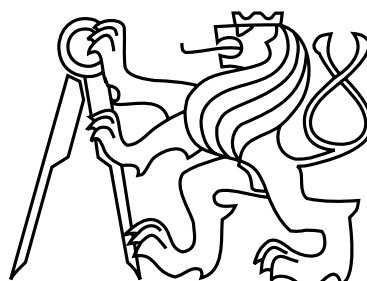


České vysoké učení technické v Praze  
Fakulta elektrotechnická  
Katedra počítačů



Bakalářská práce

## **Planární segmentace mračna bodů**

*Daniel Princ*

Vedoucí práce: Ing. David Sedláček

Studijní program: Softwarové technologie a management, Bakalářský

Obor: Web a multimedia

22. dubna 2012



## Poděkování

Zde můžete napsat své poděkování, pokud chcete a máte komu děkovat.



## Prohlášení

Prohlašuji, že jsem práci vypracoval samostatně a použil jsem pouze podklady uvedené v příloženém seznamu.

Nemám závažný důvod proti užití tohoto školního díla ve smyslu §60 Zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon).

V Praze dne 25. 5. 2015

.....



# Abstract

Translation of Czech abstract into English.

# Abstrakt

Tato bakalářská práce se zabývá problematikou zpracování mračna bodů, konkrétně segmentací na planární primitiva. Toto má uplatnění zejména při digitální 3D rekonstrukci objektů reálného světa. V současné době je snaha co nejvíce automatizovat proces zpracování naměřených dat, planární segmentace je jednou ze základních částí tohoto procesu.

Konkrétním cílem této bakalářské práce je implementovat dva algoritmy, které budou segmentaci provádět co nejvíce automaticky a zároveň v dostatečné kvalitě. Algoritmy jsou implementovány jako součást 3D rekonstrukčního nástroje ArchiRec3D.





# Obsah

<b>1</b>	<b>Úvod</b>	<b>1</b>
1.1	Segmentace mračna bodů . . . . .	2
1.2	Struktura práce . . . . .	3
<b>2</b>	<b>Analýza a návrh řešení</b>	<b>5</b>
2.1	Segmentace na základě velikosti normálového vektoru . . . . .	5
2.1.1	Adaptivní válcová definice okolí . . . . .	5
2.1.2	Výpočet parametrů pomocí okolí bodu . . . . .	8
2.2	Segmentace pomocí vyhledávání dominantních os . . . . .	8
<b>3</b>	<b>Implementace</b>	<b>9</b>
<b>4</b>	<b>Testování</b>	<b>11</b>
<b>5</b>	<b>Závěr</b>	<b>13</b>
<b>A</b>	<b>Pokyny a návody k formátování textu práce</b>	<b>17</b>
A.1	Vkládání obrázků . . . . .	17
A.2	Kreslení obrázků . . . . .	18
A.3	Tabulky . . . . .	18
A.4	Odkazy v textu . . . . .	19
A.4.1	Odkazy na literaturu . . . . .	19
A.4.2	Odkazy na obrázky, tabulky a kapitoly . . . . .	21
A.5	Rovnice, centrovaná, číslovaná matematika . . . . .	21
A.6	Kódy programu . . . . .	22
A.7	Další poznámky . . . . .	22
A.7.1	České uvozovky . . . . .	22
<b>B</b>	<b>Seznam použitých zkratk</b>	<b>23</b>
<b>C</b>	<b>UML diagramy</b>	<b>25</b>
<b>D</b>	<b>Instalační a uživatelská příručka</b>	<b>27</b>
<b>E</b>	<b>Obsah přiloženého CD</b>	<b>29</b>



# Seznam obrázků

1.1	Ukázková scéna (nahore) a její planární segmentace (dole) . . . . .	2
2.1	Definice okolí bodu (boční pohled) (a) zobrazuje proložení původní roviny body uvnitř koule. (b) znázorňuje výslednou rovinu po iterativním procesu, který zahrnuje váhy jednotlivých bodů. Výsledné okolí tvoří modře vyznačené body uvnitř bufferu.	6
2.2	Diagram aktivit znázorňující proces definice okolí bodu . . . . .	7
A.1	Popiska obrázku . . . . .	18
E.1	Seznam přiloženého CD — příklad . . . . .	29



# Seznam tabulek

A.1 Ukázka tabulky . . . . .	19
------------------------------	----



# Kapitola 1

## Úvod

Rekonstrukce 3D modelů se dnes uplatňuje v mnoha oborech. Běžně se můžeme setkat např. s vizualizacemi, které se často využívají pro simulaci vzhledu města při výstavbě nových objektů. V takovém případě se obvykle naskenuje aktuální stav okolí výstavby, do kterého se přidá model plánované budovy. Výsledná vizualizace poté umožňuje posoudit vliv nové výstavby v kontextu okolí. Výhoda je také v tom, že takovéto simulace snadno pochopí i laický pozorovatel. Využití lze nalézt i v technické praxi, kde mohou 3D modely zachytit aktuální stav zařízení (např. potrubí v elektrárně), které nemusí odpovídat zastaralé či nekompletní dokumentaci. Přesné 3D modely krajiny se mohou využívat např. při modelování záplav nebo šíření bezdrátového signálu. Podobně se modely mohou využívat k modelování šíření zvuku v uzavřených prostorech. Dále můžeme najít rozsáhlé využití v archeologii a ochraně kulturního dědictví, v dokumentaci důlních děl, monitorování krajiny pro detekci nebezpečných sesuvů, mapování pobřežních oblastí a mořského dna v okolí přístavů apod.

Existuje několik metod, které se ke sběru těchto dat využívají. Tradiční metodou je fotogrammetrie, kdy se informace o objektu získávají z několika fotografií a pro případné zpřesnění se může využít geodetického zaměření. Hlavní nevýhodou této metody je přesnost naměřených dat, která výrazně klesá se vzdáleností měření. Tato metoda nám obvykle poskytuje souřadnice charakteristických bodů, jako jsou hrany, vrcholy apod. To může být velká nevýhoda zejména u nepravidelných objektů.

Jednou z nejnovějších metod sběru dat je laserové skenování. Jeho hlavní výhodou je rychlý sběr velkého množství přesných dat v terénu. Nevýhodou oproti předchozí metodě je zejména špatná identifikace hran a vrcholů. Další nevýhodou může být fakt, že přístroje pro laserové skenování jsou velmi drahé. Slabinou této metody je také náročné zpracování, které se neobejde bez výkonné výpočetní techniky. Základním výstupem z laserového skenování je mračno bodů. Jedná se o tisíce až miliony bodů, které jsou definovány třemi kartézskými souřadnicemi ( $x$ ,  $y$ ,  $z$ ), dále mohou obsahovat informaci o barvě ( $r$ ,  $g$ ,  $b$ ) a případně také normálu ( $n_x$ ,  $n_y$ ,  $n_z$ ) k ploše, na které se bod vyskytuje.

## 1.1 Segmentace mračna bodů

V této práci se zabývám zpracováním mračna bodů, konkrétně jeho planární segmentací. Cílem práce je tedy identifikovat ve vstupním mračnu bodů rovinné útvary, přiřadit jednotlivé body do těchto rovinných útvarů a případně zahodit body, které v žádné rovině neleží (viz Obr. 1.1). Tato segmentace má smysl zejména v architektuře a archeologii, tedy při rekonstrukci budov a podobných objektů, které jsou složeny z relativně malého počtu velkých rovin (tedy stěny, střechy apod.).



Obrázek 1.1: Ukázková scéna (nahore) a její planární segmentace (dole)

Segmentační metody můžeme hrubě rozdělit do dvou základních kategorií [8]. Jednak jsou to metody, které segmentují na základě vlastností jako je vzdálenost bodů v prostoru a případně podobnost lokálně odhadnutých normál. Sem spadají např. metody, jako je segmentace na základě skenovacích linií<sup>1</sup> nebo surface growing. První zmíněná metoda vychází z toho, že data jsou pořizována postupně podél skenovacích linií. Body v těchto liniích jsou nejprve rozděleny do rovných přímek a poté jsou ve 3D prostoru na základě podobných atributů slučovány do jednotlivých segmentů. Surface growing algoritmy fungují tak, že je vybrán rovinný či nerovinný prvek (seed region) a ten je poté postupně spojován s blízkými body, které mají podobné atributy. Tato metoda je ale silně závislá na volbě vhodného původního prvku.

---

<sup>1</sup>V anglické literatuře se označuje jako `scan line segmentation` [8]



Do druhé kategorie spadají metody, které přímo odhadují parametry roviny na základě shlukování bodů a vyhledávání lokálních maxim v prostoru parametrů. Sem patří např. rozšíření Houghovi transformace pro 3D prostor. Jeden bod v mračnu, který se nachází na ploše v objektovém prostoru definuje rovinu v prostoru parametrů. Body na stejné ploše mají poté podobné parametry (vzdálenost od počátku a naklonění rovinny) a na základě těchto parametrů jsou poté shlukovány. Jednotlivé shluky bodů pak podle dalších kritérií vytvoří jeden segment. Tato a další podobné metody mají zejména výpočetní problémy, protože mají velké paměťové nároky. Obvyklým problémem obou výše zmíněných kategorií je fakt, že jednotlivé metody se často zaměřují pouze na specifický typ vstupních dat (letecké skeny nebo pozemní skeny).

Článek [2] navrhuje řešení, které zohledňuje jak podobnost bodů v prostoru atributů, tak vzdálenost bodů v objektovém prostoru. Zároveň snižuje počet atributů pro vyšší efektivitu a menší paměťové nároky. Dále by také metoda neměla být závislá na typu vstupních dat.

## 1.2 Struktura práce

Tato práce nejprve popisuje zadané algoritmy [2] a [5] a věnuje se návrhu obou implementací. Poté následuje část popisující samotnou implementaci algoritmů. Algoritmy jsou implementovány v rámci dodaného nástroje ArchiRec3D [6], který je psaný v jazyku Java. V závěrečné části práce je otestována správná funkčnost obou algoritmů na uměle generovaných datech a na reálných datech je otestována kvalita algoritmů a jejich implementací. V závěru jsou shrnuty výsledky testování a analyzovány nedostatky algoritmů. Je zda také zhodnoceno, zda jsou nedostatky způsobeny návrhem algoritmu nebo jeho implementací.



## Kapitola 2

# Analýza a návrh řešení

### 2.1 Segmentace na základě velikosti normálového vektoru

Navržený proces segmentace v článku [2] zahrnuje tři hlavní kroky - definice okolí, výpočet atributů a clustering bodů. Definice okolí bodu bere v úvahu 3D vzdálenost mezi body a tvar povrchu, na kterém se bod nachází. Atributy daného bodu jsou poté vypočteny na základě tohoto definovaného okolí, pro nižší paměťové nároky jsou pro každý bod použity pouze dva parametry. Po vypočtení parametrů je proveden clustering bodů ležících na stejné ploše, který zohledňuje jak podobnost v prostoru atributů, tak vzdálenost bodů v prostoru.

#### 2.1.1 Adaptivní válcová definice okolí

Správná definice okolí bodu je zásadní podmínkou pro správnou funkčnost algoritmu, protože okolí bodu přímo ovlivňuje parametry, které jsou pro daný bod vypočteny. V tomto algoritmu je okolí definováno tak, že bere v úvahu vzdálenost bodů v prostoru, ale zároveň také tvar povrchu, na kterém se bod nachází. To znamená, že do okolí jsou zahrnuty pouze takové body, které jsou prostorově blízko, ale zároveň leží na stejné ploše.

Proces definice okolí je znázorněn v diagramu na obrázku 2.2. Prvním krokem je vytvořit kouli se středem v bodu, pro který okolí definujeme. Poloměr koule volíme takový, aby koule obsahovala dostatečný počet bodů pro spolehlivý výpočet parametrů. Tuto hodnotu nejde přesněji specifikovat, musí být odvozena z konkrétního mračka bodů, které zpracováváme<sup>1</sup>. Body, které leží uvnitř této koule, proložíme pomocí metody nejmenších čtverců výchozí rovinou (viz Obr. 2.1). Poté, co je rovina proložena, vypočítáme pro každý bod v kouli jeho vzdálenost od této roviny. Inverzi této vzdálenosti použijeme v další iteraci jako váhu bodu<sup>2</sup>, jak můžeme vidět v rovnici 2.1:

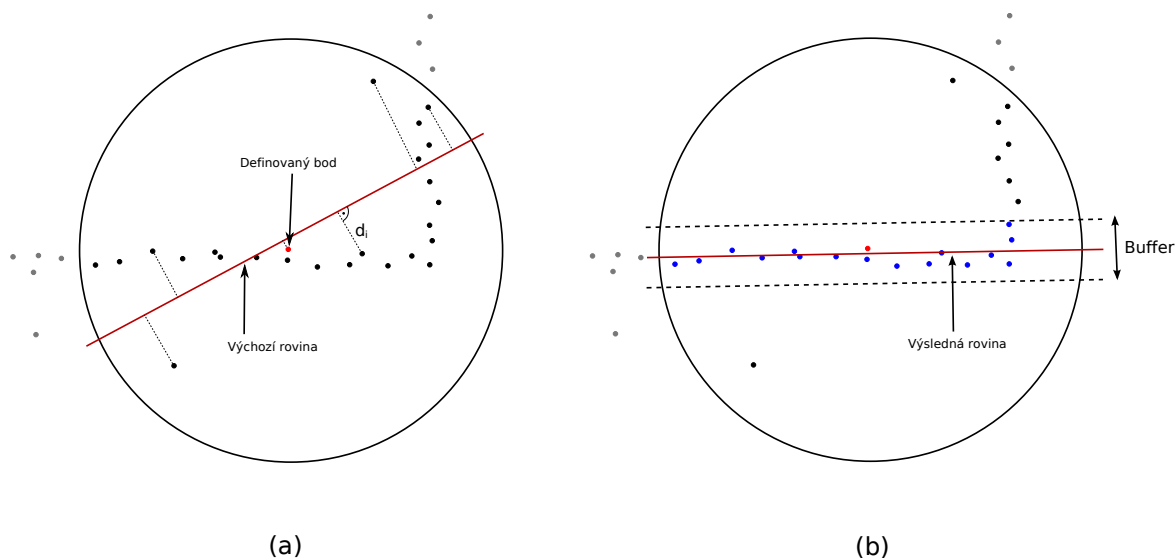
$$p_i = 1/d_i \quad (2.1)$$

kde  $p_i$  značí váhu  $i$ -tého bodu a  $d_i$  je vzdálenost bodu od proložené roviny. Tento proces převažování roviny se iterativně opakuje, dokud se roviny v rámci iterací již nemění nebo

<sup>1</sup>Nicméně praxe ukazuje, že vhodná volba jsou řádově desítky až stovky bodů uvnitř koule.

<sup>2</sup>Jednoduše řečeno, čím je bod od plochy dál, tím je jeho váha nižší.

dokud není dosaženo daného počtu iterací (např. 10). Poté je rovnoběžně nad a pod výslednou rovinou definován buffer. Velikost tohoto bufferu je závislá na očekávané velikosti šumu ve vstupních datech. Body, které jsou uvnitř tohoto bufferu, tvoří výsledné okolí (viz Obr. 2.1(b)). Ostatní body mimo buffer se již na okolí nijak nepodílejí.

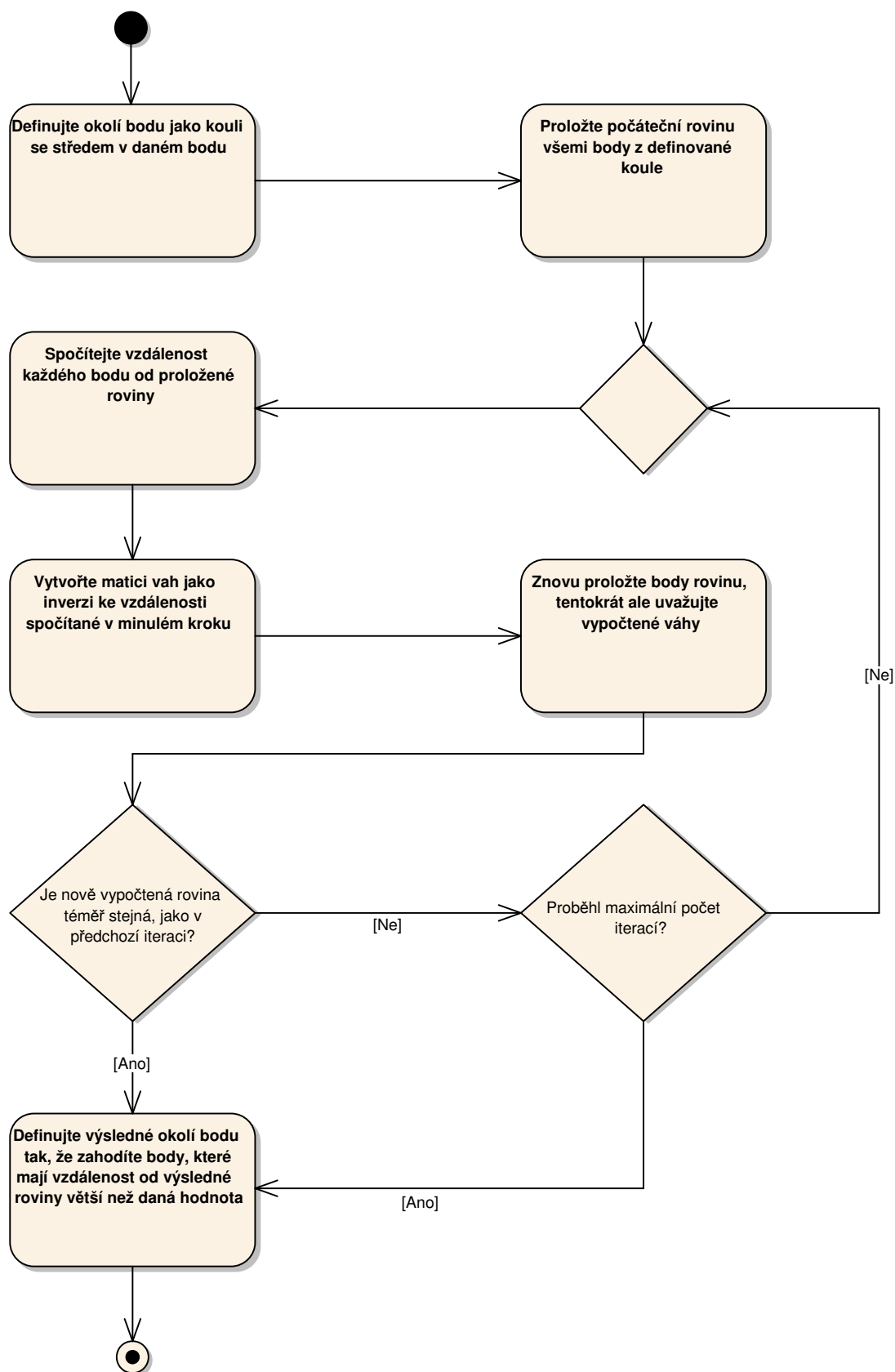


Obrázek 2.1: Definice okolí bodu (boční pohled)

(a) zobrazuje proložení původní roviny body uvnitř koule.

(b) znázorňuje výslednou rovinu po iterativním procesu, který zahrnuje váhy jednotlivých bodů. Výsledné okolí tvoří modře vyznačené body uvnitř bufferu.

Výsledné okolí se tedy nachází v útvaru, který téměř odpovídá nízkému válci, jehož osa je normála k ploše, na které se původní bod nachází. Tato osa navíc může být v průběhu algoritmu oproti výchozí hodnotě značně upravena, proto je tato metoda pojmenována adaptivní válcová definice okolí.



Obrázek 2.2: Diagram aktivit znázorňující proces definice okolí bodu

### 2.1.2 Výpočet parametrů pomocí okolí bodu

Lorem ipsum

## 2.2 Segmentace pomocí vyhledávání dominantních os

[\[5\]](#)

## Kapitola 3

# Implementace

Popis implementace/realizace se zaměřením na nestandardní části řešení.





## Kapitola 4

# Testování

- Způsob, průběh a výsledky testování.
- Srovnání s existujícími řešeními, pokud jsou známy.



## Kapitola 5

### Závěr

- Zhodnocení splnění cílů DP/BP a vlastního přínosu práce (při formulaci je třeba vzít v potaz zadání práce).
- Diskuse dalšího možného pokračování práce.



# Literatura

- [1] HAINDL, M. – KMENT, Ľ. – SLAVÍK, P. Virtual Information Systems. In *WSCG'2000 — Short communication papers*, s. 22–27. University of West Bohemia, Pilsen, 2000.
- [2] KIM, C. et al. Segmentation of Laser Scanning Data using Approach based on Magnitude of Normal Vector. *ISPRS Journal of Photogrammetry and Remote Sensing*. 2012.
- [3] Příspěvatelé Wikipedie. *Framework* [online]. 2009. [cit. 10. 9. 2009]. Dostupné z: <http://cs.wikipedia.org/wiki/Framework>.
- [4] Příspěvatelé Wikipedie. *Object-relational mapping* [online]. 2009. [cit. 6. 12. 2009]. Dostupné z: [http://en.wikipedia.org/wiki/Object-relational\\_mapping](http://en.wikipedia.org/wiki/Object-relational_mapping).
- [5] REISNER-KOLLMANN, I. – LUKSCH, C. – SCHWÄRZLER, M. Reconstructing Buildings as Textured Low Poly Meshes from Point Clouds and Images. *EUROGRAPHICS*. 2011.
- [6] SEDLÁČEK, D. ArchiRec3D.  
<http://dcgi.felk.cvut.cz/cs/members/sedlad1/>, stav z.
- [7] SLAVÍK, P. Grammars and Rewriting Systems as Models for Graphical User Interfaces. *Cognitive Systems*. 1997, 4, 3/4, s. 381–399.
- [8] VOSSELMAN, G. et al. Recognising structure in laser scanner point clouds. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*. 2004.
- [9] web:cstug. CSTUG —  $\text{\LaTeX}$  Users Group — hlavní stránka.  
<http://www.cstug.cz/>, stav z 2. 3. 2009.
- [10] web:infodp. K336 Info — pokyny pro psaní diplomových prací.  
<https://info336.felk.cvut.cz/clanek.php?id=400>, stav ze 4. 5. 2009.
- [11] web:infogs. Knihovna Grafické skupiny.  
<http://www.cgg.cvut.cz/Bib/library/>, stav z 30. 8. 2001.
- [12] web:ipe. Grafický vektorový editor pro práce vhodné pro práci  $\text{\LaTeX}$ em.  
<http://tclab.kaist.ac.kr/ipe/>, stav z 4. 5. 2009.
- [13] web:latexdocweb.  $\text{\LaTeX}$  — online manuál.  
<http://www.cstug.cz/latex/lm/frames.html>, stav ze 4. 5. 2009.

- [14] web:latexwiki. Wiki Books L<sup>A</sup>T<sub>E</sub>X.  
<http://en.wikibooks.org/wiki/LaTeX/>, stav z 3.4.2009.
- [15] ŽÁRA, J. – BENEŠ, B. – FELKEL, P. *Moderní počítačová grafika*. Computer Press s.r.o, Brno, 1st edition, 1998. In Czech.

## Příloha A

# Pokyny a návody k formátování textu práce

Tato příloha samozřejmě nebude součástí vaší práce. Slouží pouze jako příklad formátování textu.

Používat se dají všechny příkazy systému L<sup>A</sup>T<sub>E</sub>X. Existuje velké množství volně přístupné dokumentace, tutoriálů, příruček a dalších materiálů v elektronické podobě. Výchozím bodem, kromě Googlu, může být stránka CSTUG (Czech Tech Users Group) [9]. Tam najdete odkazy na další materiály. Většinou dostačující a přehledně organizovanou elektronickou dokumentaci najdete například na [13] nebo [14].

Existují i různé nadstavby nad systémy T<sub>E</sub>X a L<sup>A</sup>T<sub>E</sub>X, které výrazně usnadní psaní textu zejména začátečníkům. Velmi rozšířený v Linuxovém prostředí je systém Kile.

### A.1 Vkládání obrázků

Obrázky se umísťují do plovoucího prostředí **figure**. Každý obrázek by měl obsahovat **název** (`\caption`) a **návěští** (`\label`). Použití příkazu pro vložení obrázku `\includegraphics` je podmíněno aktivací (načtením) balíku `graphicx` příkazem `\usepackage{graphicx}`.

Budete-li zdrojový text zpracovávat pomocí programu `pdflatex`, očekávají se obrázky s příponou `*.pdf`<sup>1</sup>, použijete-li k formátování `latex`, očekávají se obrázky s příponou `*.eps`.<sup>2</sup>

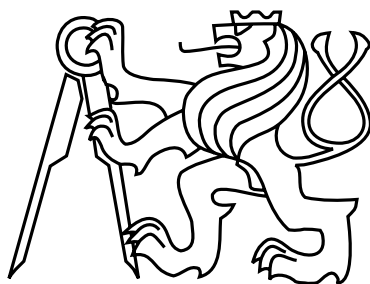
Příklad vložení obrázku:

```
\begin{figure}[h]
\begin{center}
\includegraphics[width=5cm]{figures/LogoCVUT}
```

---

<sup>1</sup>`pdflatex` umí také formáty PNG a JPG.

<sup>2</sup>Vzájemnou konverzi mezi snad všemi typy obrázku včetně změn velikosti a dalších vymožeností vám může zajistit balík ImageMagick (<http://www.imagemagick.org/script/index.php>). Je dostupný pod Linuxem, Mac OS i MS Windows. Důležité jsou zejména příkazy `convert` a `identify`.



Obrázek A.1: Popiska obrázku

```
\caption{Popiska obrazku}
\label{fig:logo}
\end{center}
\end{figure}
```

## A.2 Kreslení obrázků

Zřejmě každý z vás má nějaký oblíbený nástroj pro tvorbu obrázků. Jde jen o to, abyste dokázali obrázek uložit v požadovaném formátu nebo jej do něj konvertovat (viz předchozí kapitola). Je zřejmě vhodné kreslit obrázky vektorově. Celkem oblíbený, na ovládání celkem jednoduchý a přitom dostatečně mocný je například program Inkscape.

Zde stojí za to upozornit na kreslicí programe Ipe [12], který dokáže do obrázku vkládat komentáře přímo v latexovském formátu (vzroce, stejné fonty atd.). Podobné věci umí na Linuxové platformě nástroj Xfig.

Za pozornost ještě stojí schopnost editoru Ipe importovat obrázek (jpg nebo bitmap) a krelit do něj latexovské popisky a komentáře. Výsledek pak umí exportovat přímo do pdf.

## A.3 Tabulky

Existuje více způsobů, jak sázet tabulky. Například je možno použít prostředí `table`, které je velmi podobné prostředí `figure`.

Zdrojový text tabulky A.1 vypadá takto:

```
\begin{table}
\begin{center}
\begin{tabular}{|c|l|l|}
\hline
\textbf{DTD} & \textbf{construction} & \textbf{elimination} \\
\hline
 $\mid$  & \verb+in1|A|B a:sum A B+ & \verb+case([_:A]a)([_:B]a)ab:A+\\
```



DTD	construction	elimination
	$\text{in1} A B \text{ a:sum } A \ B$ $\text{in1} A B \text{ b:sum } A \ B$	$\text{case}([\_ :A]a)([\_ :B]a)ab:A$ $\text{case}([\_ :A]b)([\_ :B]b)ba:B$
+	$\text{do\_reg:A} \rightarrow \text{reg } A$	$\text{undo\_reg:reg } A \rightarrow A$
*, ?	the same like   and + with $\text{empty\_el:empty}$	the same like   and + with $\text{empty\_el:empty}$
$R(a,b)$	$\text{make\_R:A} \rightarrow B \rightarrow R$	$a: R \rightarrow A$ $b: R \rightarrow B$

Tabulka A.1: Ukázka tabulky

```

&\verb+in1|A|B b:sum A B+ & \verb+case([\_ :A]b)([\_ :B]b)ba:B+\
\hline
$+$&\verb+do_reg:A -> reg A+&\verb+undo_reg:reg A -> A+\
\hline
$*,?& the same like $\mid$ and $+$ & the same like $\mid$ and $+$\
& with \verb+empty_el:empty+ & with \verb+empty_el:empty+\
\hline
R(a,b) & \verb+make_R:A->B->R+ & \verb+a: R -> A+\
& & \verb+b: R -> B+\
\hline
\end{tabular}
\end{center}
\caption{Ukázka tabulky}
\label{tab:tab1}
\end{table}
\begin{table}

```

## A.4 Odkazy v textu

### A.4.1 Odkazy na literaturu

Jsou realizovány příkazem `\cite{odkaz}`.

Seznam literatury je dobré zapsat do samostatného souboru a ten pak zpracovat programem `bibtex` (viz soubor `reference.bib`). Zdrojový soubor pro `bibtex` vypadá například takto:

```

@Article{Chen01,
  author   = "Yong-Sheng Chen and Yi-Ping Hung and Chiou-Shann Fuh",
  title    = "Fast Block Matching Algorithm Based on
              the Winner-Update Strategy",
  journal  = "IEEE Transactions On Image Processing",
  pages    = "1212--1222",

```

```

volume = 10,
number = 8,
year = 2001,
}

@Misc{latexdocweb,
  author = "",
  title = "{\LaTeX} --- online manuál",
  note = "\verb|http://www.cstug.cz/latex/lm/frames.html|",
  year = "",
}
...

```

**Pozor:** Sazba názvů odkazů je dána BibTeX stylem (`\bibliographystyle{abbrv}`). BibTeX tedy obvykle vysází velké pouze počáteční písmeno z názvu zdroje, ostatní písmena zůstanou malá bez ohledu na to, jak je napíšete. Přesněji řečeno, styl může zvolit pro každý typ publikace jiné konverze. Pro časopisecké články třeba výše uvedené, jiné pro monografie (u nich často bývá naopak velikost písmen zachována).

Pokud chcete BibTeXu napovědět, která písmena nechat bez konverzí (viz `title = "{\LaTeX} --- online manuál"` v předchozím příkladu), je nutné příslušné písmeno (zde celé makro) uzavřít do složených závorek. Pro přehlednost je proto vhodné celé parametry uzavírat do uvozovek (`author = "..."`), nikoliv do složených závorek.

Odkazy na literaturu ve zdrojovém textu se pak zapisují:

```

Podívejte se na \cite{Chen01},
další detaily najdete na \cite{latexdocweb}

```

Vazbu mezi soubory `*.tex` a `*.bib` zajistíte příkazem `\bibliography{}` v souboru `*.tex`. V našem případě tedy zdrojový dokument `thesis.tex` obsahuje příkaz `\bibliography{reference}`.

Zpracování zdrojového textu s odkazy se provede postupným voláním programů `pdflatex <soubor>` (případně `latex <soubor>`), `bibtex <soubor>` a opět `pdflatex <soubor>`.<sup>3</sup>

Níže uvedený příklad je převzat z dříve existujících pokynů studentům, kteří dělají svou diplomovou nebo bakalářskou práci v Grafické skupině.<sup>4</sup> Zde se praví:

...

---

<sup>3</sup>První volání `pdflatex` vytvoří soubor s koncovkou `*.aux`, který je vstupem pro program `bibtex`, pak je potřeba znovu zavolat program `pdflatex` (`latex`), který tentokrát zpracuje soubory s příponami `.aux` a `.tex`. Informaci o případných nevyřešených odkazech (cross-reference) vidíte přímo při zpracovávání zdrojového souboru příkazem `pdflatex`. Program `pdflatex` (`latex`) lze volat vícekrát, pokud stále vidíte nevyřešené závislosti.

<sup>4</sup>Několikrát jsem byl upozorněn, že web s těmito pokyny byl zrušen, proto jej zde přímo necituji. Nicméně příklad sám o sobě dokumentuje obecně přijímaný konsensus ohledně citací v bakalářských a diplomových pracích na KP.

j) Seznam literatury a dalších použitých pramenů, odkazy na WWW stránky, ...  
 Pozor na to, že na veškeré uvedené prameny se musíte v textu práce odkazovat -- [1].

Pramen, na který neodkazujete, vypadá, že jste ho vlastně nepotřebovali a je uveden jen do počtu. Příklad citace knihy [1], článku v časopise [2], stati ve sborníku [3] a html odkazu [4]:

- [1] J. Žára, B. Beneš;, and P. Felkel.  
 Moderní počítačová grafika. Computer Press s.r.o, Brno, 1 edition, 1998.  
 (in Czech).
- [2] P. Slavík. Grammars and Rewriting Systems as Models for Graphical User Interfaces. Cognitive Systems, 4(4--3):381--399, 1997.
- [3] M. Haindl, Š. Kment, and P. Slavík. Virtual Information Systems.  
 In WSCG'2000 -- Short communication papers, pages 22--27, Pilsen, 2000.  
 University of West Bohemia.
- [4] Knihovna grafické skupiny katedry počítačů:  
<http://www.cgg.cvut.cz/Bib/library/>

... abychom výše citované odkazy skutečně našli v (automaticky generovaném) seznamu literatury tohoto textu, musíme je nyní alespoň jednou citovat: Kniha [15], článek v časopisu [7], příspěvek na konferenci [1], www odkaz [11].

Ještě přidáme další ukázkou citací online zdrojů podle české normy. Odkaz na wiki o frameworkch [3] a ORM [4]. Použití viz soubor `reference.bib`. V seznamu literatury by nyní měly být živé odkazy na zdroje. V `reference.bib` je zcela nový typ publikace. Detaily dohledal a dodal Petr Dlouhý v dubnu 2010. Podrobnosti najdete ve zdrojovém souboru tohoto textu v komentáři u příkazu `\thebibliography`.

#### A.4.2 Odkazy na obrázky, tabulky a kapitoly

- Označení místa v textu, na které chcete později čtenáře práce odkázat, se provede příkazem `\label{navesti}`. Lze použít v prostředích `figure` a `table`, ale též za názvem kapitoly nebo podkapitoly.
- Na návěští se odkážeme příkazem `\ref{navesti}` nebo `\pageref{navesti}`.

### A.5 Rovnice, centrováná, číslovaná matematika

Jednoduchý matematický výraz zapsaný přímo do textu se vysází pomocí prostředí `math`, resp. zkrácený zápis pomocí uzavření textu rovnice mezi znaky `$`.

Kód `$ S = \pi * r^2 $` bude vysázen takto:  $S = \pi * r^2$ .

Pokud chcete nečíslované rovnice, ale umístěné centrovane na samostatné řádky, pak lze použít prostředí `displaymath`, resp. zkrácený zápis pomocí uzavření textu rovnice mezi znaky `$$`. Zdrojový kód: `$$$ S = \pi * r^2 $$$` bude pak vysázen takto:

$$S = \pi * r^2$$

Chcete-li mít rovnice číslované, je třeba použít prostředí `equation`. Kód:

```
\begin{equation}
  S = \pi * r^2
\end{equation}
```

```
\begin{equation}
  V = \pi * r^3
\end{equation}
```

je potom vysázen takto:

$$S = \pi * r^2 \quad (\text{A.1})$$

$$V = \pi * r^3 \quad (\text{A.2})$$

## A.6 Kódy programu

Chceme-li vysázet například část zdrojového kódu programu (bez formátování), hodí se prostředí *verbatim*:

```
(* nickname2 *)
Lego> Refine in1
      (do_reg (nickname1 h));
Refine by in1 (do_reg (nickname1 h))
  ?4 : pcddata
  ?5 : pcddata
      (* surname2 *)
Lego> Refine surname1 h;
Refine by surname1 h
  ?5 : pcddata
      (* email2 *)
Lego> Refine undo_reg (email1 h);
Refine by undo_reg (email1 h)
*** QED ***
```

## A.7 Další poznámky

### A.7.1 České uvozovky

V souboru `k336_thesis_macros.tex` je příkaz `\uv{}` pro sázení českých uvozovek. ``Text` uzavřený do českých uvozovek.

## Příloha B

# Seznam použitých zkratek

**2D** Two-Dimensional

**ABN** Abstract Boolean Networks

**ASIC** Application-Specific Integrated Circuit

⋮



## Příloha C

# UML diagramy

Tato příloha není povinná a zřejmě se neobjeví v každé práci. Máte-li ale větší množství podobných diagramů popisujících systém, není nutné všechny umísťovat do hlavního textu, zvláště pokud by to snižovalo jeho čitelnost.





## Příloha D

# Instalační a uživatelská příručka

Tato příloha velmi žádoucí zejména u softwarových implementačních prací.



## Příloha E

# Obsah přiloženého CD

Tato příloha je povinná pro každou práci. Každá práce musí totiž obsahovat přiložené CD. Viz dále.

Může vypadat například takto. Váš seznam samozřejmě bude odpovídat typu vaší práce. (viz [10]):



Obrázek E.1: Seznam přiloženého CD — příklad

Na GNU/Linuxu si strukturu přiloženého CD můžete snadno vyrobit příkazem:

```
$ tree . >tree.txt
```

Ve vzniklém souboru pak stačí pouze doplnit komentáře.

Z **README.TXT** (případně index.html apod.) musí být rovněž zřejmé, jak programy instalovat, spouštět a jaké požadavky mají tyto programy na hardware.

Adresář **text** musí obsahovat soubor s vlastním textem práce v PDF nebo PS formátu, který bude později použit pro prezentaci diplomové práce na WWW.