

[1] LizardTech jest od 2003 roku częścią Celartem Technology Inc.



Rysunek 1.1: Litery *a* w tekście (fragment [5, t. VI, s. 337]) wyglądają identycznie; w rzeczywistości są jedynie podobne.



Rysunek 1.2: Wyróżnione piksele, którymi różnią się górne i dolne litery *a* z rysunku 1.1.

- Utworzeniu słownika kształtów i ciągu *naniesień* (ang. *blits*) tych kształtów na początkowo pusty obraz, które odtworzą obraz źródłowy. Niektóre kształty mogą być zakodowane na podstawie innych; tworzy się w ten sposób swoista ich hierarchia, przykładowo zilustrowana na rysunku 1.3. Nie jest zabronione pojawianie się w słowniku kształtów, które *nie występowały* w oryginalnym obrazie i nigdy nie zostaną naniesione.

Wewnętrznie, jako uniwersalna metoda kompresji, stosowane jest kodowanie arytmetyczne.

Opcjonalnie, słownik kształtów może być wieloczęściowy: kształty z jednego słownika mogą być definiowane na podstawie kształtów z innego. Ponieważ spodziewane jest, że kształty występujące na różnych stronach będą do siebie podobne, wspólny słownik dla różnych stron może istotnie polepszyć kompresję.

Podobną do JB2 techniką kompresji jest, stosowana w formacie PDF (zobacz [6, s. 80–84]), JBIG2.

Według [7], typowa strona zeskanowana w rozdzielczości 300 dpi, zajmuje w formacie JB2 od 5 do 30 KB.

Specyfiakcja formatu JB2 znajduje się w [2, s. 44–57].

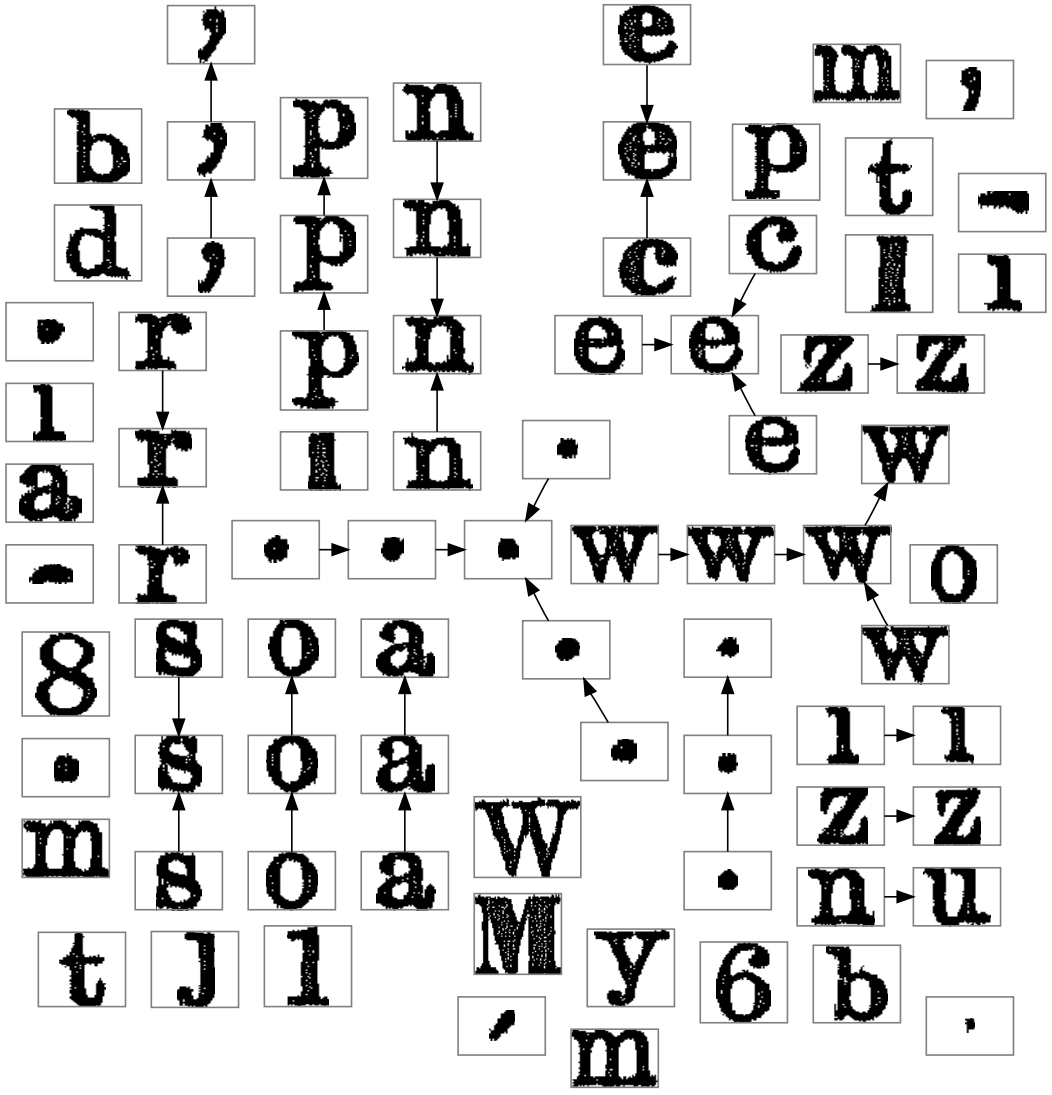
1.2.2. IW44

Konkurencyjnymi w stosunku do wszechobecnego JPEG sposoby kompresji obrazów o płynnych przejściach barwnych (np. zdjęć czy innych obrazów naturalnych) są, oparte na dyskretnej transformacie falkowej: *JPEG 2000* i *IW44* (znany też jako *DjVuPhoto*). JPEG 2000 może być użyty w dokumencie PDF (zobacz [6, s. 86–89]) a IW44 — w dokumencie DjVu.

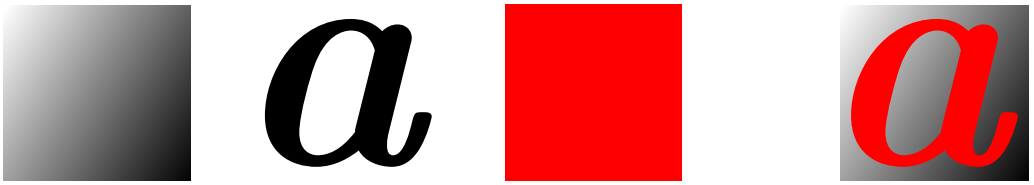
Według [7]:

- Przy zachowaniu tego samego stosunku sygnału do szumu, fotografie zakodowane w IW44 są ok. 2 razy mniejsze niż w JPEG i porównywalnego rozmiaru jak w JPEG 2000.
- Dekompresja obrazu IW44 może być 3 razy szybsza niż dekompresja JPEG 2000.

Podstawowym założeniem projektowym IW44 była możliwość efektywnej rastryzacji jedynie fragmentu obrazu, bez potrzeby utrzymywania w pamięci całego zdekompresowanego



Rysunek 1.3: Hierarchia kształtów dla fragmentu z rysunku 1.1, skompresowanego przy pomocy *cjb2*.



tło maska pierwszy plan obraz wynikowy

Rysunek 1.4: Przykład rekonstrukcji obrazu z trzech warstw składowych.

obrazu. Dzięki temu wymagania pamięciowe przeglądarek DjVu są niewielkie, a mimo to duże obrazy dają się płynnie przewijać.

Koncepcyjnie, obraz IW44 składa się z ciągu *plastrów* (ang. *slices*), na podstawie których odtwarzane są kolejne przybliżenia obrazu wyjściowego. Plastry grupowane są w *kawałki* (ang. *chunks*), a sposób tego grupowania ma wpływ na progresywną rastryzację — przeglądarka może wyświetlić kolejne przybliżenie dopiero po przetworzeniu całego kawałka. Typowy obraz IW44 składa się z 80–120 plastrów, pogrupowanych w 1–4 kawałki, przy czym większość plastrów znajduje się w pierszym z nich.

Specyfikacja formatu IW44 znajduje się w [1, s. 8–22] oraz w [2, s. 30–44],

1.2.3. Struktura strony

W najprostszym przypadku strona dokumentu DjVu to po prostu pojedynczy obraz w formacie JB2 lub IW44. Struktura strony może być jednak bardziej skomplikowana, zgodna mianowicie z modelem *mieszanej treści rastrowej* (ang. *mixed raster content*), zaproponowanej w [8]. W tym modelu, w najprostszym, 3-warstwowym wariancie, strona jest skomponowana z następujących warstw:

- pierwszego planu (ang. *foreground*),
- tła (ang. *background*) i
- binarnej *maski* (ang. *mask*).

Maska opisuje sposób rekonstrukcji obrazu z tła i pierwszego planu: każdy piksel obrazu wyjściowego ma wartość piksela pierwszego planu (jeśli odpowiadający mu piksel maski ma wartość 1) lub tła (w przeciwnym przypadku); na przykładzie ilustruje to rysunek 1.4. Rozbicie obrazu na warstwy pozwala kodować każdą z nich przy zastosowaniu innego algorytmu i z inną rozdzielczością. Taka technika kompresji znakomicie nadaje się do obrazów, na których występują zarówno obiekty o płynnych przejściach barw (np. zdjęcia, faktura papieru), jak i mocno kontrastujące z tłem (np. tekst).

W DjVu, typowo:

- maska jest kodowana w formacie JB2, z pełną rozdzielczością;
- tło jest kodowane w formacie IW44, z możliwością redukcji rozdzielczości (zwykle 3-krotną);
- pierwszy plan jest kodowany:
  - w formacie IW44, z możliwością redukcji rozdzielczości (zwykle 12-krotną);
  - lub poprzez ciąg kolorów kolejnych naniesień kształtów z maski.

Te i inne, rzadko spotykane, sposoby kodowania warstw opisuje [2, s. 6–7].

Fakt, że o niektórych pikselach obrazu IW44 wiadomo, że nie zostaną nigdy wyświetlone, można wykorzystać do uzyskania bardziej efektywnej kompresji. Zgodnie z pomysłem przedstawionym w [9], kolory wspomnianych pikseli wystarczy zastąpić takimi, dla których koszt ich zakodowania będzie minimalny.

Model obrazu w formacie PDF jest znacznie bardziej wyszukany (zobacz [6, s. 34–38]): strona składa się z obiektów różnego rodzaju (grafika wektorowa, obiekty tekstowe, obrazy rastrowe), niekoniecznie rozłącznych, więc być może częściowo się przesłaniających. Ma on wystarczającą siłę wyrazu, by wyrazić w nim model mieszanej treści rastrowej (zobacz [6, s. 349–351]). Z drugiej strony ta ogólna natura powoduje, że trudno w nim optymalizować szybkość rastryzacji (zobacz [7]).

1.3. Struktura dokumentu

Jednostronicowy (ang. *single-page*) dokument DjVu zawiera dokładnie jeden obraz rastrowy (być może z dołączonymi danymi nierastrowymi). *Wielostronicowy* dokument DjVu zawiera 2 lub więcej *pliki składowe* (ang. *component files*). Mogą one zawierać:

- obrazy rastrowy strony (być może z dołączonymi danymi nierastrowymi),
- adnotacje dzielone między stronami,
- słowniki kształtów JB2 dzielone między stronami,
- wcześniej obliczone miniaturki stron.

Dokumenty wielostronicowe występują w dwóch opisanych poniżej formatach.

1.3.1. Spakowane dokumenty wielostronicowe

W formacie *spakowanym* (ang. *bundled*), wszystkie pliki składowe, katalog tych plików i inne pomocnicze informacje zawarte są w pojedynczym pliku dyskowym. Takie tradycyjne rozwiązanie umożliwia wygodne składowanie dokumentów na dysku, przenoszenie na nośnikach danych czy e-mailem. Nie jest jednak ono optymalne w przypadku udostępniania dużych dokumentów w Internecie (lub za pomocą innego medium o niewielkiej przepustowości) — pobieranie dokumentu odbywa się liniowo, zatem obejrzenie n-tej strony wymaga uprzedniego ściągnięcia danych o stronach o numerach mniejszych niż n.

Ten sam problem dotyka dokumentów PDF. W ogólnym przypadku sytuacja jest nawet jeszcze gorsza, mianowicie ściągnięcie *całego* dokumentu może być konieczne do zobaczenia *którejkolwiek* strony. Swobodny dostęp do stron zapewnia dopiero specjalna, tzw. *uliniowiona* (ang. *linearized*), organizacja dokumentu (zobacz [6, s. 1021–1024]). Efektywne nawigowanie po różnych częściach dokumentu jest realizowane w takim przypadku przy pomocy zrywania połączenia (jako jedynego dostępnego sposobu przerwania rozpoczętego transferu) i transferu fragmentu pliku (zobacz [10, “Range Units”]), o ile odpowiednie funkcje obsługują wtyczka przeglądarki i serwer WWW (wraz z ewentualnymi serwerami *proxy*). Niestety, jedyna istniejąca implementacja tego mechanizmu, tj. wtyczka firmy Adobe, jest nieoptymalna, a tworzenie uliniowionych dokumentów jest skomplikowane, m.in. z powodów nieścisłości w specyfikacji (zobacz [4, “Linearisation”]).

1.3.2. Rozdzielone dokumenty wielostronicowe

W formacie *rozdzielonym* (ang. *indirect*), każdy plik składowy znajduje się w osobnym pliku dyskowym; oprócz tego, *plik indeksowy* (ang. *index file*) zawiera katalog plików skła-

dowych i inne pomocnicze informacje. Dzięki temu, swobodny dostęp do stron daje się zaimplementować w oczywisty sposób, bez nakładania dodatkowych wymagań na serwer WWW.

Przeglądarki dokumentów DjVu oferują jednolity sposób dostępu do obu formatów, tak że użytkownik może dowiedzieć się, z którym z nich ma do czynienia, może jedynie explicite prosząc o podanie tej informacji.

1.4. Dane nierastrowe

1.4.1. Adnotacje

Do każdej strony dokumentu DjVu mogą być dołączone *adnotacje* (ang. *annotations*), niosące pewne dodatkowe informacje na jej temat.

1.4.1.0. Ogólne uwagi dotyczące składni

Kolory kodowane są w postaci `#`rgb, gdzie r, g, b są, zapisanymi dwoma szesnastkowymi cyframi, liczbami wyznaczającymi kolor w przestrzeni RGB[2]. Napisy kodowane są poprzez otoczenie ich z obu stron znakami `"`; szczegóły i wątpliwości wyjaśnione są w 1.6.1.

1.4.1.1. Interfejs przeglądarki

Za pomocą adnotacji można określić pewne szczegóły interfejsu użytkownika przeglądarki:

- kolor k obszaru wokół obrazu strony — `(background` k`)`;
- początkowe powiększenie obrazu strony — `(zoom` z`)`:
  - jeśli $z =$ `stretch`, to obraz strony będzie zajmował całe okno, bez zachowania proporcji szerokość/wysokość,
  - jeśli $z =$ `page`, to obraz strony będzie zajmował możliwie całe okno, z zachowaniem proporcji szerokość/wysokość,
  - jeśli $z =$ `width`, to obraz strony będzie zajmował całą szerokość okna, z zachowaniem proporcji szerokość/wysokość,
  - jeśli $z =$ `one2one`, to jeden piksel obrazu strony będzie odpowiadał jednemu pikselowi ekranu,
  - jeśli $z =$ `d`p, gdzie $p \in \mathbb{N}, 1 \leqslant n \leqslant 999$, to obraz będzie wielkości $p\%$ naturalnego rozmiaru strony;
- początkowy sposób wyświetlania obrazu rastrowego — `(mode` m`)`:
  - jeśli $m =$ `color`, to wyświetlane będą pierwszy plan i tło,
  - jeśli $m =$ `fore`, to wyświetlany będzie jedynie pierwszy plan,
  - jeśli $m =$ `back`, to wyświetlane będzie jedynie tło,
  - jeśli $m =$ `bw`, to wyświetlana będzie jedynie maska pierwszego planu;
- położenie obrazu strony względem okna przeglądarki — `(align` h v`)`, gdzie:
  - jeśli $h =$ `left`, `center`, `right` to obraz będzie wyrównany w poziomie, odpowiedno: do lewej strony, środka lub prawej strony,

[2]Na przykład `#ff0000` oznacza w pełni nasycony kolor czerwony.

– jeśli v = `top`, `center`, `bottom` to obraz będzie wyrównany w pionie, odpowiedno: do góry, środka lub dołu.

1.4.1.2. Adnotacje nanoszone

Wzbogacić stronę o informacje na temat pewnej konkretnej jej części można za pomocą *adnotacji nanoszonych* (ang. *overprinted annotations*). Zamiennie używa się również terminu *hiperłącze* (ang. *maparea* lub po prostu *hyperlink*).

Adnotacje nanoszone mają postać `(maparea` u c a p_1 p_2 ...`)`.

Jeżeli u jest napisem, to u określa adres docelowy; jeżeli u = `(url` u' t`)`, to napis u' określa adres docelowy a t docelową ramkę. Adres docelowy jest adresem internetowym (jego składnię definiuje [11], zobacz też 1.6.2) lub wskazuje na pewną stronę[3] bieżącego dokumentu:

- jeżeli jest postaci `#`i, gdzie i identyfikatorem pewnej strony — na stronę o tym identyfikatorze;
- jeżeli jest postaci `#+`n lub `#-`n, gdzie $n \in \mathbb{N}$ — na stronę o n późniejszą lub wcześniejszą;
- jeżeli jest postaci `#`n, gdzie $n \in \mathbb{N}$ — na n-tą stronę.

Napis t określa docelową ramkę, w której ma być otwarty dokument w przypadku gdy jest osadzony w HTML-u, wg składni [12, "Frame target names"], tj.:

- jeżeli t = `_blank` — w nowym oknie;
- jeżeli t = `_self` — w bieżącej ramce;
- jeżeli t = `_parent` — w nadrzędnej ramce (lub bieżącej, jeżeli nie ma nadrzędnej);
- jeżeli t = `_top` — w bieżącym oknie.
- w przeciwnym przypadku — w ramce o nazwie t.

Napis c to komentarz, który może być wyświetlony przez przeglądarkę gdy kursor myszy znajduje się nad adnotacją.

a określa kształt i położenie adnotacji na stronie:

- `(rect` x y w h`)` — prostokąt o lewym-dolnym rogu w punkcie (x, y), szerokości w, wysokości h;
- `(oval` x y w h`)` — elipsa wpisana w prostokąt o lewym-dolnym rogu w punkcie (x, y), szerokości w, wysokości h;
- `(text` x y w h`)` — tekst wewnątrz prostokąta o lewym-dolnym rogu w punkcie (x, y), szerokości w, wysokości h;
- `(line` x_0 y_0 x_1 y_1`)` — odcinek o końcach w punktach (x_0, y_0) i (x_1, y_1);
- `(poly` x_0 y_0 x_1 y_1 ...`)` — wielokąt o wierzchołkach w punktach (x_i, y_i).

Przyjmuje się, że lewy-dolny róg strony ma współrzędne $(0, 0)$ a współrzędne rosną w górę i w prawo.

Rozmaitych dodatkowych efektów można zażądać przy pomocy parametrów p_i:

[3] Sposób wyboru strony staje się bardziej skomplikowany, gdy dokument zawiera strony o identyfikatorach, tytułach lub nazwach numerycznych. Zostało to wyjaśnione w [3, "Page References in `maparea` Links"].

- braku ramki, jeśli $p_i =$ `(none)`; przerywanej czarno-białej ramki jeśli $p_i =$ `(xor)` lub ramki o jednolitym kolorze k, jeśli $p_i =$ `(border` k`)`;
- pseudotrójwymiarowej ramki grubości t pikseli, jeśli $p_i =$ `(shadow_`q t`)`, gdzie $q \in \{$`in`, `out`, `ein`, `eout`$\}$, $t \in \mathbb{N}$, $1 \leqslant t \leqslant 32$ (tylko dla adnotacji o kształcie prostokąta);
- wyświetlania ramki zawsze, a nie tylko, gdy kursor myszy znajdowałby się nad hiperłączem, jeśli $p_i =$ `(border_avis)` (tylko dla adnotacji o kształcie prostokąta, elipsy lub wielokąta);
- podświetlenia kolorem k, jeśli $p_i =$ `(hilite` k`)`; przeźroczystości podświetlenia p%, jeśli $p_i =$ `(opacity` p`)` (tylko dla adnotacji o kształcie prostokąta);
- strzałki na końcu odcinka, jeśli $p_i =$ `(arrow)`; jego grubości w, jeśli $p_i =$ `(width` w`)`; jego koloru k jeśli $p_i =$ `(lineclr` k`)`;
- koloru tła tekstu k, jeśli $p_i =$ `(backclr` k`)`; koloru tego tekstu k, jeśli jeśli $p_i =$ `(textclr` k`)`.
- tego, by zamiast tekstu wyświetlana była tylko „pinezka", a sam tekst pokazywał się dopiero po jej kliknięciu, jeśli $p_i =$ `(pushpin)`.

Graficznym edytorem m.in. adnotacji nanoszonych w dokumentach DjVu jest, powstały w ramach niniejszej pracy, program *DjVuSmooth* (zobacz 3.6).

Zestaw adnotacji nanoszonych dostępnych w formacie PDF jest szerszy (zobacz [6, s. 604–647]) niż w DjVu. Podobnie, znacznie szerszy jest wachlarz dostępnych *akcji* wywoływanych po kliknięciu hiperłącza (zobacz [6, s. 647–671]) — w DjVu brakuje zwłaszcza możliwości przejścia do jakiegoś *fragmentu* strony.

1.4.1.3. Metadane

Możliwość zawierania *metadanych* to nowa, nieujęta jeszcze w żadnym opublikowanym standardzie, cecha dokumentów DjVu (zobacz. [3, "Metadata Annotations"]). Metadane to skończona funkcja M ze zbioru symboli (tj. ciągów znaków o umownym znaczeniu) w zbiór napisów, o następującej reprezentacji: `(metadata (`k_0 v_0`) (`k_1 v_1`) ...)`, gdzie $M(k_i) = v_i$.

Metadane, podobnie jak wszystkie innego rodzaju adnotacji, muszą być związane z konkretną stroną. Tymczasem wydaje się, że bardziej użyteczna byłaby możliwość specyfikowania metadanych *całego* dokumentu. Aby obejść ten problem, można umieścić tego typu adnotacje w pliku adnotacji dzielonych, włączanych przez wszystkie strony. Należy oczekiwać, że w przyszłych wersjach standardu, zostanie on rozwiązany w inny sposób.

Nie ma ustalonego repertuaru kluczy, wszakże dwa ich zestawy zostały uznane za *godne uwagi*:

- pochodzące z narzędzia formatującego bibliografię BibTeX, tj. `address`, `annote`, `author`, `booktitle`, `chapter`, `crossref`, `edition`, `editor`, `howpublished`, `institution`, `journal`, `key`, `month`, `note`, `number`, `organization`, `pages`, `publisher`, `school`, `series`, `title`, `type`, `volume`, `year` (lista sporządzona na podstawie [13, s. 9–11]);
- zapożyczone ze specyfikacji PDF — zobacz 3.3.4.4.1.

1.4.2. Ukryty tekst

Każda strona może zawierać warstwę tekstową, która z obszarami obrazu rastrowego wiąże odpowiadający im tekst. Dzięki temu możliwe jest indeksowanie dokumentów, przeszukiwanie ich, czy kopiowanie tekstu z zaznaczonego fragmentu obrazu do schowka. Ręczne wprowadzanie danych tekstowych jest zazwyczaj zbyt żmudne, dlatego najczęściej warstwa tekstowa pochodzi albo z dokumentu elektronicznego albo jest wynikiem OCR-u.

Automatyczne odtwarzanie warstwy tekstowej i jej ręczną korektę umożliwiają programy powstałe w ramach tej pracy: *pdf2djvu* (zobacz 3.3), *ocrodjvu* (zobacz 3.5) oraz *DjVuSmooth* (zobacz 3.6).

Koncepcyjnie, warstwa tekstowa to uporządkowane ukorzenione drzewo, w którego węzłami są strefy tekstu. Każda ze stref:

- ma jeden z 7 liniowo uporządkowanych typów: strona > łam > region > akapit > linia > słowo > znak; typ ten jest większy niż typ każdego syna;
- obejmuje obszar pewnego prostokąta o krawędziach równoległych do krawędzi strony; obszar ten zawiera wszystkie obszary synów, które z kolei są parami rozłączne;
- jeżeli jest liściem, jest etykietowana tekstem.

Kolejność synów w drzewie odpowiada kolejności czytania.

1.4.3. Konspekt

Wielostronicowy dokument DjVu może zawierać *konspekt* (ang. *document outline*)[4], czyli elektroniczny spis treści, dający użytkownikom szybki dostęp do różnych części dokumentu.

Koncepcyjnie, konspekt to uporządkowane ukorzenione drzewo, którego każdy węzeł nie będący korzeniem etykietowany jest tekstowym opisem i adresem. Składnia i siła wyrazu adresów jest taka sama jak w przypadku adnotacji nanoszonych.

Niekiedy[5] zamiast terminu *konspekt* spotyka się termin *zakładki* (ang. *bookmarks*). Nie ma to jednak uzasadnienia w specyfikacji formatu (zobacz [2, s. 4, 14–13]). Co więcej, może to być mylące: *zakładki* w przeglądarkach internetowych tworzy i modyfikuje użytkownik; książkę zakłada jej czytelnik; natomiast *konspekt* jest *częścią* dokumentu DjVu. Tę samą terminologię stosuje się w przypadku formatu PDF (zobacz [6, s. 584–587]).

1.5. DjVu w Internecie

Oficjalnym *Internet media type* (identyfikatorem formatów plików w Internecie, czyli tzw. *typem MIME*) dla dokumentów DjVu jest od 2002 roku `image/vnd.djvu`[6]. Przed tą datą używane były niestandardowe `image/x.djvu` i `image/x-djvu`. Należy unikać ich stosowania, ze świadomością jednak, że mogą być nadal spotykane.

Dokumenty DjVu można udostępniać w Internecie w zasadzie na dwa sposoby: podanie adresu lub osadzenie w stronie HTML.

1.5.1. Adresy dokumentów

Adresem dokumentu DjVu jest po prostu adres pliku lub — w przypadku rozdzielonego dokumentu wielostronicowego — adres pliku indeksowego. Jeżeli *zapytanie* (ang. *query*,

[4] Takie tłumaczenie na język polski zaproponował Janusz S. Bień.

[5] Przykładami są: strona podręcznika programu *djvused*, interfejs programu *DjVu Viewer* firmy Lizardtech.

[6] Zobacz `http://www.iana.org/assignments/media-types/image/vnd-djvu`

```
...
<object
  data="http://www.mimuw.edu.pl/polszczyzna/SGKP/SG06.djvu"
  type="image/x-djvu" width="100%" height="99%"
>
  <param name="page" value="tom06-399.djvu">
  <param name="zoom" value="width">
  Przeglądarka nie obsługuje formatu DjVu.
</object>
...
```

Listing 1.1: Przykładowy fragment pliku HTML, w którym został osadzony dokument DjVu.

zobacz [11, "Syntax Components"]) w adresie ma postać `djvuopts&`p lub q`&djvuopts&`p to p jest interpretowane jako ciąg oddzielonych znakiem `&` ustawień przeglądarki dokumentów DjVu. Zestaw dostępnych ustawień nie jest ustandaryzowany, ale najpopularniejsze z nich i działające we wszystkich znaczących[7] wtyczkach do przeglądarek internetowych to:

- `zoom=`z, gdzie $z \in \left\{ \texttt{stretch}, \texttt{page}, \texttt{width}, \texttt{one2one} \right\}$ lub $z \in \mathbb{N}, 1 \leqslant z \leqslant 999$ — ustawia początkowe powiększenie obrazu strony, analogicznie jak adnotacja `(zoom ...)`;
- `page=`n — wybiera stronę o numerze lub identyfikatorze[8] n;
- `highlight=`x`,`y`,`w`,`h lub `highlight=`x`,`y`,`w`,`h`,`k powoduje wyróżnienie fragmentu strony, analogiczne do adnotacji nanoszonej o kształcie `(rect` x y w h`)` z efektem `(hilite #`k`)`.

1.5.2. Osadzanie DjVu w HTML

Dokument DjVu można osadzić w dokumencie HTML przy pomocy elementu `OBJECT` a dodatkowe opcje przekazać przy pomocy elementu `PARAM` (zobacz [12, "Generic inclusion: the `OBJECT` element"]). Metodę tę ilustruje na przykładzie listing 1.1.

1.6. Nieścisłości w specyfikacji

1.6.1. Sekwencje specjalne w adnotacjach

Specyfikacja DjVu przewiduje, że napisy są otaczane znakami `"`, a jedyną sekwencją specjalną (ang. *escape sequence*)[9] jest `\"`, reprezentującą znak `"` (zobacz [2, s. 16]). Oznacza to w szczególności, że *żadnej* reprezentacji nie mają napisy zawierające `\"` ani kończące się na `\`. Wydaje się więc, że przyjęcie takiego schematu reprezentacji napisów jest wynikiem niedopatrzenia.

DjVuLibre stosuje inny schemat (zobacz [3, "Escape Sequences in Annotation Chunk Strings"]), podobny do znanego z języka C. Sekwencjami specjalnymi są w nim `\a`, `\b`,

[7] *djview3* i *djview4* z pakietu *DjVuLibre* oraz *DjVu Browser Plug-in* firmy LizardTech.

[8] Sposób wyboru strony staje się bardziej skomplikowany, gdy dokument zawiera strony o identyfikatorach, tytułach lub nazwach numerycznych. Zostało to wyjaśnione w [3, "DjVu CGI Argument `page=`"]

[9] Takie tłumaczenie przyjęto za Danutą i Markiem Kruszewskimi — zobacz [14, s. 26].

napis	reprezentacja wg specyfikacji	reprezentacja w *DjVuLibre*
`"`	`"\""`	`"\""`
`\`	—	`"\\"`
`\"`	—	`"\\\""`
`\x`	`"\x"`	`"\\x"`
`\\`	`"\\"`	`"\\\\"`
`x↩x`	`"x↩x"`	`"x\nx"`
`ξ`	`"ξ"`	`"\316\276"`

Tabela 1.1: Przykładowe napisy i ich reprezentacje wg „starych" zasad ze specyfikacji i wg „nowych" zasad z biblioteki *DjVuLibre*.

`\f`, `\n`, `\r`, `\t`, `\v`, `\\`, `\"` i `\`o, gdzie o jest niepustym ciągiem co najwyżej 3 cyfr ósemkowych.[10] Aby zredukować problemy ze wsteczną zgodnością, jeśli analiza składniowa reprezentacji napisu nie powiedzie się z powodu napotkania nieznanej sekwencji rozpoczynającej się od `\`, stosowane są reguły ze specyfikacji.

Różnice między schematami ilustruje tabela 1.1.

1.6.2. Adresy internetowe w adnotacjach

Specyfikacja DjVu zawiera zastrzeżenie (zobacz [2, s. 14, 16]), że adresy internetowe mają nie być poddane *kodowaniu procentowemu* (ang. *percent-encoding*). Kodowanie to pozwala reprezentować znaki, które mają w adresach specjalne znaczenie lub których użycie byłoby nielegalne, np. spoza ASCII (zobacz [11, s. 12]). Ponieważ niektórych adresów nie da się zapisać bez stosowania kodowania procentowego, należy uznać, że wspomniane zastrzeżenie znalazło się w specyfikacji w wyniku pomyłki.

1.6.3. Współczynnik podpróbkowania (redukcji rozdzielczości)

Specyfikacja DjVu w wersji 2 wymaga by dla dokumentów wielowarstwowych, dla każdej warstwy IW44 spełnione były następujące więzy:

$$\begin{aligned} w &= \left\lceil \frac{W}{i} \right\rceil \quad \text{dla pewnego } i \in \{1, 2, \ldots, 12\}\,, \\ h &= \left\lceil \frac{H}{j} \right\rceil \quad \text{dla pewnego } j \in \{1, 2, \ldots, 12\}\,, \end{aligned} \tag{S1a}$$

$$\left\lceil \frac{W}{w} \right\rceil = \left\lceil \frac{H}{h} \right\rceil, \tag{S1b}$$

gdzie $W \times H$ to rozmiar obrazu a $w \times h$ rozmiar warstwy (zobacz [1, s. 11–12]). Specyfikacja DjVu w wersji 3 powtarza dosłownie zapis z poprzedniej wersji (zobacz [2, s. 33]), ale w innym,

[10] Definicja języka C przewiduje jeszcze `\?` i `\'`, `\x`x, gdzie x to 2 cyfry szesnastkowe (zobacz [14, s. 64]).

mniej formalnym fragmencie, wspomniane więzy opisane są w następujący sposób:

$$\begin{aligned} w &= \left\lceil \frac{W}{k} \right\rceil, \\ h &= \left\lceil \frac{H}{k} \right\rceil \end{aligned} \quad \text{dla pewnego } k \in \{1, 2, \ldots, 12\}. \tag{S2}$$

(zobacz [2, s. 6]). Zapisy te, choć podobne, nie są równoważne:

- Z S1a i S1b wynika S2 (fakt 1.1).
- Z S2 wynika S1a, ale niekoniecznie S1b (fakty 1.2 i 1.3).

W rzeczywistości, biblioteka referencyjna „od zawsze" sprawdzała jedynie warunek S2, a liczba k ze wzoru jest nazywana w dokumentacji *współczynnikiem podpróbkowania* (ang. *subsampling ratio* lub *subsampling factor*). Zbyt restrykcyjne warunki S1a i S1b zostały wprowadzone do specyfikacji omyłkowo.

Opisana tu nieścisłość została zauważona przez autora niniejszej pracy i zgłoszona (zobacz A.22), czego skutkiem było udokumentowanie jej w [3, "Background Reduction Ratio"].

Usterka ta ma niewielkie znaczenie praktyczne, bowiem:

- większość oprogramowania korzysta przy przetwarzaniu plików DjVu z *DjVuLibre*;
- alternatywna implementacja, *JavaDjVu* (zobacz 2.4.1), stosuje te same zasady co *DjVuLibre*;
- na mocy faktu 1.2, oba zapisy są równoważne o ile $W, H > 121$.

Fakt 1.1. *Jeżeli* $\mathbb{N} \ni x, j \geqslant 1$ *oraz:*

$$\left\lceil \frac{x}{\left\lceil \frac{x}{j} \right\rceil} \right\rceil = k,$$

to:

$$\left\lceil \frac{x}{j} \right\rceil = \left\lceil \frac{x}{k} \right\rceil.$$

Dowód. Liczbę x możemy zapisać w postaci $x = aj - b$, gdzie $a = \left\lceil \frac{x}{j} \right\rceil \geqslant 1$, $b < j$. Wówczas:

$$k = \left\lceil \frac{x}{\left\lceil \frac{x}{j} \right\rceil} \right\rceil = \left\lceil \frac{aj - b}{a} \right\rceil = j - \left\lfloor \frac{b}{a} \right\rfloor. \tag{1.1}$$

Oczywisty układ nierówności:

$$\left\lfloor \frac{b}{a} \right\rfloor \leqslant a \left\lfloor \frac{b}{a} \right\rfloor \leqslant b$$

przekształcamy do postaci:

$$\left\lfloor \frac{b}{a} \right\rfloor - b \leqslant a \left\lfloor \frac{b}{a} \right\rfloor - b \leqslant 0,$$

z której wynika, że:

$$\left\lfloor \frac{b}{a} \right\rfloor - j < a \left\lfloor \frac{b}{a} \right\rfloor - b \leqslant 0.$$

Podstawiając równość 1.1 otrzymujemy:

$$-k < a \left\lfloor \frac{b}{a} \right\rfloor - b \leqslant 0,$$
$$-1 < \frac{a \left\lfloor \frac{b}{a} \right\rfloor - b}{k} \leqslant 0,$$

czyli:

$$0 = \left\lceil \frac{a \left\lfloor \frac{b}{a} \right\rfloor - b}{k} \right\rceil,$$
$$a = \left\lceil \frac{ak + a \left\lfloor \frac{b}{a} \right\rfloor - b}{k} \right\rceil = \left\lceil \frac{aj - b}{k} \right\rceil = \left\lceil \frac{x}{k} \right\rceil.$$

Zatem:

$$\left\lceil \frac{x}{j} \right\rceil = a = \left\lceil \frac{x}{k} \right\rceil.$$

□

Fakt 1.2. *Jeżeli* $\mathbb{N} \ni x, n \geqslant 1$ *oraz* $x > (n-1)^2$ *to:*

$$\left\lceil \frac{x}{\left\lceil \frac{x}{n} \right\rceil} \right\rceil = n.$$

Dowód. Jeżeli $x \geqslant n^2 - 1$ to:

$$\frac{x}{\left\lceil \frac{x}{n} \right\rceil} > \frac{x}{\frac{x}{n} + 1} = \frac{xn}{x+n} \geqslant \frac{xn - x + n^2 - 1}{x+n} = \frac{(x+n)(n-1)}{x+n} = n - 1. \tag{1.2}$$

W przeciwnym przypadku, mamy $(n-1)^2 < x < n^2 - 1$, więc:

$$\frac{x}{\left\lceil \frac{x}{n} \right\rceil} > \frac{(n-1)^2}{\left\lceil \frac{x}{n} \right\rceil} > \frac{(n-1)^2}{\left\lceil \frac{n^2-1}{n} \right\rceil} = \frac{(n-1)^2}{\left\lceil n - \frac{1}{n} \right\rceil} = \frac{(n-1)^2}{n-1} = n - 1. \tag{1.3}$$

Zatem, z nierówności 1.2 i 1.3:

$$n - 1 < \frac{x}{\left\lceil \frac{x}{n} \right\rceil} \leqslant \frac{x}{\frac{x}{n}} = n.$$

□

Fakt 1.3. *Jeżeli* $\mathbb{N} \ni n \geqslant 1$ *oraz* $x = (n-1)^2$ *to:*

$$\left\lceil \frac{x}{\left\lceil \frac{x}{n} \right\rceil} \right\rceil = n - 1.$$

Dowód.

$$\frac{x}{\left\lceil \frac{x}{n} \right\rceil} = \frac{(n-1)^2}{\left\lceil \frac{n^2-2n+1}{n} \right\rceil} = \frac{(n-1)^2}{\left\lceil n - 2 + \frac{1}{n} \right\rceil} = \frac{(n-1)^2}{n-1} = n - 1.$$

□

Rozdział 2

DjVuLibre

2.1. Wprowadzenie

DjVuLibre to swobodny pakiet oprogramowania obsługującego dokumenty w formacie DjVu. W jego skład wchodzą:

- biblioteka dzielona,
- narzędzia linii poleceń,
- przeglądarki dokumentów,
- wtyczka do przeglądarek internetowych.

Oprogramowanie jest udostępniane na zasadach Powszechnej Licencji Publicznej GNU (GPL) w wersji 2. Witryna internetowa przedsięwzięcia to `http://djvu.sf.net`.

2.1.1. Historia

W 1998 roku grupa programistów pod kierownictwem Yanna LeCunna i Léona Bottou stworzyła i udostępniła za darmo dla użytku niekomercyjnego pierwsze oprogramowanie do obsługi DjVu: prototypowe kompresory i wtyczki do przeglądarek.

W 1999 roku firma AT&T udostępniła, na zasadach swobodnej licencji *AT&T Source Code License*, wersję 2.0 biblioteki referencyjnej DjVu (*DjVu Reference Library*), autorstwa głównie Léona Bottou.

W marcu 2000 roku firma LizardTech, razem z całą technologią DjVu, nabyła bibliotekę referencyjną w wersji 3.0 (obsługującą DjVu w wersji 3).

W październiku 2002 roku LizardTech udostępnił bibliotekę referencyjną w wersji 2.2 na licencji GNU GPL; następnie udostępniony został również kod wersji 3.0 i 3.5.

Uwolniony kod był trudny do kompilacji, instalacji, przenoszenia na inne platformy, czy nawet zrozumienia. *DjVuLibre* jest wynikiem uporządkowania kodu *DjVuReference Library* w wersji 3.5, dzięki pracy Léona Bottou i innych.

2.1.2. Ochrona patentowa

Zarówno implementacja kodowania arytmetycznego i kompresji falkowej są przedmiotem kilku patentów. Posiadaczem większości z nich jest AT&T, ale LizardTech ma do nich szerokie prawa, m.in. licencjonowania.

Wersja 3.5 biblioteki referencyjnej została udostępniona razem z licencją patentową dla użytkowników, ale dotyczyła ona, wedle niektórych interpretacji, jedynie *oryginalnego kodu*.

W 2002 roku LizardTech jednoznacznie wyraził zgodę, by kod wywodzący się z biblioteki referencyjnej naruszał patenty tak dalece, jak robił to oryginalny jej kod.

2.2. Biblioteka dzielona

2.2.1. Część publiczna

Niewielki fragment biblioteki dzielonej *DjVuLibre* ma publiczny interfejs programisty (API) dla języka C. Fragmenty te są opisane w kolejnych podrozdziałach.

W ramach niniejszej pracy magisterskiej powstały dowiązania publicznej części biblioteki dla języka Python — zobacz 3.4.

2.2.1.1. *MiniExp*

MiniExp jest fragmentem biblioteki zawierającym funkcje umożliwiające tworzenie i modyfikację danych semistrukturalnych, a także operacje wejścia/wyjścia dla ich tekstowych reprezentacji. Wyrażenia (dane) mogą być następujących typów:

- liczby całkowite z przedziału $\left[-2^{29}, 2^{29}\right)$;
- symbole (ciągi znaków o umownym znaczeniu);
- napisy;
- listy puste;
- pary dwóch wyrażeń dowolnych (być może różnych) typów.

Niepuste listy można budować za pomocą par, przyjmując że niepusta lista to para: pierwszy element — ogon listy.

Zewnętrzną reprezentacją tych danych semistrukturalnych są, znane z języka *Lisp*, S-wyrażenia. Dalej, o ile nie będzie to powodowało nieporozumień, zarówno samo wyrażenie (pewną strukturę danych) jak i jej reprezentację (ciąg bajtów) będę nazywał S-wyrażeniem.

S-wyrażenia używane do zapisu adnotacji (zobacz 1.4.1), a do opisu innych danych również w podbibliotece *DDJVU* (zobacz 2.2.1.2) i programie *djvused* (zobacz 2.3.1.11).

S-wyrażenia można uznać jako alternatywę dla XML-a: mogą reprezentować te same dane w sposób bardziej ekonomiczny.

Opis interfejsu podbiblioteki znajduje się w pliku nagłówkowym `libdjvu/miniexp.h`.

2.2.1.1.1. Ogólna składnia S-wyrażeń

Liczby całkowite zapisywane są w systemie dziesiętnym, lub poprzedzone zerem w systemie ósemkowym; np. liczbę 42 można zapisać jako `42` lub `052`.

Składnia napisów została opisana w 1.6.1.

Symbole, o ile nie zawierają białych znaków lub symboli o specjalnym znaczeniu, zapisuje się po prostu podając ich nazwę. Inne symbole nie mają w zasadzie zastosowania w DjVu.

`(a . b)` reprezentuje parę wyrażeń o reprezentacjach a i b. $(a_0\ a_1\ \ldots\ a_k)$ reprezentuje k-elementową (być może pustą) listę złożoną z elementów o reprezentacjach a_0, a_1, ..., a_k.

2.2.1.1.2. Adnotacje

Składnia adnotacji została opisana w 1.4.1.

```
(bookmarks
  ("Wprowadzenie" "#9")
  ("Format␣DjVu" "#11"
    ("Wprowadzenie" "#11")
    ("Obrazy␣rastrowe" "#11"
      ...
    )
    ...
  )
  ...
  ("Bibliografia" "#71")
)
```

Listing 2.1: Fragment przykładowego S-wyrażenia reprezentującego konspekt dokumentu.

2.2.1.1.3. Konspekt

S-wyrażenie reprezentujące kontekst to lista, której pierwszym elementem jest symbol `bookmarks`, a kolejnymi — reprezentację pozycji konspektu najwyższego szczebla.

Reprezentacja każdej pozycji konspektu to lista, której pierwszym elementem jest tytuł (napis), drugim adres (napis), a kolejnymi — reprezentacje pozycji podrzędnych.

Przykładowe S-wyrażenie reprezentujące konspekt przedstawione jest na listingu 2.1.

2.2.1.1.4. Warstwa tekstowa

S-wyrażenie reprezentujące strefę tekstu to lista, której pierwszy element określa typ strefy, cztery kolejne to współrzędnego lewego-dolnego i prawego-górnego rogu strefy, kolejne — reprezentacje stref podrzędnych.

Typ określa się za pomocą jednego z następujących symboli:

- `page` — strona;
- `column` — łam;
- `region` — region;
- `para` — akapit;
- `line` — linia;
- `word` — słowo;
- `char` — znak.

Przyjmuje się, że lewy-dolny róg strony ma współrzędne $(0, 0)$ a współrzędne rosną w górę i w prawo.

Przykładowe S-wyrażenie reprezentujące konspekt przedstawione jest na listingu 2.2.

2.2.1.2. *DDJVU*

Podbiblioteka *DDJVU* zawiera wszystkie funkcje niezbędne do zaimplementowania przeglądarki dokumentów DjVu.

DDJVU nie korzysta bezpośrednio z żadnych protokołów sieciowych; za dostarczenie danych spoza plików lokalnych jest odpowiedzialny użytkownik.

```
(page 0 0 2480 3508
  ...
  (paragraph 366 2064 2184 2161
    ...
    (line 426 2120 2184 2161
      (word 426 2121 658 2161 "S-wyra\305\274enie")
      (word 682 2120 965 2160 "reprezentuj\304\205ce")
      ...
    )
    ...
  )
  ...
)
```

Listing 2.2: Fragment przykładowego S-wyrażenia reprezentującego warstwę tekstową dokumentu.

Wiele kluczowych funkcji ma asynchroniczną naturę — ich wywołanie nie powoduje natychmiastowego wykonania zleconej operacji, a jedynie inicjację *zadania* (ang. *jobs*). Informacje o zmianach *statusu* zadania (nie rozpoczęte, rozpoczęte, zakończone sukcesem, zakończone błędem, zatrzymane przez użytkownika) są przekazywane do programu korzystającego z *DDJVU* poprzez *komunikaty* (ang. *messages*), które można pobrać z *kolejki zdarzeń* (ang. *event queue*). Komunikaty wspomagają też stosowanie funkcji, których wynik jest natychmiastowy: niosą informacje o błędach czy możliwości ponownego wywołania funkcji (gdy pobrano i zdekodowano więcej danych niż przy poprzednio).

Opis interfejsu podbiblioteki znajduje się w pliku nagłówkowym `libdjvu/ddjvuapi.h`.

2.2.2. Część prywatna

Większość biblioteki dzielonej nie ma publicznego interfejsu. W praktyce oznacza to, że do funkcji, takich jak:

- tworzenie od podstaw nowych dokumentów (kompresja),
- modyfikacja istniejących dokumentów,
- dostęp do niskopoziomowych struktur danych (np. słowników kształtów),

bezpośrednio mają jedynie programy będące częścią *DjVuLibre*. Skorzystanie z bogactwa tej części w nowych programach jest utrudnione, bo w zasadzie oznacza konieczność zmian w kodzie pakietu.

2.3. Programy

2.3.1. Narzędzia linii poleceń

2.3.1.1. *cjb2*

Konwersję plików czarno-białych w formacie PBM lub TIFF do DjVu (JB2) można przeprowadzić za pomocą programu *cjb2*. Kompresja może być bezstratna lub stratna, przy czym agresywność kompresji stratnej jest konfigurowalne. Mniejszy rozmiar pliku wynikowego jest osiągany przy zastosowaniu dwóch technik:

- usuwanie z obrazu bardzo małych elementów, które najczęściej są tylko artefaktami procesu skanowania;
- odstępowanie od kodowania różnic pomiędzy kształtami bardzo podobnymi (tj. zastępowanie jednego kształtu — innym, bardzo podobnym).

cjb2 kompresuje na raz dokładnie jedną stronę. Wynika stąd, że nie jest w stanie stworzyć słowników kształtów dzielonych między stronami. Takie możliwości ma *minidjvu* — zobacz 2.4.2.

2.3.1.2. *c44*

Konwersję plików PGM, PPM lub JPEG do DjVu (IW44) można przeprowadzić za pomocą programu *c44*. Kompresja jest prawie zawsze stratna; jakość/wielkość pliku wynikowego można kontrolować na wiele sposobów:

- podając liczbę *plastrów* w każdym *kawałku*[1];
- podając rozmiar każdego kawałka w bitach na piksel, bajtach, lub procentach rozmiaru pliku wyjściowego;
- podając pożądany stosunek sygnału do szumu.

2.3.1.3. *cpaldjvu*

Konwersję plików o niewielkiej liczbie kolorów w formacie PPM do DjVu (kolorowe JB2 + IW44) można przeprowadzić za pomocą programu *cpaldjvu*. Kompresja polega na:

- ewentualnej redukcji liczby kolorów przy pomocy prostego algorytmu,
- zakodowaniu pikseli w kolorze dominującym (lub, opcjonalnie, najjaśniejszym) jako tła w formacie IW44;
- zakodowaniu pozostałych pikseli jako pierwszego planu w formacie JB2 z kolorami.

2.3.1.4. *csepdjvu*

Rozseparowne tło i pierwszy plan potrafi w wielowarstwowy dokument DjVu przekształcić program *csepdjvu*. Dodatkowo, umożliwia on również włączenie do tak powstałego dokumentu: warstwy tekstowej, hiperłączy i konspektu.

Zastosowanie niekonwencjonalnych formatów danych i skomplikowany interfejs tego programu powoduje, że właściwie jedynymi użytkownikami tego programu są *inne programy*, np. *djvudigital* (zobacz 2.3.1.5) czy *pdf2djvu* (zobacz 3.3).

2.3.1.5. *djvudigital*

Konwersję plików w formacie PostScript lub PDF do formatu DjVu może przeprowadzić program *djvudigital*. Zasadnicza część konwersji delegowana jest do programu *gsdjvu*, a jego wyniki składane są w spakowany dokument DjVu przez *csepdjvu*.

Oprócz warstwy graficznej, do dokumentu włączane są: warstwa tekstowa (opcjonalnie), hiperłącza i konspekt.

Konkurencją dla *djvudigital* jest, powstały w ramach niniejszej pracy, program *pdf2djvu* (zobacz 3.3).

[1] Słowa *kawałek* i *plaster* należy rozumieć jak w 1.2.2

2.3.1.6. *any2djvu*

any2djvu to interfejs linii poleceń do publicznego serwisu internetowego `http://any2djvu.djvuzone.org/`, który oferuje możliwość konwersji plików w formatach PostScript, PDF, TIFF, JPEG lub PNM do DjVu.

Oprócz warstwy graficznej, do dokumentu włączane są: warstwa tekstowa powstała w wyniku OCR-u dokumentu (opcjonalnie) i hiperłącza (w przypadku formatu PDF).

Z internetowej natury programu wynika szereg wad:

- nie nadaje się do konwersji dokumentów poufnych;
- wydajność konwersji jest ograniczona przez łącze internetowe;
- jego przyszłość jest niepewna: nie ma gwarancji, że serwis internetowy nie zostanie kiedyś zamknięty.

2.3.1.7. *djvuextract*, *djvumake*

djvuextract potrafi wyodrębnić z dokumentu DjVu poszczególne jego kawałki. *djvumake* z tego rodzaju kawałków złożyć dokument DjVu.

Niskopoziomowy charakter tych programów powoduje, że głównymi ich użytkownikami są *inne programy*, np. *pdf2djvu* (zobacz 3.3).

2.3.1.8. *djvm*, *djvmcvt*

Połączyć wiele jednostronicowych dokumentów DjVu w jeden wielostronicowy można przy pomocy programu *djvm*. Służy on również do dodawania i usuwania stron z istniejących dokumentów.

Konwersję pomiędzy dokumentami spakowanymi a rozdzielonymi można przeprowadzić przy pomocy programu *djvmcvt*.

Wydaje się osobliwe, że operacje, które przeprowadza *djvm*, daje się wykonać tylko na dokumencie spakowanym — wszystkie te operacje dałoby się efektywniej zaimplementować właśnie dla dokumentów rozdzielonych.

2.3.1.9. *ddjvu*, *djvups*

Konwersję dokumentu DjVu (lub jego części) do jednego z konwencjonalnych formatów rastrowych (PBM, PGM, PPM, TIFF) daje się wykonać za pomocą programu *ddjvu*. Istnieje możliwość wyboru stron do konwersji jak i interesującego fragmentu strony.

Konwersję dokumentu DjVu (lub jego niektórych stron) do formatu PostScript może przeprowadzić program *djvups*.

Konwersję do niektórych konwencjonalnych formatów potrafi przeprowadzić również przeglądarka *djview4* (zobacz 2.3.2.2).

2.3.1.10. *djvutoxml*, *djvuxmlparser*

djvutoxml umożliwia eksport *części* danych nierastrowych do pliku XML; po ewentualnej modyfikacji dane te mogą być z powrotem zaimportowane do dokumentu DjVu przy pomocy programu *djvuxmlparser*.

Eksportowane dane to:

- informacja o rozdzielczości;

- informacja o współczynniku gamma;
- dla każdej strony:
  - jej identyfikator;
  - warstwa tekstowa;
  - hiperłącza.

Nie są eksportowane metadane, konspekt ani żadne adnotacje poza hiperłączami. Znane są również problemy z wydajnością eksportu[2]. Z tych powodów, lepszym narzędziem do przeprowadzenia ciągu operacji eksport–edycja–import może być *djvused* (zobacz 2.3.1.11).

2.3.1.11. *djvused*

Jednym ze sposobów edycji danych nierastrowych dokumentu DjVu jest użycie programu *djvused*. Jego funkcje to:

- wypisanie listy plików składowych;
- wypisanie rozmiarów (w pikselach) poszczególnych stron;
- wypisanie wewnętrznej struktury dokumentu (tak jak *djvudump*);
- stworzenie pliku składowego adnotacji dzielonych;
- wypisywanie tekstu: czystego lub w postaci S-wyrażenia;
- zastąpienie warstwy tekstowej — inną, podaną w postaci S-wyrażenia (dla pojedynczej strony);
- usunięcie warstwy tekstowej (dla pojedynczej strony lub całego dokumentu);
- wypisywanie adnotacji;
- zastąpienie adnotacji innymi (dla pojedynczej strony lub dzielonych między stronami);
- usuwanie adnotacji (wszystkich z pojedynczej strony lub całego dokumentu);
- wypisywanie metadanych;
- zastąpenie metadanych innymi;
- wypisywanie konspektu w postaci S-wyrażenia;
- zastąpienie konspektu — innym, podanym w postaci S-wyrażenia;
- wyliczenie i zapisanie miniaturek o podanym rozmiarze;
- usunięcie miniaturek;
- zapisanie pod inną nazwą: całego dokumentu (w formacie spakowanym lub rozdzielonym) albo pojedynczej strony;
- zmiana tytułu strony.

Interfejs programu jest dość skomplikowany — użytkownik musi sporządzić skrypt, którego wykonaniem zajmie się *djvused*.

Część funkcji programu *djvused* opakowuje w graficzny interfejs *DjVuSmooth* (zobacz 3.6), program powstały w ramach niniejszej pracy.

[2]Zobacz `http://sf.net/tracker/?func=detail&aid=1704049&group_id=32953&atid=406583`.

2.3.1.12. *djvutxt*

Wyodrębnienie tekstu, czystego lub w postaci S-wyrażenia, z dokumentu DjVu można przeprowadzić za pomocą programu *djvutxt*.

2.3.1.13. *djvudump*

Program *djvudump* wypisuje, w formie czytelnej dla człowieka, ale dającej się łatwo analizować składniowo, wewnętrzną strukturę dokumentu DjVu.

2.3.1.14. *bzz*

bzz to narzędzie do kompresji, ogólnego przeznaczenia.

2.3.2. Przeglądarki

2.3.2.1. *djview3*

Podstawową przeglądarką dokumentów DjVu był jeszcze do niedawna *djview3* (dawniej nazywany po prostu *djview*), napisana przy użyciu biblioteki *Qt* w wersji 3. Jej podstawowe cechy to:

- niskie wymagania pamięciowe;
- progresywna rastryzacja obrazów;
- efektywne powiększanie/pomniejszanie i przewijanie obrazów;
- możliwość przeszukiwania warstwy tekstowej i kopiowana tekstu do schowka.

djview3 ma szereg niedostatków:

- wymaga uniksowego systemu X11;
- jednocześnie wyświetla co najwyżej jedną stronę dokumentu;
- obsługuje tylko *niektóre* rodzaje adnotacji;
- nie pozwala na wyświetlenie konspektu ani metadanych.

2.3.2.2. *djview4*

Ograniczeń *djview3* nie ma nowa, dostępna od 2007 roku, przeglądarka *djview4*. Została napisana przy użyciu biblioteki *Qt* w wersji 4.

Prócz usunięcia wspomnianych niedostatków, przeglądarka wzbogaciła się o funkcję eksportu do następujących formatów: PDF, TIFF, PostScript, BMP, ICO, JPEG, PNG, PPM, XBM i XPM.

Daje się skompilować w systemach uniksowych, na MacOS X (bez potrzeby instalacji systemu X11) i w Windows.

2.3.2.3. *nsdejavu*

Obie przeglądarki dają się osadzić w przeglądarce internetowej za pomocą wtyczki o nazwie *nsdejavu*. Obsługiwane są uniksowe przeglądarki: *Netscape Navigator*, *Mozilla*, *Galeon*, *Firefox*, *Konquerror* i *Opera*.

2.3.3. *djvuserve*

djvuserve to program, który przy wykorzystaniu protokołu CGI (zobacz [15]), w locie konwertuje spakowane dokumenty DjVu do formatu rozdzielonego.

2.3.4. *gsdjvu*

gsdjvu to sterownik *Ghostscripta*, który analizuje ciąg operacji prowadzący do rastryzacji stron dokumentu elektronicznego i klasyfikuje każdą z nich jako pierwszy plan lub tło. Stanowi podstawę implementacji konwertera *djvudigital*.

Z powodu problemów licencyjnych, formalnie nie jest częścią *DjVuLibre*, ale jego kod jest dostępny na stronach przedsięwzięcia.

2.3.4.1. Licencja

gsdjvu powstał na podstawie kodu udostępnionego w 2005 przez AT&T na licencji Common Public License (CPL). Kod ten użyteczny jest jednak tylko jako część programu *Ghostscript*, który jest dostępny na zasadach dwóch licencji: GPL i Aladdin Free Public License (AFPL). Niestety, CPL nie jest zgodna z żadną z nich. Oznacza to, że nie można dystrybuować *gsdjvu* w pakiecie z pełnym kodem źródłowym ani w postaci binarnej.

Aby móc skorzystać z *gsdjvu*, należy samodzielnie pobrać źródła jego oraz *Ghostscripta* i przeprowadzić ich kompilację. Nie są znane zakończone sukcesem próby kompilacji na systemach nieuniksowych.

2.3.4.2. Algorytm separacji warstw

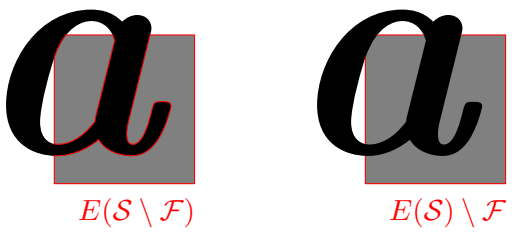
gsdjvu stosuje wyrafinowany algorytm separacji warstw, szczegółowo opisany w [16]. Poniżej przedstawiony jest jego zarys.

Będziemy tu zakładać, że obraz składa się z n jednobarwnych komponentów: $\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_n$. Rastryzacja polega na narysowaniu *po kolei* wszystkich komponentów. Spostrzeżenia:

- koszt zakodowania komponentu jako pierwszego planu jest z grubsza proporcjonalny do obwodu widocznej części tego komponentu (P_c);
- koszt zakodowania komponentu jako tła jest z grubsza proporcjonalny do:
  - długości odcinków brzegu figury nie zasłoniętych przez pierwszy plan (P_b),
  - różnicy kolorów wzdłuż brzegu komponentu (δ).

prowadzą do następującego zachłannego algorytmu:

0. Niech $E(\mathcal{S})$ oznacza brzegi figury $\mathcal{S}$.
1. $T :=$ ustalony przez użytkownika próg.
2. $\mathcal{F} := \varnothing$ (pierwszy plan); $\mathcal{B} := \varnothing$ (tło).
3. Dla $i := n, n-1, \ldots, 1$:
   (a) $\mathcal{S} := \mathcal{S}_i \setminus \mathcal{B}$.
   (b) $P_\mathrm{c} :=$ długość $E(\mathcal{S} \setminus \mathcal{F})$ (zobacz rysunek 2.1).
   (c) $P_\mathrm{b} :=$ długość $E(\mathcal{S}) \setminus \mathcal{F}$ (zobacz rysunek 2.1).
   (d) Oszacuj różnicę kolorów δ wzdłuż $E(\mathcal{S}) \setminus \mathcal{F}$.



Rysunek 2.1: $\mathcal{F}$ = litera $\boldsymbol{a}$, $\mathcal{S}$ = szary prostokąt.

(e) Jeżeli $\delta P_\mathrm{b} < TP_\mathrm{c}$, to $\mathcal{B} := \mathcal{B} \cup (\mathcal{S} \setminus \mathcal{F})$, tj. zaklasyfikuj widoczną część $\mathcal{S}$ do tła.

(f) W przeciwnym przypadku, $\mathcal{F} := \mathcal{F} \cup \mathcal{S}$, tj. zaklasyfikuj widoczną część $\mathcal{S}$ do pierwszego planu.

2.4. Alternatywne implementacje

DjVuLibre to najpopularniejsze, ale nie jedyna swobodna implementacja formatu DjVu.

2.4.1. *JavaDjVu*

Efektem przepisania kodu *DjVuLibre* na język Java jest pakiet *JavaDjVu*. Częścią pakietu jest samodzielna przeglądarka i aplet umożliwiający osadzanie DjVu w HTML-u; brak jest natomiast narzędzi do tworzenia czy modyfikacji dokumentów.

Witryna internetowa przedsięwzięcia to `http://javadjvu.foxtrottechnologies.com/`.

2.4.2. *minidjvu*

Na podstawie *DjVuLibre* powstał bardziej zaawansowany niż *cjb2* kompresor obrazów czarno-białych. Pakiet składa się z biblioteki i dość niskopoziomowego narzędzia linii poleceń. *minidjvu* oferuje dalej idącą (ale dającą się kontrolować) kompresję stratną i słowniki kształtów dzielone pomiędzy stronami.

Witryna internetowa przedsięwzięcia to `http://minidjvu.sf.net/`. Wydaje się, że prace rozwojowe ustały — ostatnie wydanie pochodzi z 2006 roku.

2.5. Podsumowanie

DjVuLibre daje szerokie możliwości tworzenia, modyfikowania i dekodowania dokumentów DjVu. Obsługa tego formatu nie jest jednak kompletna i ustępuje w niektórych aspektach oprogramowaniu licencjonowanemu przez LizardTech. Podstawowe braki w *DjVuLibre* to:

- brak graficznego edytora dokumentów;
- brak *swobodnego* konwertera dokumentów elektronicznych;
- brak narzędzia do separacji warstw obrazów rastrowych;
- brak integracji z narzędziami OCR.

Niektóre z tych niedostatków naprawiają programy powstałe w ramach tej pracy.

Rozdział 3

Nowe oprogramowanie

3.1. Motywacja

W ramach niniejszej pracy powstały programy, które w istotny sposób poszerzają zasób dostępnego swobodnego oprogramowania przetwarzającego dokumenty DjVu. Motywacją za każdym razem był brak wygodnych narzędzi do realizacji niektórych typowych zadań, ani w pakiecie *DjVuLibre*, ani wśród innych programów udostępnianych na swobodnych licencjach.

3.2. Założenia

Autor przyjął, że:

- Oprogramowanie będzie kompilować się i działać co najmniej w nowoczesnych systemach Linux.
- Oprogramowanie będzie dostępne na zasadach Powszechnej Licencji Publicznej GNU (GPL) i nie będzie łączyć kodu z oprogramowaniem niezgodnym z tą licencją.

3.3. *pdf2djvu*

pdf2djvu tworzy dokumenty DjVu z dokumentów w formacie PDF. Potrafi przy tym odtworzyć, oprócz warstwy graficznej: warstwę tekstową, hiperłącza, konspekt i metadane.

Program licencjonowany jest na zasadach Powszechnej Licencji Publicznej GNU w wersji 2. Kod źródłowy, pakiety binarne i dokumentacja udostępnione zostały na witrynie programu: `http://pdf2djvu.googlecode.com/`.

3.3.1. Motywacja

Częścią składową *DjVuLibre* jest *djvudigital* (omówiony w 2.3.1.5), który pozwala na konwersję dokumentów PDF do formatu DjVu. Jednakże w momencie rozpoczęcia prac program ten:

- korzystał z *gsdjvu*, na nieswobodnej licencji;
- nie włączał do dokumentu DjVu:
  - tekstu, jeśli był niewidoczny (jakie to ma znaczenie, jest wyjaśnione w 3.3.4.4.4),

  - hiperłączy,
  - konspektu,
  - metadanych;

W obecnej wersji nie ma już ograniczenia dotyczących hiperłączy i konspektu. Natomiast brak możliwości ekstrakcji metadanych i niewidocznego tekstu wynika z ograniczeń zastosowanego *Ghostscripta* i brak jest perspektyw ich zniesienia.

any2djvu (omówiony w 2.3.1.6), również będący częścią *DjVuLibre*, pozwala na konwersję do DjVu wielu formatów, w tym PDF. Ma on wszakże szereg wad i ograniczeń:

- korzysta z zewnętrznego serwisu internetowego;
- nie włącza do dokumentu DjVu:
  - oryginalnego tekstu,
  - konspektu,
  - metadanych.

Wedle najlepszej wiedzy autora, nie było do tej pory żadnego dostępnego na swobodnej licencji oprogramowania pozwalającego uzyskać dokumenty DjVu z innych, popularnych formatów dokumentów elektronicznych. *pdf2djvu* wypełnia tę lukę.

3.3.2. Założenia

Prócz założeń wyliczonych w 3.2, autor przyjął, że:

- Aby umożliwić korzystanie z programu na komputerach, gdzie w cenie bardziej jest *stabilność* niż *nowość* oprogramowania (np. serwerach), w miarę możliwości obsługiwane będą również stare wersje wykorzystywanych bibliotek. W szegółności program ma kompilować się i działać w Debianie 4.0 (Etch).
- W celu uniknięcia duplikacji kodu, możliwie dużo operacji na plikach PDF, plikach DjVu i na obrazach rastrowych zostanie oddelegowanych do wyspecjalizowanych bibliotek.

3.3.3. Zarys implementacji

3.3.3.1. Obsługa plików PDF

Podstawowym problemem, jaki dotyka *niemal każdy* program w jakiś sposób przetwarzający dokumenty PDF, jest ogromne skomplikowanie formatu. Istotnie, jego specyfikacja ([6]) liczy sobie aż 1310 stron, a i tak nie zawiera kompletnego opisu żadnego z używanych nieoczywistych sposobów kompresji: JPEG, JPEG2000, CCITT (Group 3 lub Group 4), JBIG2, LZW ani Flate. Pisanie *od postaw* kodu analizującego składniowo pliki PDF, wykonującego rastryzację etc. wyłącznie na potrzeby konwertera byłoby zbyt kosztowne i sprzeczne z przyjętymi założeniami.

Naturalnym kandydatem na narzędzie, które pozwoliłoby przetwarzać dokumenty PDF, jest *Ghostscript* — użyto go bowiem do implementacji *djvudigital*. Niestety, użycie go spowodowałoby, że *pdf2djvu* dzieliłby z nim ograniczenia co do metadanych i niewidocznego tekstu. Ponieważ *Ghostscript* nie występuje jako dzielona biblioteka, kompilacja programu wymagałaby jego źródeł. Duplikacja kodu utrudnia również zachowanie bezpieczeństwa oprogramowania, o czym dalej. Doświadczenie pokazuje też, że *Ghostscript* słabo radzi sobie z obsługą plików nie całkiem zgodnych ze specyfikacją PDF, a jednak produkowanych przez rozmaite narzędzia.

W przeszłości często stosowaną metodą uzyskania możliwości operowania na plikach PDF było rozmnożenie (ang. *fork*) kodu *xpdf*[1], popularnej, swobodnej przeglądarki dokumentów tego formatu. Istotnie, wydanie Debian GNU/Linuksa z 2005 roku zawierało aż 9 innych pakietów oprogramowania zawierających kod tej przeglądarki (zobacz [17]). Skorzystanie z takiej metody powoduje niepotrzebnie duży rozmiar kodu źródłowego i plików wykonywalnych, ale — przede wszystkim — niesie za sobą konieczność dodatkowej staranności w kwestii bezpieczeństwa. W kodzie *xpdf* często bowiem były znajdowane poważne błędy, które potencjalnie dały się wykorzystać do wykonania dowolnego kodu na komputerze ofiary, która jedynie otworzyła złośliwie spreparowany plik. Rzeczywiście, w okresie gdy trwała implementacja *pdf2djvu*, odkrytych zostało 5 takich usterek: [18], [19], [20], [21], [22]. Podatne na tego typu błędy najczęściej są również pochodne programy i wszystkie je należy załatać.

Na szczęście, od 2005 roku rozwijana jest, powstała właśnie na bazie *xpdf*, biblioteka *poppler*[2]. Oferuje programiście właściwie to samo co jej protoplasta, bez konieczności duplikacji kodu. W szczególności dało się za jej pomocą nietrudno zaimplementować rastryzację połączoną z separacją warstw, podobnie jak ekstrakcję z dokumentu informacji niegraficznych.

pdf2djvu korzysta z niskopoziomowego fragmentu biblioteki, chętnie zmienianego przez autorów i to w sposób, który jest niezgodny w poprzednimi wersjami. W konsekwencji, ponad 20% kodu *pdf2djvu* stanowi warstwa zgodności z różnymi wydaniami biblioteki. Co więcej, implementacja niektórych funkcji, które da się wyrazić zwięźle korzystając z jednych wersji *popplera*, wymagała w innych skopiowania z biblioteki dużych ilości kodu. W takim wypadku, gdy znaczenie funkcji nie było fundamentalne, zrezygnowano z jej implementacji. W rzeczywistości *pdf2djvu* jest w pełni funkcjonalny jedynie z *popplerem* w wersji $\geqslant 0.7.0$.

3.3.3.2. Język programowania

Użycie biblioteki *poppler*, napisanej w C++, wymusiło implementację *pdf2djvu* w tym samym języku. Istniejące dowiązania dla innych języków programowania (Python[3], Ruby[4]) oferowały dostęp jedynie do fragmentu biblioteki, niewystarczającego do implementacji programu.

3.3.3.3. Operacje na obrazach rastrowych

Wszelkie nietrywialne operacje na obrazach rastrowych są delegowane do biblioteki *GraphicsMagick*. Jest ona rozwidleniem bardziej popularnej biblioteki *ImageMagick*, wspomagającej manipulacje obrazami, w rozwoju której nacisk kładzie się na stabilność interfejsu programistycznego. Dostępna jest na licencji w stylu X11, zgodnej z GPL 2.

3.3.3.4. *DjVuLibre*

pdf2djvu korzysta zarówno z dzielonej biblioteki jak i programów użytkowych będących częścią *DjVuLibre*: *djvused*, *bzz*, *csepdjvu*, *djvuextract*, *djvumake* i *djvm*. Komunikacja między głównym programem a pomocniczymi narzędziami odbywa się z pomocą biblioteki *PStreams*, dostępnej na licencji LGPL 2, zgodnej z GPL 2.

Autor podjął starania, by błędy w *DjVuLibre* nie wpływały negatywnie na działanie *pdf2djvu*. Zostały w tym celu zastosowane obejścia niektórych (opisanych w A.5 i A.15) napotkanych usterek. Dzięki temu, wspierane są niemal wszystkie wersje co najmniej od

[1] `http://www.foolabs.com/xpdf/`
[2] `http://poppler.freedesktop.org/`
[3] PyPoppler: `http://www.gnome.org/~gianmt/`.
[4] Ruby-GNOME2 Project: `http://ruby-gnome2.sourceforge.jp/`.

3.5.17 (z 2006 roku) aż do współczesnych. Prawidłowo nie działają, z powodu błędu opisanego w A.16, wersje 3.5.20-4 ani 3.5.20-5, ani 3.5.20-4 z Debiana.

3.3.3.5. Separacja warstw

Głównym zadaniem konwertera jest separacja warstw (tło — pierwszy plan), tj. zaklasyfikowanie każdego piksela każdej zrastrowanej strony do jednej z warstw. Skutkiem błędnych klasyfikacji może być zarówno niepotrzebne pogorszenie jakości dokumentu wynikowego jaki i jego rozmiaru.

Wyrafinowany algorytm separacji został przedstawiony w 2.3.4.2. W [16, s. 2] zaproponowano następujące naiwne algorytmy:

1. Zaklasyfikuj tekst do pierwszego planu.
2. Zaklasyfikuj wszystkie czarno-białe elementy do pierwszego planu.
3. Zaklasyfikuj elementy rysowane pierwsze do tła.

Prosty algorytm zastosowany w *pdf2djvu* jest wariantem algorytmów 1 i 2. Jego schemat jest następujący:

Dla każdej strony p:

1. Zrastruj w zwykły sposób całą stronę p do bitmapy B_p.
2. Zrastruj całą stronę p do bitmap B'_p, pomijając przy tym rysowanie:
   - tekstu,
   - obrazów 1-bitowych,
   - grafiki wektorowej, z wyjątkiem wypełnień dużych powierzchni.
3. Dla każdego piksela bitmapy B_p o współrzędnych (x, y):
   (a) Jeżeli $B_p[x, y] \neq B'_p[x, y]$ to zaklasyfikuj piksel do pierwszego planu.
   (b) W przeciwnym przypadku, zaklasyfikuj go do tła.

3.3.4. Interfejs użytkownika; przegląd dostępnych funkcji

pdf2djvu udostępnia jedynie interfejs linii poleceń. Proces konwersji nie jest interaktywny, więc taki interfejs jest wystarczający. Dodatkowo, umożliwia to łatwe zautomatyzowanie procesu konwersji wielu plików, bez potrzeby implementacji takiej funkcji w samym programie.

Polecenie `pdf2djvu` jako argumentu wymaga nazwy konwertowanego pliku PDF. Oprócz tego, akceptowane są liczne opcje, opisane poniżej. Dla wygody użytkownika, opcje można umieszczać zarówno przed jak i po nazwie pliku.

3.3.4.1. Typ dokumentu; pliki wyjściowe

Jeśli nie podano inaczej, spakowany dokument DjVu jest wysyłany na standardowe wyjście. Niedozwolone jest w takim wypadku, by standardowym wyjściem był terminal, gdyż wypisanie na niego danych binarnych mogłoby doprowadzić go do nieużywalnego stanu.

Jeśli podano opcję `-o` *f* lub `--output=`*f*, spakowany dokument DjVu jest zapisywany do pliku *f*.

Jeśli podano opcję `-i` *f* lub `--indirect=`*f*, generowany jest rozdzielony dokument DjVu. Wówczas *f* (lub *f*`/index.djvu`, jeżeli *f* byłby katalogiem) jest używany jako

nazwa pliku indeksowego; pliki składowe są umieszczane w tym samym katalogu. Wymaga się, aby katalog ten istniał a użytkownik miał prawo zapisu do niego[5].

Schemat wg którego będą nazywane pliki składowe dokumentu można wybrać za pomocą opcji `--pageid-prefix=`p. Wówczas nazwy będą miały postać pN`.djvu`, gdzie N jest uzupełnionym zerami numerem strony. p może składać się tylko z liter alfabetu łacińskiego, cyfr i znaków: `_` (podkreślenie), `+`, `-` oraz `.`. Jeśli nie podano inaczej, przyjmuje się $p =$ `p`.

3.3.4.2. Rozdzielczość; rozmiar strony

Rozdzielczość, z jaką zostanie zrastrowany dokument można określić opcją `-d` d lub `--dpi=`d. Liczba d musi być całkowita i spełniać nierówność

$$72 \leqslant d \leqslant 6000. \tag{3.1}$$

Ograniczenie to narzuca biblioteka *DjVuLibre*[6].

Dla każdej strony p spełnione są równości:

$$W_p = \left\lfloor 0{,}5 + \frac{\mathcal{W}_p d}{25{,}4\text{ mm}} \right\rfloor, \tag{3.2}$$

$$H_p = \left\lfloor 0{,}5 + \frac{\mathcal{H}_p d}{25{,}4\text{ mm}} \right\rfloor, \tag{3.3}$$

gdzie $W_p \times H_p$ to rozmiar strony p (w pikselach) po zrastrowaniu a $\mathcal{W}_p \times \mathcal{H}_p$ to oryginalny, naturalny rozmiar strony p.

Alternatywnie, rozmiar strony po zrastrowaniu można podać explicite za pomocą opcji `--page-size=`w`x`h. Liczby w, h muszą być całkowite dodatnie. Faktyczny rozmiar strony może zostać zmieniony tak, aby spełnione były 3.1, 3.2 i 3.3.

Jeśli nie podano inaczej, przyjmuje się $d = 300$.

Dla każdej strony PDF zdefiniowane jest 5 regionów wyznaczających, w różnym sensie, granice strony (zobacz [6, s. 962–965]). Najważniejsze z nich to:

- *MediaBox*: prostokąt wyznaczający granice fizycznego medium, na którym strona ma być wyświetlona lub wydrukowana;
- *CropBox*: prostokąt wyznaczający obszar widoczny w przeglądarce lub obszar do którego ma być przycięta strona po wydruku.

W typowych dokumentach wszystkie 5 regionów jest równych.

Jeśli podano opcję `--media-box`, do określenia konwertowanego obszaru jest używany *MediaBox*. W przeciwnym przypadku jest to *CropBox*.

3.3.4.3. Jakość obrazu

3.3.4.3.1. Tło

Jakość tła ustalić można przy pomocy jednej z dwóch opcji: `--bg-slices=`$m_1 + \ldots + m_j$ lub `--bg-slices=`$n_1, \ldots, n_j$. Elementy obu ciągów muszą być całkowite dodatnie a ciąg (n_i) musi być rosnący. Opcja powoduje, że tło każdej strony zostanie zakodowane w formacie

[5] Tj. tworzenia w nim plików

[6] Zobacz też A.19.

IW44, przy użyciu j *kawałków* a i-ty *kawałek* będzie się składał z m_i *plastrów*[7]. Jeśli liczby m_i nie zostały podane explicite, stosuje się wzór:

$$\sum_{x=1}^{i} m_x = n_i.$$

Jeżeli nie podano inaczej, przyjmuje się $j = 4$, $m_1 = 72$, $m_2 = 11$, $m_3 = m_4 = 10$.[8]

Rastrowy rozmiar tła w stosunku do pierwszego planu można określić za pomocą opcji `--bg-subsample=`s. Liczba s musi być całkowita i spełniać nierówność $1 \leqslant s < 12$.[9] Dokładnie, dla każdej strony p, rastrowy rozmiar pierwszego planu $W_p \times H_p$ jest powiązany z rastrowym rozmiarem tła $w_p \times h_p$ wzorami:

$$w_p = \left\lfloor \frac{W_p}{s_p} \right\rfloor, \quad h_p = \left\lfloor \frac{H_p}{s_p} \right\rfloor,$$

gdzie s_p jest największą liczbą taką, że $\mathbb{N} \ni s_p \leqslant s$ oraz

$$\left\lceil \frac{W_p}{w_p} \right\rceil = \left\lceil \frac{H_p}{h_p} \right\rceil$$

(zobacz roważania z 1.6.3). Jeśli nie podano inaczej, przyjmuje się $s = 3$.[10]

Powyższe opcje nie mają zastosowania w sytuacji, gdy tło ma jednolity kolor. Wówczas koduje się je efektywnie w jednym kawałku i minimalnej rozdzielczości lub zupełnie pomija.

3.3.4.3.2. Pierwszy plan

Jeśli nie podano inaczej lub podano opcję `--fg-colors=web`, kolory pierwszego planu redukowane są do tzw. *palety web*. Stosowany jest szybki algorytm odcięcia liniowego: jasność poszczególnych składowych RGB każdego piksela są modyfikowane wg wzoru $x := \frac{1}{5}\left\lfloor \frac{255x+1}{43} \right\rfloor$.

Opcja `--fg-colors=`n powoduje, że kolory pierwszego planu zostaną, przy pomocy procedur zawartych w bibliotece *GraphicsMagick* zredukowane do zoptymalizowanej palety n kolorów. n musi być liczbą całkowitą spełniającą nierówność $1 \leqslant n \leqslant 4080$. Stosowanie tej opcji jest zalecane tylko w przypadku, gdy istotne jest zachowanie wierności kolorów pierwszego planu.

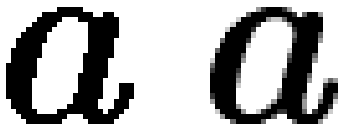
Proces rastryzacji nieuchronnie prowadzi do pewnych zniekształceń. W przypadku grafiki wektorowej, wszystkie krzywe nie będące prostymi równoległymi do jednej z krawędzi ekranu, będą w obrazie rastrowym poszarpane. Jednym ze sposobów radzenia sobie z tym problem jest wykorzystanie *antyaliasingu*, w którym wrażenie mniejszego rozmiaru piksela uzyskuje się poprzez „rozmycie" krawędzi. Ilustruje to rysunek 3.1 Opcja `--anti-alias` powoduje zastosowane tej metody do tekstu i grafiki wektorowej. Nie zaleca się korzystania z tej opcji: wyższą jakość lepiej osiągać poprzez zwiększenie rozdzielczości rastrowania, co zostało opisane w 3.3.4.2. Wówczas, w istocie, antyaliasing stosowany jest dopiero przez przeglądarkę DjVu, a nie na etapie konwersji.

[7] Słowa *kawałek* i *plaster* należy rozumieć jak w 1.2.2

[8] Takie same wartości domyślne przyjmuje *djvudigital*.

[9] Specyfikacja formatu DjVu (zobacz [2, s. 6, 33]) przewiduje również $s = 12$, ale z powodu błędu *csepdjvu* (zobacz A.6) ta wartość jest nieosiągalna.

[10] Taką samą wartość domyślną stosuje *djvudigital*. Ponadto, wedle specyfikacji formatu, jest to najczęściej stosowany współczynnik (zobacz [2, s. 6]).



Rysunek 3.1: Rastryzacja, w tej samej rozdzielczości, litery ***a*** bez antyaliasingu i z antyaliasingiem.

3.3.4.4. Dane nierastrowe

3.3.4.4.1. Metadane

Do dokumentu DjVu włączane są następujące metadane, pochodzące z *Document Information Dictionary* (zobacz [6, s. 843–844]) dokumentu PDF:

- `Title` — tytuł dokumentu;
- `Author` — nazwisko osoby, która utworzyła dokument;
- `Subject` — temat dokumentu;
- `Creator` — nazwa programu, który utworzył oryginalny dokument, z którego dokonano konwersji na format PDF (o ile dokonano takiej konwersji);
- `Producer` — nazwa programu, który dokonał konwersji na format PDF (o ile dokonano takiej konwersji);
- `CreationDate` — data i czas utworzenia dokumentu;
- `ModDate` — data i czas ostatniej modyfikacji dokumentu.

Wartości `CreationDate` i `ModDate` są konwertowane do postaci zgodnej z RFC 3339 (zobacz [23]), w której data i godzina oddzielane są spacją. Wartość `Producer` jest uzupełniania o informacje o wersji *pdf2djvu*. Pozostałe wartości są kopiowane z dokumentu PDF bez zmian.

Jeśli podano opcję `--no-metadata`, metadane nie zostaną włączone do dokumentu DjVu.

3.3.4.4.2. Konspekt

Do dokumentu DjVu jest włączany konspekt z dokumentu PDF, chyba że podano opcję `--no-outline`.

Dokumenty PDF mogą zawierać hiperłącza, mogące być częścią konspektu, których nie da się przetłumaczyć na język adnotacji DjVu (ograniczenia zostały opisane w 1.4.1.2). W takim przypadku, element konspektu w dokumencie docelowym zostanie utworzony, ale będzie nieaktywny. Dodatkowo, zostanie wyemitowane ostrzeżenie o braku możliwości pełnej konwersji.

3.3.4.4.3. Hiperłącza

Do dokumentu DjVu są włączane hiperłącza z dokumentu PDF, chyba że podano opcję `--no-hyperlinks`.

Sposób wyświetlania hiperłączy w dokumencie DjVu można ustalić za pomocą opcji `--hyperlinks=`*O*, gdzie *O* jest rozdzieloną przecinkami listą następujących parametrów:

- `border-avis` — powoduje, że ramka wokół hiperłączy będzie zawsze widoczna (inaczej byłaby widoczna tylko gdy kursor myszy znajdowałby się nad hiperłączem);

- `#rgb` — pozwala ustalić kolor ramek wokół hiperłączy; r, g, b są, zapisanymi dwoma szesnastkowymi cyframi, liczbami wyznaczającymi kolor w przestrzeni RGB[11].

Dokumenty PDF mogą zawierać hiperłącza, których nie da się przetłumaczyć na język adnotacji DjVu (ograniczenia zostały opisane w 1.4.1.2). W takim przypadku, hiperłącze w dokumencie docelowym zostanie utworzone, ale będzie prowadziło do bieżącej strony. Dodatkowo, zostanie wyemitowane ostrzeżenie o braku możliwości pełnej konwersji.

3.3.4.4.4. Warstwa tekstowa

Do dokumentu DjVu jest włączana warstwa tekstowa z dokumentu PDF, chyba że podano opcję `--no-text`.

Tekst niewidoczny

Pod uwagę brany jest zarówno tekst widoczny jak i niewidoczny. Ma to istotne znaczenie, gdyż gros popularnych narzędzi OCR produkuje jako pliki wyjściowe dokumenty PDF, w których warstwa graficzna pochodzi z oryginalnego dokumentu (skanu), a warstwa tekstowa jest niewidoczna i istnieje tylko by umożliwić wyszukiwanie w dokumencie. Przykładami są komercyjne:

- Adobe Acrobat Capture[12].
- Readiris Pro[13],
- ABBYY FineReader[14].

pdf2djvu pozwala zatem użyć tych systemów OCR do produkcji dokumentów w formacie DjVu.

Normalizacja

W dokumentach elektronicznych (zwłaszcza tych, do których składu użyto systemu TEX) typowym zjawiskiem jest występowanie ligatur: co najmniej 2 litery, w celu uzyskania określonego efektu typograficznego, są reprezentowane przez pojedynczy znak. *Naiwne* przeszukiwanie tekstu zawierającego ligatury może być nieskuteczne — a takie właśnie jest w istniejących przeglądarkach DjVu.

Standard Unicode definiuje 4 rodzaje normalizacji tekstu (zobacz [24]), tj. przekształcenia ciągu znaków do postaci równoważnej i w jakiś sposób uregulowanej:

- *NFD*: kanoniczną dekompozycję (ang. *Canonical Decomposition*),
- *NFC*: kanoniczną dekompozycję z kanoniczną kompozycją (ang. *Cannonical Decomposition, followed by Canonical Composition*),
- *NFKD*: konserwatywną dekompozycję (ang. *Compatibility Decomposition*)[15],
- *NFKC*: konserwatywną dekompozycję z kanoniczną kompozycją (ang. *Compatibilty Decomposition, followed by Canonical Composition*)[15].

[11] Na przykład `#ff0000` oznacza w pełni nasycony kolor czerwony.
[12] `http://www.adobe.com/products/acrcapture/`
[13] `http://www.neuratron.com/readiris.htm`
[14] `http://finereader.abbyy.com/`
[15] Alternatywnym tłumaczeniem angielskiego *compatibilty* jest *dostosowawczy* — zobacz [25, s. 4].

źródło	p	ó	ł	fi	n	a	ł
	U+0070	U+00F3	U+0142	U+FB01	U+006E	U+0061	U+0142
NFD	p	o	◌́	ł	fi	n	a	ł
	U+0070	U+006F	U+0301	U+0142	U+FB01	U+006E	U+0061	U+0142
NFC	p	ó	ł	fi	n	a	ł
	U+0070	U+00F3	U+0142	U+FB01	U+006E	U+0061	U+0142
NFKD	p	o	◌́	ł	f	i	n	a	ł
	U+0070	U+006F	U+0301	U+0142	U+0066	U+0069	U+006E	U+0061	U+0142
NFKC	p	ó	ł	f	i	n	a	ł
	U+0070	U+00F3	U+0142	U+0066	U+0069	U+006E	U+0061	U+0142

Tabela 3.1: Formy normalizacji na przykładzie słowa *półfinał*.

Wszystkie zostały zilustrowane w tabeli 3.1.

NFD i NFC nie nadają się do naszego celu, ponieważ nie rozbijają ligatur. NFKD rozbija ligatury, ale również diakrytyki przy wielu literach alfabetów europejskich, czego nie robi NFKC.

O ile nie podano opcji `--no-nfkc`, tekst jest poddawany normalizacji NFKC[16].

Dokładność

Dokładność z jaką zostaną zarejestrowane położenia tekstu na stronie można wybrać za pomocą następujących opcji:

- `--words`: zarejestrowane zostaną położenia każdej linii i każdego słowa;
- `--lines`: zarejestrowane zostaną tylko położenia każdej linii.

Jeśli nie podano inaczej, przyjmuje się najwyższą możliwą dokładność.

3.3.4.4.5. Strony

Strony do konwersji można wybrać za pomocą opcji `--pages=`P. P jest listą oddzielonych przecinkami specyfikacji spójnych ciągów stron postaci:

- n: strona numer n,
- m-n: wszystkie strony o numerach $m, m+1, \ldots, n$.

Numeracja stron zaczyna się od 1. Zabronione jest wybieranie wielokrotnie tej samej strony.

Jeśli nie podano inaczej, konwertowane są wszystkie strony dokumentu.

[16] Wymagany jest *poppler* $\geqslant$ 0.5.2

3.3.4.5. Gadatliwość, pomoc

O ile nie podano inaczej, *pdf2djvu* wyświetla następujące informacje podczas konwersji:

- nazwę pliku konwertowanego;
- numery stron przekonwertowanych i strony aktualnie przetwarzaniem, wraz z numerem odpowiadającego numeru strony w dokumencie docelowym;
- po zakończeniu konwersji: linię podsumowującą.

Linia podsumowująca ma postać:

`` `b` `bits/pixel; r`:1, `p`% `saved, i bytes in, o bytes out` ,

gdzie i to rozmiar pliku wejściowego, o to suma rozmiarów wszystkich plików wejściowych oraz:

$$
\begin{aligned}
r &\approx \frac{i}{o}, \\
p &\approx 100\left(1 - \frac{o}{i}\right), \\
b &\approx \frac{8o}{\sum\limits_p W_p H_p}.
\end{aligned}
$$

Opcje `-q` i `--quiet` całkowicie wyłączają wyświetlanie komunikatów informacyjnych podczas konwersji.

Opcje `-v` i `--verbose` zwiększają liczbę komunikatów informujących o szczegółach przebiegu konwersji. Można ich użyć wielokrotnie.

Opcja `--version` powoduje wyświetlenia informacji o wersji *pdf2djvu* i wersjach bibliotek, z którą został połączony podczas kompilacji.

Opcja `--help` wyświetla krótką ściągawkę z dostępnych opcji. Bardziej szczegółowy opis znajduje się na stronie podręcznika programu.

3.3.5. Przenośność

Program był w zasadzie pisany z zamiarem uruchamiania w Linuksie. Wiadomo jednak, że daje się go skompilować i używać w innych systemach uniksowych: FreeBSD, MacOS X, Cygwin. W wyniku inicjatywy niezwiązanej z autorem niniejszej pracy powstała także graficzna nakładka na *pdf2djvu* dla systemu Windows[17].

3.3.6. Możliwości rozwoju

Prosty algorytm separacji warstw wydaje się być wystarczający do typowych dokumentów PDF i daje zazwyczaj satysfakcjonujące rezultaty. Jednakże implementacja algorytmu separacji warstw, który stosuje *djvudigital* (opsianego w 2.3.4.2), powinna skutkować:

- zwiększeniem prędkości konwersji,
- zmniejszeniem zapotrzebowania na pamięć operacyjną oraz
- w przypadku dokumentów o skomplikowanej strukturze: trafnością separacji, zatem również lepszą kompresją.

[17] `http://www.trustfm.net/GeneralTools/SoftwarePdfToDjvuGUI.php?b2=1`

Niestety, wykonanie tego zadania przy pomocy biblioteki *poppler*, wymagałoby skopiowania znacznej ilości kodu i wprowadzeniu w nim licznych, być może drobnych, zmian. Byłoby to niezgodne z wcześniej przyjętymi założeniami. Szacuje się, że długość kodu programu wzrosłaby co najmniej 3-krotnie.

pdf2djvu modeluje kolory pierwszego planu zawsze ten sposób, mianowicie w formacie JB2 z kolorami. Pożądane byłoby, aby możliwy był również wybór alternatywnego sposobu, który dopuszcza DjVu, mianowicie formatu IW44 z maską JB2 (zobacz 1.2.3).

Obsługa adnotacji jest w *pdf2djvu* dość skromna, ogranicza się bowiem jedynie do hiperłączy. Tymczasem plik PDF może zawierać wiele innych rodzajów adnotacji (zobacz [6, s. 604–647]), spośród których niektóre mają swoje analogi w formacie DjVu. Niestety, pełna obsługa adnotacji w bibliotece *poppler* pojawiła się dopiero w lutym 2008 r. Ze względów czasowych, nie było możliwe zatem skorzystanie z oferowanych przez nią funkcji.

3.4. *python-djvulibre*

python-djvulibre jest zestawem dowiązań (ang. *bindings*) do publicznej części biblioteki *DjVuLibre* dla języka Python.

Biblioteka została udostępniona na zasadach zasadach Powszechnej Licencji Publicznej GNU w wersji 2 na stronie `http://freshmeat.net/projects/python-djvulibre/`.

3.4.1. Motywacja

Biblioteka *DjVuLibre* została zaimplementowana w C++, a jej publiczny interfejs jest przeznaczony dla języków C i C++. Nie istniały dotąd dowiązania dla żadnych innych języków programowania.

Python jest językiem:

- obiektowym i modularnym,
- dynamicznie typowanym,
- z automatyczną alokacją (i odśmiecaniem) pamięci,
- o czytelnej i zwartej składni,
- z obsługą błędów opartą na mechanizmie wyjątków,
- z rozległą biblioteką standardową,
- rozszerzalnym poprzez moduły pisane w C.

Cechy te pozwalają na efektywne tworzenie łatwych do utrzymania programów, w tym szybkich prototypów (ang. *rapid prototyping*).

Powstanie biblioteki *python-djvulibre* znacznie ułatwiło implementację kolejnych programów: *ocrodjvu* (zobacz 3.5) i *DjVuSmooth* (zobacz 3.6).

3.4.2. Zarys implementacji

3.4.2.1. Język programowania

Najbardziej oczywistą metodą napisania modułu roszerzeń (ang. *extension module*), tj. modułu napisanego w *innym* języku programowania jest skorzystanie z przygotowanego do

tego celu interfejsu dla języków C i C++ (Python/C API; zobacz [26]). Daje on niskopoziomowy dostęp do interpretera Pythona, pozwalając definiować moduły w całości napisane w C (lub C++).

Konsekwencją przyjęcia tej metody jest konieczność pisania *rozwlekłych i schematycznych* (ang. *boilerplate*) fragmentów kodu odpowiedzialnych za m.in. inicjację struktur danych, zliczanie referencji, obsługę sytuacji wyjątkowych. Taki kod jest trudny do utrzymania i podatny na subtelne błędy, a mógłby być, w wielu wypadkach, generowany automatycznie.

Remedium na przedstawiony problem jest Pyrex[18], język przeznaczony specjalnie do pisanie modułów rozszerzeń. Oferuje on elegancką składnię (zbliżoną do składni samego Pythona) i izoluje programistę od niskopoziomowych aspektów Python/C API; jednocześnie zapewnia ekspresywność języka C.

Użycie Pyreksa do implementacji *python-djvulibre* przyniosło realne i mierzalne oszczędności — wygenerowany przez Pyreksa kod w C jest ponad trzykrotnie większy od kodu źródłowego.

3.4.2.2. Błędy programistyczne

Biblioteka *python-djvulibre* został wyposażony w zestaw testów regresyjnych. Powinien on pomóc zapobiec w przyszłości powstawaniu niektórych typów błędów programistycznych — zarówno w dowiązaniach jak i samym *DjVuLibre*. Jest to o tyle ważne, że *DjVuLibre* nie posiada bowiem takich testów i w przeszłości zdarzały się w niej poważne regresje.

Testy regresyjne pozwoliły zauważyć nowy, opisany w A.8, błąd w *DjVuLibre*. Oprócz tego, prace implementacyjne ujawniły kilka innych usterek, opisanych w: A.9, A.10, A.13 i A.14.

Aby zmniejszyć ryzyko natknięcia się na *znane* błędy *DjVuLibre*, autor zastosował obejścia niektórych usterek: opisanej w A.15 oraz błędu w zapisywaniu dokumentów[19]. Tam, gdzie obejścia nie dało się zastosować, o możliwości napotkania błędu ostrzega dokumentacja techniczna.

3.4.3. Interfejs programisty

python-djvulibre składa się z 3 niemal niezależnych modułów. Są one przedstawione w kolejnych podrozdziałach.

Alokacja i zwalnianie pamięci i innych zasobów odbywa się automatycznie. Tam, gdzie to możliwe, obsługa błędów odbywa się za pomocą mechanizmu wyjątków.

Pełna dokumentacja techniczna biblioteki (w języku angielskim) jest dostępna za pomocą standardowego mechanizmu pomocy dla Pythona.

3.4.3.1. `djvu.sexpr`

Moduł `djvu.sexpr`, będący odpowiednikiem biblioteki *MiniExp* (zobacz 2.2.1.1), zawiera funkcje służące do konstrukcji, analizy składniowej i serializacji S-wyrażeń.

Wszelkie operacje na S-wyrażeniach są bezpieczne dla wątków (co nie jest prawdą w przypadku *MiniExp*).

[18] `http://www.cosc.canterbury.ac.nz/~greg/python/Pyrex/`
[19] Zobacz `http://bugs.debian.org/467282`.

3.4.3.2. `djvu.decode`

Moduł `djvu.decode`, będący odpowiednikiem biblioteki *DDJVU* (zobacz 2.2.1.2), zawiera wszystkie niezbędne funkcje niezbędne do implementacji przeglądarki plików DjVu.

Tam, gdzie to było możliwe, udostępnione zostały warianty funkcji działające w sposób synchroniczny. W ten sposób znacznie ułatwione zostało pisanie programów nieinteraktywnych lub działających wyłącznie na plikach lokalnych, w których asynchroniczne wywołania dają co najwyżej niewielkie korzyści, a korzystanie z nich utrudnia implementację.

3.4.3.3. `djvu.const`

Moduł `djvu.const` zawiera rozmaite stałe nie mające związku z dekodowaniem obrazów DjVu, takie jak:

- symbole, których można użyć do skonstruowania adnotacji (zobacz 1.4.1);
- symbole, których można użyć do skonstruowania innych S-wyrażeń, interpretowanych przez program *djvused* (zobacz 2.3.1.11);
- zbiory *godnych uwagi* kluczy metadanych (zobacz 1.4.1.3).

3.4.4. Możliwości rozwoju

Pożytecznym byłoby rozbudowanie *python-djvulibre* o możliwości oferowane przez narzędzia linii poleceń *DjVuLibre* (zobacz 2.3.1). Szczególne korzyści przyniosłoby to w wypadku programów o skomplikowanym interfejsie: *djvused* oraz *csepdjvu*.

3.5. *ocrodjvu*

OCRopus[20] jest sponsorowanym przez Google swobodnym systemem rozpoznawania pisma (OCR). Jego rozwojem zajmuje się obecnie zespół badawczy Image Understanding and Pattern Recognition w Niemieckim Centrum Badań nad Sztuczną Inteligencją (Deutsches Forschungszentrum für Künstliche Intelligenz).

ocrodjvu to nakładka na ten system, pozwalająca dokonywać OCR-u na dokumentu DjVu.

Program został udostępniony na zasadach zasadach Powszechnej Licencji Publicznej GNU w wersji 2 na stronie `http://freshmeat.net/projects/ocrodjvu/`.

3.5.1. Motywacja

Powstanie *pdf2djvu* w istotny sposób powiększyło możliwości przeprowadzania optycznego rozpoznawania tekstu w dokumentach DjVu (zobacz 3.3.4.4.4). Dotąd jednak jedynym sposobem wykonania tego zadania za pomocą *wyłącznie* swobodnego oprogramowania było skorzystanie z programu *gscan2pdf*[21], który — wbrew nazwie — od sierpnia 2008 roku potrafi zapisywać również dokumenty DjVu z warstwą tekstową będącą wynikiem OCR-u. Program ten ma jednak szereg wad:

- wymusza użycie interfejsu graficznego do zadań, które w istocie nie wymagają interakcji;
- nie daje żadnej kontroli nad kompresją wynikowych dokumentów DjVu;

[20] `http://ocropus.googlecode.com/`
[21] `http://gscan2pdf.sourceforge.net/`

```
<!DOCTYPE html PUBLIC "-//W3C//DTD␣XHTML␣1.0␣Transitional//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  ...
  <title>OCR Output</title>
</head>
<body>
<div class="ocr_page" title="image␣SG06p337.pbm;␣bbox␣0␣0␣4306␣6064">
...
<div class="ocr_carea">
  <p class="ocr_par">
  <span class="ocr_line" title="bbox␣2099␣4047␣3647␣4138">
    <b>Mielec</b>, msto pow., zbudowano w piaszczy-</span>
  <span class="ocr_line" title="bbox␣2017␣4134␣3651␣4223">
    stej równinie, 186 m. npm, na praw. brz. Wi-</span>
  ...
  </p>
</div>
...
</div>
</body>
</html>
```

Listing 3.1: Przykładowy dokument hOCR (porównaj z rysunkiem 1.1).

- nie umieszcza w wynikowych dokumentach żadnych informacji o położeniu rozpoznanego tekstu.

Wspomnianych wad nie ma *ocrodjvu*.

3.5.2. Zarys implementacji

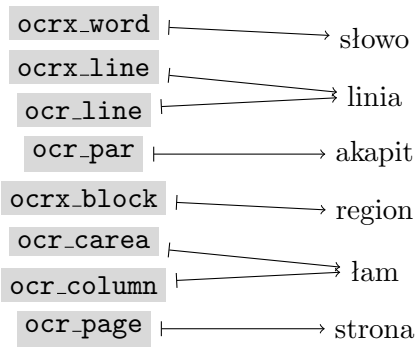
3.5.2.1. hOCR

Wynikiem działania *OCRopusa* jest plik w formacie *hOCR*. Jest to format, w którym informacje o rozmieszczeniu poszczególnych elementów na stronie zostało zaszyte, w sposób który powinien być obojętny dla zwykłych przeglądarek internetowych, w dokument HTML lub XHTML.

Przykładowy wynik OCR przedstawiony jest na listingu 3.1. Klasy `ocr_line`, `ocr_par`, `ocr_carea`, `ocr_page` i inne, niosą informacje o segmentacji tekstu. Umieszczana w atrybutach `title` własność `bbox` x_0 y_0 x_1 y_1 niesie informacje o położeniu tekstu w obrazie źródłowym: że jest o zawarty w prostokącie, którego lewy-górny róg ma współrzędne (x_0, y_0), a prawy-dolny (x_1, y_1). Przyjmuje się, że lewy-górny róg strony ma współrzędne $(0, 0)$ a współrzędne rosną w dół i w prawo; jest to konwencja odmienna od tej przyjętej w *DjVuLibre* (porównaj 2.2.1.1.4).

Pełna specyfikacja formatu hOCR znajduje się w [27].

Mapowanie regionów elementów hOCR na strefy tekstu DjVu odbywa się wg schematu przedstawionego na rysunku 3.2.



Rysunek 3.2: Odpowiedniość pomiędzy elementami hOCR a strefami tekstu DjVu.

3.5.2.2. Język programowania

Dzięki bibliotece *python-djvulibre*, program został w całości zaimplementowany w Pythonie. Użycie tak wysokopoziomowego języka pozwoliło uzyskać krótki i łatwy do utrzymania kod.

3.5.3. Interfejs użytkownika

OCRopus, w przeciwieństwie do wielu innych systemów OCR, które pozwalają na uczenie *w trakcie* analizy dokumentu, nie jest interaktywny i ma jedynie interfejs linii poleceń. Te same cechy ma *ocrodjvu*.

Niskopoziomowe polecenie `hocr2djvu` czyta ze standardowego wejścia plik w formacie hOCR i produkuje odpowiadający mu skrypt dla programu *djvused*. Niekiedy jest konieczne, aby pliki źródłowe, na których przeprowadzane było rozpoznawania tekstu, nadal istniały.

Polecenie `ocrodjvu` oczekuje jednego argumentu — nazwy pliku DjVu. Jego wywołanie powoduje uruchomienie procesu rozpoznawania tekstu, którego wyniki zostaną umieszczone, jako ukryty tekst, w dokumencie DjVu. Strony do konwersji można wybrać w ten sam sposób jak dla *pdf2djvu* (zobacz 3.3.4.4.5).

3.5.4. Możliwości rozwoju

Oczywistym ulepszeniem jest zwiększenie precyzji wydobywania tekstu tak, aby w wynikowym dokumencie DjVu zapisana była informacja o położeniu każdego znaku, a nie — jak do tej pory — jedynie każdej linii. Wymagałoby to zmian w kodzie *OCRopusa*, gdyż obecna wersja nie udostępnia w pliku wyjściowym informacji o segmentacji w obrębie linii.

3.6. *DjVuSmooth*

DjVuSmooth to prosty graficzny edytor dokumentów DjVu.

Program został udostępniony na zasadach zasadach Powszechnej Licencji Publicznej GNU w wersji 2 na stronie `http://freshmeat.net/projects/djvusmooth/`.

3.6.1. Motywacja

Zadania:

- dodania lub modyfikacji metadanych,

- dodania lub modyfikacji konspektu,
- dodania lub modyfikacji hiperłączy,
- korekty warstwy tekstowej

dają się w *DjVuLibre* co prawda wykonać przy pomocy narzędzia linii poleceń *djvused* (zobacz 2.3.1.11), jednakże jego interfejs jest dla nieprogramisty stanowczo zbyt skomplikowany. Co więcej, wymaga on by przy ustaleniu lub zmianie położeń obiektów (hiperłączy, stref tekstu) podania współrzędnych, w pikselach, punktów wyznaczających to położenie. Czyni to niektóre z tych typowych zadań wyjątkowo uciążliwymi.

Rozwiązaniem tego problemu byłby graficzny edytor dokumentów DjVu. *DjVuSmooth* jest, spośród dostępnych na swobodnej licencji, pierwszym takim programem.

3.6.2. Zarys implementacji

3.6.2.1. Język programowania

Podobnie jak w przypadku *ocrodjvu*, dzięki bibliotece *python-djvulibre*, program został w całości zaimplementowany w Pythonie. Użycie tak wysokopoziomowego języka pozwoliło uzyskać stosunkowo krótki i łatwy do utrzymania kod.

3.6.2.2. Interfejs graficzny

Do zaprogramowania interfejsu graficznego została użyta biblioteka *wxWidgets*[22] w wariancie przeznaczonym dla Pythona, tj. *wxPython*[23]. Wymagana jest wersja 2.6 biblioteki.

3.6.2.3. Edycja warstwy tekstowej

DjVuSmooth wykorzystuje fakt, że — przy pewnych założeniach — da się użyć zwykłego edytora tekstu do przeprowadzenia korekty warstwy tekstowej. Takie rozwiązanie umożliwia korzystanie z udogodnień, jakie ten edytor może zapewniać, np. sprawdzanie na bieżąco pisowni, funkcja *wyszukaj i zamień* itp.

Wspomniane założenia to:

- W strukturze tekstu znajdują się linie.
- Ewentualna istniejąca segmentacja na pojedyncze znaki może zostać utracona.
- Nie będą przeprowadzane operacje (dodawanie, usuwanie) na całych liniach.
- W obrębie każdej linii, wykonane operacje na tekście są minimalne w sensie odległości redakcyjnej.

Wówczas, na podstawie tekstu początkowego i tekstu po edycji da się odtworzyć ciąg wykonanych operacji na zwykłym tekście, które następnie można powtórzyć na strefach tekstu.

Do wyznaczenia minimalnej odległości redakcyjnej oraz ciągu operacji, który tej odległości odpowiada, użyto klasycznego algorytmu dynamicznego.

[22] `http://wxwidgets.org/`
[23] `http://www.wxpython.org/`

3.6.3. Interfejs użytkownika

3.6.3.1. Konfiguracja

Włączyć lub wyłączyć pasek boczny można za pomocą menu Settings → Show sidebar (skrót F9).

Zewnętrzny edytor należy skonfigurować wybierając menu Settings → External editor, a następnie wprowadzając w oknie dialogowym linię poleceń (tj. nazwę z ewentualnymi argumentami) wybranego edytora tekstu. Program powinien działać w pierwszym planie, by dało się programowo sprawdzić, kiedy skończył działanie.

Konfiguracja jest zapisywana w katalogu domowym użytkownika, w pliku `.DjVuSmooth`. Oprócz wyżej wymienionych ustawień, zapamiętywane są też ostatnie położenie i rozmiar okna programu.

3.6.3.2. Otwieranie, zamykanie i zapisywanie dokumentów

Dokument DjVu można otworzyć podając nazwę pliku jako argument przy wywołaniu *DjVuSmooth* lub korzystając z menu File → Open (skrót Ctrl + O). Jednocześnie może być otwarty co najwyżej jeden dokument; by równolegle móc edytować wiele dokumentów równolegle, trzeba uruchomić kolejne kopie programu.

Zamknąć otwarty dokument można:

- korzystając z menu File → Close (skrót Ctrl + W);
- zamykając cały program, np. poprzez menu File → Quit (skrót Ctrl + Q);
- pośrednio, otwierając inny dokument.

Zapisać zmieniony dokument można:

- korzystając z menu File → Save (skrót Ctrl + S);
- odpowiadając twierdząco na pytanie o zapis przy zamykaniu dokumentu.

3.6.3.3. Funkcje przeglądarki

DjVuSmooth może służyć jako (dość prymitywna) przeglądarka dokumentów DjVu.

3.6.3.3.1. Strony

W dokumencie N-stronicowym, przy bieżącej stronie nr $n \in \{1, 2, \ldots, N\}$ (jej numer wyświetlany jest w pasku statusu), skoczyć do strony nr k można za pomocą następujących pozycji menu:

- dla $k = 1$: Go → First page (skrót Ctrl + Home);
- dla $k = \max(1, n-1)$: Go → Previous page (skrót Page Up);
- dla $k = \min(N, n+1)$: Go → Next page (skrót Page Down);
- dla $k = N$: Go → Last page (skrót Ctrl + End);
- dla dowolnego k: Go → Go to page....

3.6.3.3.2. Powiększenie

Aby obraz zawsze zajmował całą szerokość okna, przy zachowaniu proporcji szerokość/wysokość, należy skorzystać z menu View → Zoom → Fit width.

Aby obraz zawsze zajmował możliwie całe okno, przy zachowaniu proporcji szerokość/wysokość, należy skorzystać z menu View → Zoom → Fit page.

Aby obraz zawsze zajmował całe okno, bez zachowania proporcji szerokość/wysokość, należy skorzystać z menu View → Zoom → Stretch.

Aby jeden piksel obrazu odpowiadał jednemu pikselowi ekranu, należy skorzystać z menu View → Zoom → One to one.

Aby uzyskać obraz wielkości $p\%$ naturalnego rozmiaru strony, przy założeniu że ekran ma rozdzielczość 100 dpi, należy skorzystać z menu View → Zoom → p%. Następnie pomniejszenie/powiększenie można uzyskać za pomocą menu View → Zoom → Zoom in (skrót +) i View → Zoom → Zoom out (skrót −).

3.6.3.3.3. Obrazy rastrowe

Sposób wyświetlania obrazów rastrowych można ustawić w następujący sposób:

- Aby wyświetlać pierwszy plan i tło, należy skorzystać z menu View → Image → Color (skrót Alt + C).
- Aby wyświetlać jedynie maskę pierwszego planu, należy skorzystać z menu View → Image → Stencil.
- Aby wyświetlać tylko pierwszy plan lub tylko tło, należy skorzysać z menu (odpowiednio) View → Image → Foreground lub View → Image → Background.
- Aby zrezygnować z wyświetlania obrazów rastrowych, należy skorzystać z menu View → Image → None (skrót Alt + N).

3.6.3.3.4. Dane nierastrowe

Na obraz mogą być naniesione niektóre dane nierastrowe:

- hiperłącza — za pomocą menu View → Non-raster data → Hyperlinks (skrót Alt + H);
- wartstwa tekstowa — za pomocą menu View → Non-raster data → Text (skrót Alt + T).

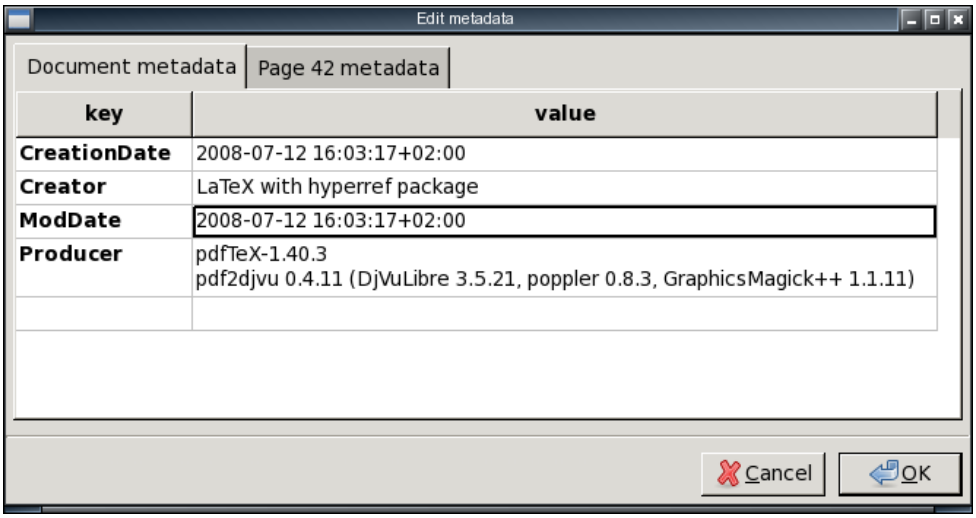
Jednocześnie może być wyświetlany co najwyżej jeden rodzaj dodatkowych danych. Domyślnie nie są wyświetlane żadne; do tego stanu można powrócić za pomocą menu View → Non-raster data → None.

3.6.3.4. Edycja

3.6.3.4.1. Metadane

Menu Edit → Metadata (skrót Ctrl + M) otwiera okno dialogowe umożliwiające edycję metadanych. Aby ją przeprowadzić należy:

1. Wybrać zakładkę, by zdecydować, które metadane mają być zmienione:
   - całego dokumentu — zakładkę Document metadata;
   - tylko bieżącej (n-tej) strony — zakładkę Page n metadata.
2. Zmienić zawartość pól tekstowych ułożonych w siatkę, gdzie każdy wiersz reprezentuje jedną parę klucz–wartość. Ostatni wiersz siatki jest przeznaczony do wprowadzania



Rysunek 3.3: Zrzut ekranu — okno dialogowe edycji metadanych.

nowych par klucz. Klucze *godne uwagi* (zdefiniowane w 1.4.1.3) wyróżniane są pismem pogrubionym. Znak nowego wiersza podczas edycji pola tekstowego można uzyskać naciskając Shift + Enter.

3.6.3.4.2. Konspekt

Konspekt w postaci rozwijalnego drzewka można obejrzeć w panelu bocznym, wybierając zakładkę Outline. W drzewku można:

- zmieniać tytuły poszczególnych pozycji;
- przesuwać poszczególne pozycje w drzewie za pomocą mechanizmu przeciągnij i upuść; upuszczenie pozycji A na innej pozycji B (o ile A nie jest przodkiem B) spowoduje, że A stanie się ostatnim dzieckiem B.

Aby dodać do konspektu bieżącą stronę, należy skorzystać z menu Edit → Outline → Bookmark this page (skrót Ctrl + B).

Największą swobodę daje edycja konspektu przy pomocy zewnętrznego edytora tekstu. Aby skorzystać z tej możliwości, należy skorzystać z menu Edit → Outline → External editor. Edytowany plik składa się z linii opisującej poszczególne pozycje konspektu. i-ta linia ma postać $t_i u_i\ d_i$, gdzie:

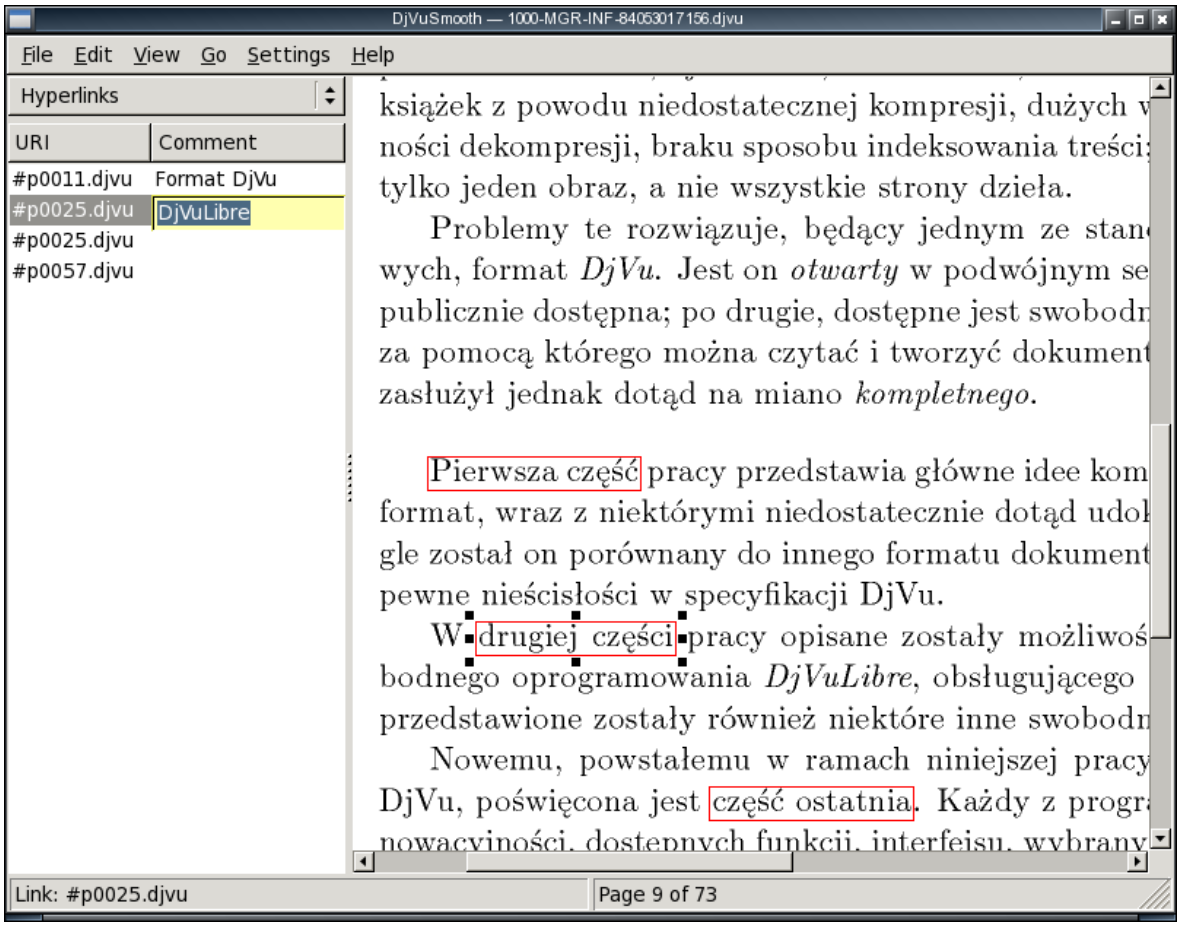
- t_i jest wcięciem (ciągiem spacji);
- u_i jest adresem;
- d_i jest opisem.

Znaczenie wcięć jest następujące: j-ta pozycja jest rodzicem k-tej wtedy i tylko wtedy gdy $|t_j| < |t_k| \leqslant |t_i|$ dla każdego $j < i \leqslant k$.

Wybranie menu Outline → Remove spowoduje usunięcie całego konspektu.

3.6.3.4.3. Hiperłącza

Listę wszystkich hiperłączy znajdujących się na bieżącej stronie można obejrzeć w panelu bocznym, wybierając zakładkę Hyperlinks. Na liście można:



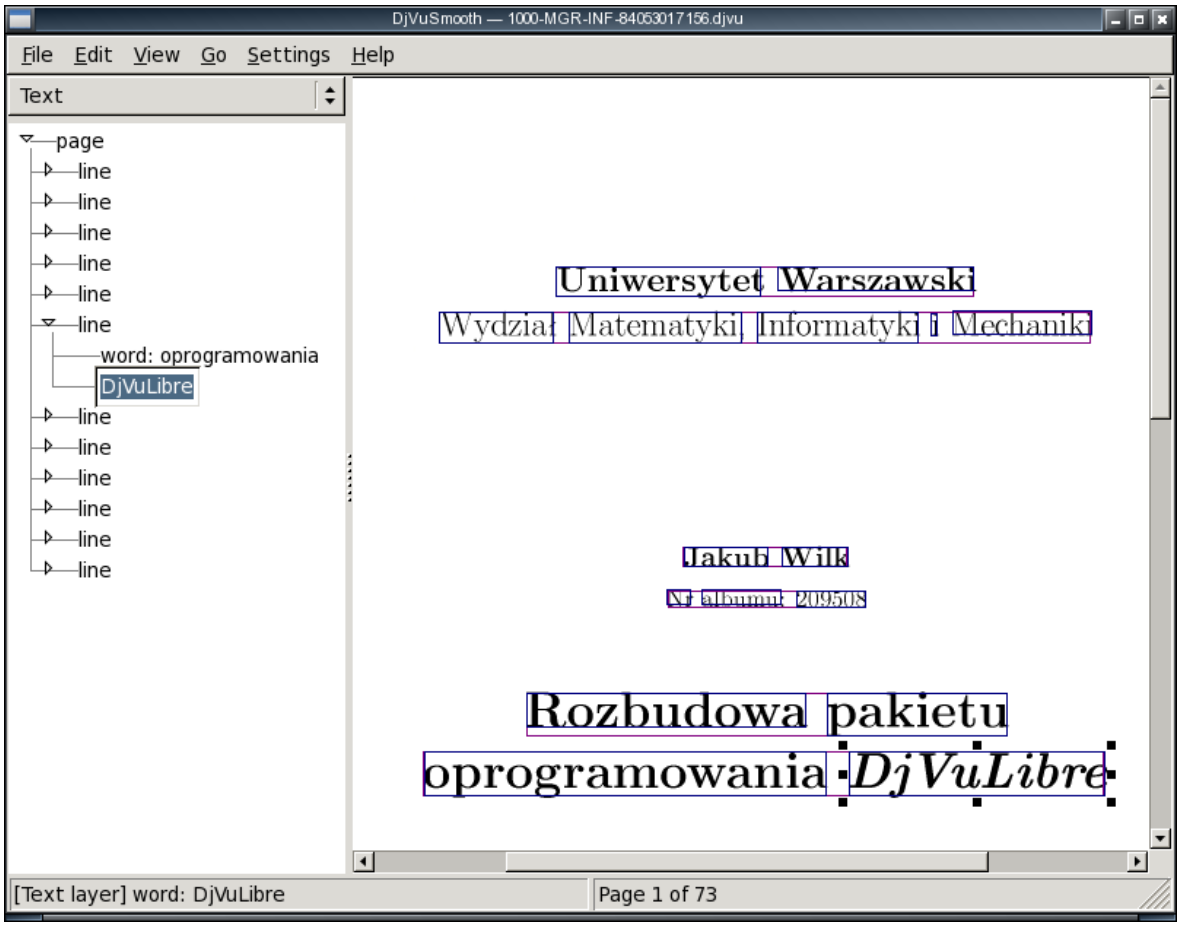
Rysunek 3.4: Zrzut ekranu — edycja hiperłączy.

- zmienić adres hiperłącza;
- zmienić komentarz hiperłącza;
- zmienić pozostałe właściwości hiperłącza w oknie dialogowym, wybierając Properties z menu kontekstowego;
- usunąć hiperłącze, wybierając Delete z menu kontekstowego (skrót Delete);
- utworzyć nowe hiperłącze, wybierając New hyperlink z menu kontekstowego; hiperłącze pojawi się w lewym-dolnym rogu strony.

Jeżeli włączone jest wyświetlanie hiperłączy na obrazie rastrowym (domyślnie po wybraniu Hyperlinks w panelu bocznym; zobacz też 3.6.3.3.4):

- emblemat hiperłącza można zaznaczyć myszką, a następnie zmieniać jego położenie i rozmiar;
- można korzystać z tego samego menu kontekstowego jak w przypadku panelu bocznego.

Panel właściwości hiperłącza umożliwia ustawienie jego adresu, ramki docelowej, kształtu i innych parametrów. Obsługa kształtów *linia* i *wielokąt* jest zaimplementowana tylko częściowo: nie można stworzyć hiperłączy tego kształtu ani zmienić kształtu istniejącego hiperłącza na jeden z tych; ograniczona jest też możliwość określania ich położeń.



Rysunek 3.5: Zrzut ekranu — edycja warstwy tekstowej.

3.6.3.4.4. Warstwa tekstowa

Strukturę stref tekstu na bieżącej stronie można obejrzeć w panelu bocznym, wybierając zakładkę Text. W drzewku można:

- zmienić tekst w liściu;
- usunąć strefę tekstu klawiszem Delete.

Jeżeli włączone jest wyświetlanie stref tekstu na obrazie rastrowym (domyślnie po wybraniu Text w panelu bocznym; zobacz też 3.6.3.3.4):

- strefy różnych typów oznaczone są innymi kolorami;
- emblemat hiperłącza można zaznaczyć myszką, a następnie:
  - zmieniać jego położenie i rozmiar lub
  - usunąć klawiszem Delete.

Jeżeli w strukturze tekstu znajdują się linie, wygodnym sposobem poprawiania drobnych zniekształceń tekstu i błędów w segmentacji jest edycja tekstu strony w zewnętrznym edytorze (zobacz 3.6.2.3). Funkcję tę można wywołać z menu Edit → Text → External editor (skrót Control + T).

Spłaszczyć strukturę stref tekstu (na bieżącej stronie lub w całym dokumencie) można wybierając z menu Edit → Text → Flatten i korzystając z okna dialogowego.

3.6.4. Możliwości rozwoju

DjVuSmooth jest wciąż programem mocno niedoskonałym; brakuje mu wielu funkcji, które *powinien* posiadać pełnowartościowy edytor dokumentów DjVu. Niektóre z nich to:

- konwersja z konwencjonalnych formatów plików graficznych do DjVu;
- manipulowanie stronami dokumentu wielostronicowego (dodawanie, usuwanie, przesuwanie stron, łączenie wielu dokumentów jeden);
- wyszukiwanie w warstwie tekstowej;
- pełna edycja warstwy tekstowej, w tym tworzenie jej *od podstaw*;
- obsługa *wszystkich* kształtów hiperłączy;
- historia zmian i możliwość ich wycofywania;
- zapisywanie pliku pod inną nazwą i w innym formacie (spakowanym lub rozdzielonym);
- eksport do innych formatów graficznych.

3.7. Podsumowanie

Efektem niniejszej pracy jest:

- znaczne powiększenie repertuaru swobodnego oprogramowania integrującego proces OCR z formatem DjVu;
- pojawienie się graficznego edytora dokumentów DjVu, który powinien znacznie ułatwić ich modyfikację tzw. *zwykłym użytkownikom*.

Nowo powstałe oprogramowanie, pomimo niedojrzałości, cieszą się już pewną popularnością: program *pdf2djvu* stał się już częścią dwóch dystrybucji Linuksa — Debian i Ubuntu. Będzie to autora tej pracy motywacją do dalszej pracy nad rozwojem tych, a być może kolejnych, programów obsługujących format DjVu.

Dodatek A

Błędy w *DjVuLibre*

Niniejszy dodatek zestawia usterki odnalezione w *DjVuLibre* przez autora niniejszej pracy podczas prac implementacyjnych. Wszystkie błędy zostały zgłoszone; większość została w krótkim czasie naprawiona przez twórców *DjVuLibre*.

Tekst ten ma w dużej mierze charakter techniczny i może zawierać nieobjaśnione niezrozumiałe terminy.

A.1.

djvumake nie pozwala włączyć ograniczonej liczby kawałków pliku IW44. Błąd zgłoszony 25 lutego 2007 r. wraz z łatą; naprawiony 26 marca 2007 r.

`http://bugs.debian.org/412316`

A.2.

Mylące literówki w dokumentacji biblioteki dzielonej. Błąd zgłoszony 8 października 2007 r; naprawiony 29 listopada 2007 r.

`http://sf.net/tracker/?func=detail&aid=1809441&group_id=32953&atid=406585`

A.3.

c44 nie pozwala na zakodowanie plików o rozmiarze < 16 bajtów. Błąd zgłoszony 9 grudnia 2007 r. wraz z łatą; naprawiony 11 stycznia 2008 r.

`http://bugs.debian.org/455331`

A.4.

djvuextract narusza ochronę pamięci w przypadku próby użycia na uszkodzonym pliku. Błąd zgłoszony 12 grudnia 2007 r.; naprawiony 11 stycznia 2008 r.

`http://bugs.debian.org/455992`

A.5.

djvused nie pozwala ustawić konspektu w dokumentach rozdzielonych. Błąd zgłoszony 28 grudnia 2007 r.; naprawiony 11 stycznia 2008 r.

http://bugs.debian.org/458086

A.6.

csepdjvu nie pozwala na tło o rozdzielczości 12-krotnie mniejszej niż pierwszego planu. Błąd zgłoszony 29 grudnia 2007 r. wraz z łatą; naprawiony 11 stycznia 2008 r.

http://bugs.debian.org/458211

A.7.

Myląca literówka w dokumentacji *djvused*. Błąd zgłoszony 29 grudnia 2007 r.; naprawiony 11 stycznia 2008 r.

http://bugs.debian.org/458241

A.8.

Usterka w obsłudze sytuacji wyjątkowych doprowadza do zawieszenia się programu. Błąd zgłoszony 2 lutego 2008 r.; naprawiony następnego dnia.

http://sf.net/tracker/?func=detail&aid=1885172&group_id=32953&atid=406583

A.9.

djvups z opcją `-text=yes` narusza ochronę pamięci w przypadku próby użycia na pliku z tekstem ze znakami spoza ASCII. Błąd zgłoszony 3 marca 2008 r.

http://sf.net/tracker/?func=detail&aid=1906108&group_id=32953&atid=406583

A.10.

Usterka w implementacji *sprytnych wskaźników* doprowadza do naruszenia ochrony pamięci. Błąd zgłoszony 4 marca 2008 r.; naprawiony następnego dnia.

http://sf.net/tracker/?func=detail&aid=1907101&group_id=32953&atid=406583

A.11.

DjVuLibre nie kompiluje się pod Cygwinem. Błąd zgłoszony 10 marca 2008 r.; naprawiony tego samego dnia.

http://sf.net/tracker/?func=detail&aid=1911036&group_id=32953&atid=406583

A.12.

Zmiana ABI biblioteki nie pociągnęła za sobą zmiany *soname* biblioteki dzielonej. Błąd zgłoszony 12 marca 2008 r.; naprawiony 14 marca 2008 r.

http://sf.net/tracker/?func=detail&aid=1912753&group_id=32953&atid=406583

A.13.

Funkcja `ddjvu_document_get_anno()` zwraca `miniexp_nil` w przypadku błędu. Błąd zgłoszony 12 marca 2008 r.; naprawiony 14 marca 2008 r.

`http://sf.net/tracker/?func=detail&aid=1912781&group_id=32953&atid=406583`

A.14.

Dokumentacja funkcji `ddjvu_document_get_pagetext()` jest niekompletna. Błąd zgłoszony 14 marca 2008 r; naprawiony 17 marca 2008 r. Kolejne usterki zostały zgłoszone 17 kwietnia 2008 r; naprawione tego samego dnia.

`http://sf.net/tracker/?func=detail&aid=1914312&group_id=32953&atid=406583`

A.15.

Usterka w implementacji odśmiecacza S-wyrażeń doprowadza do naruszenia ochrony pamięci. Błąd zgłoszony 16 marca 2008 r.; naprawiony tego samego dnia.

`http://sf.net/tracker/?func=detail&aid=1915053&group_id=32953&atid=406583`

A.16.

djvused opuszcza jeden znak podczas ustawiania adnotacji. Błąd zgłoszony 16 marca 2008 r.; naprawiony tego samego dnia.

`http://sf.net/tracker/?func=detail&aid=1915337&group_id=32953&atid=406583`

A.17.

djvused ustawie metadane w dokumencie rozdzielonym w czasie $\Theta(n^2)$, gdzie n jest liczbą stron. Objawy jako pierwszy zaobserwował Janusz S. Bień. Błąd zgłoszony 6 marca 2008 r; do tej pory[1] nie naprawiony.

`http://sf.net/tracker/?func=detail&aid=1935916&group_id=32953&atid=406583`

A.18.

csepdjvu, w określonych warunkach, produkuje pliki niezgodne ze specyfikacją. Błąd zgłoszony 2 maja 2008 r., naprawiony 5 maja 2008 r.

`http://sf.net/tracker/?func=detail&aid=1956075&group_id=32953&atid=406583`

A.19.

Różne narzędzia mają różne zakresy dopuszczalnych rozdzielczości. Błąd zgłoszony 2 maja 2008 r.; naprawiony 5 maja 2008 r.

`http://sf.net/tracker/?func=detail&aid=1956093&group_id=32953&atid=406583`

[1] Stan na 13 lipca 2008 r.

A.20.

Przeglądarka *djview4* niepoprawnie obraca ukryty tekst. Błąd zgłoszony 4 maja 2008 r.; naprawiony następnego dnia.

`http://sf.net/tracker/?func=detail&aid=1957416&group_id=32953&atid=406583`

A.21.

csepdjvu skleja ze sobą obrócone słowa. Błąd zgłoszony 22 maja 2008 r.; do tej pory[2] nie naprawiony.

`http://sf.net/tracker/?func=detail&aid=1969580&group_id=32953&atid=406583`

A.22.

Rozbieżność pomiędzy wymaganiami dotyczącymi rozdzielczości warstw strony dokumentu DjVu określonymi w specyfikacji a faktycznie zaimplementowanymi w *DjVuLibre*. Błąd zgłoszony 25 maja 2008 r; usterka w specyfikacji została udokumentowana następnego dnia.

`http://sf.net/tracker/?func=detail&aid=1972089&group_id=32953&atid=406583`

A.23.

Literówka w dokumentacji programu *djvused*. Błąd zgłoszony 17 czerwca 2008 r; naprawiony tego samego dnia.

`http://sf.net/tracker/?func=detail&aid=1995605&group_id=32953&atid=406583`

A.24.

djvused nie pozwala ustawić tekstu o płaskiej strukturze. Błąd zgłoszony 17 czerwca 2008 r; naprawiony tego samego dnia.

`http://sf.net/tracker/?func=detail&aid=1995613&group_id=32953&atid=406583`

A.25.

djvudigital nie wyodrębnia konspektu niektórych dokumentów. Błąd zgłoszony 18 czerwca 2008 r; do tej pory[2] nie naprawiony.

`http://sf.net/tracker/?func=detail&aid=1997033&group_id=32953&atid=406583`

A.26.

Nieudokumentowana funkcja programu *djvused*. Błąd zgłoszony 18 czerwca 2008 r; naprawiony tego samego dnia.

`http://sf.net/tracker/?func=detail&aid=1997190&group_id=32953&atid=406583`

[2]Stan na 13 lipca 2008 r.

Dodatek B

Zawartość płyty CD dołączonej do pracy

Załączona płyta CD zawiera:

`text/`	wersję elektroniczną niniejszej pracy:
`  thesis.tar.gz`	— w postaci źródłowej,
`  thesis.pdf`	— w formacie PDF,
`  thesis.djvu`	— w formacie DjVu;
`pdf2djvu/`	program *pdf2djvu* w wersji 0.4.11:
`  pdf2djvu_0.4.11.tar.gz`	— kod źródłowy,
`  pdf2djvu_0.4.11_amd64.deb`	— binarny pakiet debianowy[1] (i386),
`  pdf2djvu_0.4.11_i386.deb`	— binarny pakiet debianowy[1] (AMD64);
`python-djvulibre/`	bibliotekę *python-djvulibre* w wersji 0.1.8:
`  python-djvulibre_0.1.8.tar.gz`	— kod źródłowy,
`  python-djvulibre_0.1.8-doc.tar.gz`	— dokumentację techniczną,
`  python-djvulibre_0.1.8_i386.deb`	— binarny pakiet debianowy[1] (i386),
`  python-djvulibre_0.1.8_amd64.deb`	— binarny pakiet debianowy[1] (AMD64);
`ocropus/`	program *OCRopus* w wersji 0.2:
`  ocropus_0.2.orig.tar.gz`	— oryginalny kod źródłowy,
`  ocropus_0.2-1.diff.gz`	— łata źródłowego pakietu debianowego,
`  ocropus_0.2-1.dsc`	— opis źródłowego pakietu debianowego,
`  ocropus_0.2-1_i386.deb`	— binarny pakiet debianowy[1] (i386),
`  ocropus_0.2-1_amd64.deb`	— binarny pakiet debianowy[1] (AMD64);
`ocrodjvu/`	program *ocrodjvu* w wersji 0.1.2:
`  ocrodjvu_0.1.2.tar.gz`	— kod źródłowy,
`  ocrodjvu_0.1.2_all.deb`	— binarny pakiet debianowy[1].
`djvusmooth/`	program *DjVuSmooth* w wersji 0.1.3:
`  djvusmooth_0.1.3.tar.gz`	— kod źródłowy.
`  djvusmooth_0.1.3_all.deb`	— binarny pakiet debianowy[1];
`README.build`	instrukcję kompilacji i instalacji programów.

[1]Pakiety debianowe przeznaczone są dla dystrybucji Debian 5.0 (Lenny) i Ubuntu 8.10 (Intrepid Ibex).

Dodatek C

GNU Free Documentation License

Preamble

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

C.1. Applicability and Definitions

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "*Document*", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "*you*". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "*Modified Version*" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "*Secondary Section*" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "*Invariant Sections*" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "*Cover Texts*" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "*Transparent*" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "*Opaque*".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "*Title Page*" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section "*Entitled XYZ*" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "*Acknowledgements*", "*Dedications*", "*Endorsements*", or "*History*".) To "*Preserve the Title*" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included

by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

C.2. Verbatim copying

You may copy and distribute the Document in any medium, either commercially or non-commercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section C.3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

C.3. Copying in quantity

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

C.4. Modifications

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution

and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.

N. Do not retitle any existing section to be Entitled “Endorsements” or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version’s license notice. These titles must be distinct from any other section titles.

You may add a section Entitled “Endorsements”, provided it contains nothing but endorsements of your Modified Version by various parties–for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

C.5. Combining Documents

You may combine the Document with other documents released under this License, under the terms defined in section C.4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled “History” in the various original documents, forming one section Entitled “History”; likewise combine any sections Entitled “Acknowledgements”, and any sections Entitled “Dedications”. You must delete all sections Entitled “Endorsements”.

C.6. Collections of documents

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

C.7. Aggregation with Independent Works

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section C.3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

C.8. Translation

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section C.4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section C.4) to Preserve its Title (section C.1) will typically require changing the actual title.

C.9. Termination

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

C.10. Future revisions of this license

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See `http://www.gnu.org/copyleft/`.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

Addendum: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

> Copyright © *year your name*.
>
> Permission is granted to copy, distribute and/or modify this document
> under the terms of the GNU Free Documentation License, Version 1.2
> or any later version published by the Free Software Foundation;
> with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.
> A copy of the license is included in the section entitled ‘‘GNU
> Free Documentation License’’.

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with ... Texts." line with this:

> with the Invariant Sections being *list their titles*, with the
> Front-Cover Texts being *list*, and with the Back-Cover Texts being *list*.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

Bibliografia

[1] AT&T. *Specification of DjVu image compression format*, Kwiecień 1999.
http://djvu.sf.net/specs/djvu2spec.djvu

[2] Lizardtech, a Celartem Company. *Lizardtech DjVu Reference. DjVu v3*, Wrzesień 2005.
http://www.lizardtech.com/files/doc/techinfo/DjVu3Spec.djvu

[3] *Actual and proposed changes to the DjVu format*, Maj 2008.
http://djvu.cvs.sf.net/djvu/djvulibre-3.5/doc/djvuchanges.txt?revision=1.13

[4] Adam Langley, Dan S. Bloomberg. Google books: Making the public domain universally accessible. Xiaofan Lin, Berrin A. Yanikoglu, redaktorzy, *Document Recognition and Retrieval XIV*, wolumen 6500 serii *SPIE Proceedings*, strony 65000H1–65000H10, San Jose, CA, USA, 2007. SPIE.
http://link.aip.org/link/?PSI/6500/65000H/1

[5] *Słownik Geograficzny Królestwa Polskiego i innych krajów słowiańskich.*
http://www.mimuw.edu.pl/polszczyzna/SGKPi/

[6] Adobe Systems Incorporated. *PDF Reference, sixth edition: Adobe Portable Document Format version 1.7*, Listopad 2006.
http://www.adobe.com/devnet/acrobat/pdfs/pdf_reference_1-7.pdf

[7] Yann LeCun, Léon Bottou, Patrick Haffner, Jeffery Triggs, Bill Riemers, Luc Vincent. Overview of the DjVu document compression technology. *Proceedings of the Symposium on Document Image Understanding Technologies (SDIUT'01)*, strony 119–122, Columbia, MD, Kwiecień 2001.
http://leon.bottou.org/papers/lecun-2001b

[8] Ricardo de Queiroz, Robert Buckley, Ming Xu. Mixed raster content (MRC) model for compound image compression. *Proceedings of the IS&T/SPIE Symposium on Electronic Imaging, Visual Communications and Image Processing*, wolumen 3653, strony 1106–1117, San Jose, CA, Luty 1999. SPIE.
http://signal.ece.utexas.edu/~queiroz/papers/ei99mrc.pdf

[9] Léon Bottou, Steven Pigeon. Lossy compression of partially masked still images. *Proceedings of IEEE Data Compression Conference*, Snowbird, UT, Kwiecień 1998.
http://leon.bottou.org/papers/bottou-pigeon-98

[10] Roy T. Fielding, Jim Gettys, Jeff Mogul, Henrik Frystyk Nielsen, Larry Masinter, Paul J. Leach, Tim Berners-Lee. *Hypertext Transfer Protocol — HTTP/1.1*, Czerwiec 1999.
http://www.ietf.org/rfc/rfc2616

[11] Tim Bernes-Lee, Larry Masinter, Roy T. Fielding. *Uniform Resource Identifier (URI): Generic Syntax*, Styczeń 2005.
http://www.ietf.org/rfc/rfc3986

[12] *HTML 4.01 Specification*, Grudzień 1999.
http://www.w3.org/TR/html401/

[13] Oren Patashnik. BibTeX*ing*, Luty 1988. Dokumentacja dla użytkowników BibTeXa.
http://www.ctan.org/get/biblio/bibtex/contrib/doc/btxdoc.pdf

[14] Brian W. Kernighan, Dennis M. Ritchie. *Język ANSI C*. Klasyka Informatyki. Wydawnictwa Naukowo-Techniczne, wydanie piąte, 2000. Z angielskiego przełożyli Dantua i Marek Kruszewscy.

[15] David Robinson, Ken A. L. Coar. *The Common Gateway Interface (CGI) Version 1.1*, Październik 2004.
http://www.ietf.org/rfc/rfc3875

[16] Yann LeCun, Léon Bottou, Patrick Haffner, Jeffery Triggs, Bill Riemers, Luc Vincent. Efficient conversion of digital documents to multilayer raster formats. *Proceedings of the Sixth International Conference on Document Analysis and Recognition*, strony 444–448, Seattle, Wrzesień 2001. IEEE.
http://leon.bottou.org/papers/bottou-2001

[17] Moritz Muehlenhoff. Bits from the security team, Październik 2007.
http://lists.debian.org/debian-devel-announce/2007/10/msg00010.html

[18] Vulnerability summary CVE-2007-4352, Lipiec 2007.
http://nvd.nist.gov/nvd.cfm?cvename=CVE-2007-4352

[19] Vulnerability summary CVE-2007-5392, Lipiec 2007.
http://nvd.nist.gov/nvd.cfm?cvename=CVE-2007-5392

[20] Vulnerability summary CVE-2007-5393, Lipiec 2007.
http://nvd.nist.gov/nvd.cfm?cvename=CVE-2007-5393

[21] Vulnerability summary CVE-2008-1693, Kwiecień 2008.
http://nvd.nist.gov/nvd.cfm?cvename=CVE-2008-1693

[22] Vulnerability summary CVE-2008-2950, Lipiec 2008.
http://nvd.nist.gov/nvd.cfm?cvename=CVE-2008-2950

[23] Graham Klyne, Chris Newman. *Date and Time on the Internet: Timestamps*, Lipiec 2002.
http://www.ietf.org/rfc/rfc3339

[24] Mark Davis, Martin Dürst. *Unicode Normalization Forms*, Kwiecień 2008.
http://www.unicode.org/reports/tr15/tr15-29.html

[25] Janusz S. Bień. Standard Unicode 4.0. Wybrane pojęcia i terminy. *Biuletyn GUST*, 20:9–14, 2004.
http://www.mimuw.edu.pl/~jsbien/publ/BGUST04/JSB-Bach04n.pdf

[26] Guido van Rossum. *Extending and Embedding the Python Interpreter*. Python Software Foundation, wydanie 2.5.2, Luty 2008.
`http://docs.python.org/ext/ext.html`

[27] Thomas Breuel. *The hOCR Embedded OCR Workflow and Output Format*, Grudzień 2007.
`http://docs.google.com/View?docid=dfxcv4vc_67g844kf`