

Zadání semestrální práce číslo XXXX

název semestrální práce

Josef Novák and Lukáš Novotný¹
Katedra počítačové grafiky a interakce,
Fakulta elektrotechnická, ČVUT Praha

Abstract

Tady bude zadání úlohy buď v češtině nebo angličtině.

Keywords: ray tracing, octrees, ...,

1. Úvod

Tady bude krátký úvod do semestrální práce, motivace a cíle.

2. Popis algoritmu

Zkrácený slovní popis algoritmu, je možné použít i pseudokódy, například balík Algorithm2e.sty, pak je nutné odkomentovat v tomto souboru řádek:

```
\usepackage{algorithm2e}.
```

Dokumentace k psaní pseudokódů na adrese:

<http://www.ctan.org/tex-archive/macros/latex/contrib/algorithm2e/algorithm2e.pdf>

Příklad pseudokódu algoritmu je uveden na obrázku 1:

Minimální délka popisu algoritmu 3/4 stránky.

3. Potíže při implementaci

Tady uvedete, co se Vám nedařilo a dařilo, s čím byly problémy a další záležitosti týkající se implementace semestrální práce, či problém čtení, konverze, či nalezení vhodných dat pro úlohu. Dále kde jste strávili nejvíce času, kterou chybu v implementaci jste nejdéle hledali a podobně. Časové nároky v hodinách (hodina = 60 minut) pro vypracování semestrální práce včetně tohoto reportu. Rozdělení práce mezi členy týmu, je-li úloha vypracována týmově.

¹A4M39DPG – Josef Novák a Lukáš Novotný, letní semestr 2009/10

```

1 begin
2   (entry distance, exit distance)  $\leftarrow$  intersect ray with root's AABB;
3   push ( tree root node, entry distance, exit distance) to stack ;
4   while stack is not empty do
5     (current node, entry distance, exit distance)  $\leftarrow$  pop stack;
6     while current node is not a leaf do
7       a  $\leftarrow$  current node's split axis;
8       t  $\leftarrow$  (current node's split value.a - ray origin.a) / ray dir.a;
9       (near, far)  $\leftarrow$  classify near/far with (split value.a > ray origin.a)
10      if t  $\geq$  exit distance or t < 0 then
11        | current node  $\leftarrow$  near;
12      else if t  $\leq$  entry distance then
13        | current node  $\leftarrow$  far;
14      else
15        | push ( far, t, exit distance) to stack;
16        | current node  $\leftarrow$  near;
17        | exit distance  $\leftarrow$  t;
18      end
19    end
20    if current node is not empty leaf then
21      | intersect ray with each object;
22      | if any intersection exists inside the leaf then
23        | return closest object to the ray origin;
24      | end
25    end
26  end
27  return no object intersected;
28 end

```

Algorithm 1: Recursive traversal based on ray origin classification. For direction based classification change line 10 to use (ray dir.a > 0) and remove condition (t < 0) from line 11.

model	resolution	no acceleration			n=12			
		#trav. steps [per ray]	#isect. tests [per ray]	time [ms]	#trav. steps [per ray]	#isect. tests [per ray]	time [ms]	speedup [%]
Bunny	$128 \times 128 \times 100$	76.39	529	66.4	82.59	348	59.4	10.5
Bunny	$256 \times 256 \times 200$	111.53	542	82.8	116.63	345	72.8	12.2
Bunny	$512 \times 508 \times 400$	185.86	472	123.4	191.37	292	108.9	11.7
Dragon	$128 \times 92 \times 60$	66.94	426	56.6	81.45	365	56.7	-0.2
Dragon	$256 \times 184 \times 116$	97.08	433	68.6	107.91	333	64.7	5.7
Dragon	$512 \times 364 \times 232$	15.81	377	96.5	166.93	264	87.0	9.9
Buddha	$56 \times 128 \times 56$	7.34	104	16.6	10.15	100	17.9	-7.9
Buddha	$108 \times 256 \times 108$	11.10	116	19.7	14.71	107	21.2	-7.6
Buddha	$212 \times 512 \times 212$	15.81	96	24.6	19.63	84	26.0	-5.9

Table 1: Příklad tabulky. Další popis i upřesnění parametrů může následovat v této legendě.

4. Naměřené výsledky

V této sekci budou tabulky, grafy, obrázky scén a text popisující naměřené výsledky. Nedílnou součástí této sekce bude komentář k naměřeným výsledkům, například, jestli naměřená data odpovídají teorii. Pro rychlost algoritmu je dobrá varianta uvádět čas zpracování na jeden dotaz, průměrný počet traverzačních kroků na jeden dotaz, průměrný počet incidenčních operací na jeden dotaz, dále pak dobu stavby datové struktury a to v závislosti na vstupních datech (scéně) a nastavení algoritmu (hloubka stromu, ukončující kritéria atd.). Je nutné uvést charakterizaci scény, alespoň počet primitiv, objektů ve scéně.

Dále je nezbytné uvést konfiguraci počítače, na kterém probíhalo testování, tedy typ procesoru (případně počet procesorů), taktovací frekvenci, počet jader použitých v implementaci, velikost vyrovnávací paměti (cache) procesoru, operační systém použitý během testování (32/64 bit), velikost hlavní paměti, typ a verzi kompilátoru, přepínače kompilátoru použité pro překlad programu pro účely měření výsledků a další.

Příklad formátování tabulky viz tabulka 1.

Příklad vložení souboru PDF je uveden na obrázku 1 a 2.

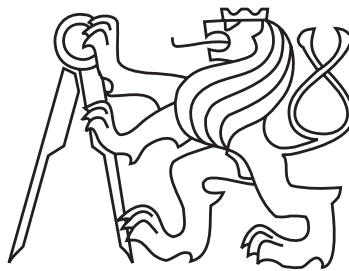


Figure 1: Příklad vloženého obrázku v PDF.

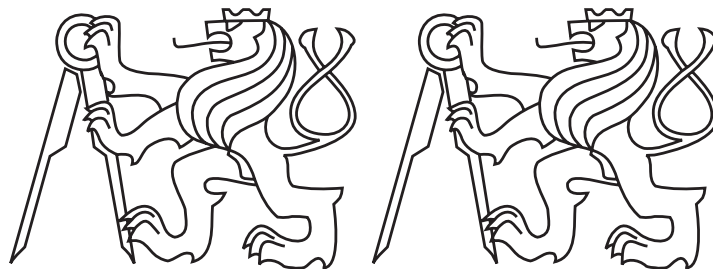


Figure 2: Příklad vložených obrázků v PDF vedle sebe.

5. Závěr

V této části uvedete závěr, doporučení pro opakování implementace této semestrální práce, omezení implementace či co se nepodařilo.

Poděkování

Práce je samostatná, ale pokud chcete někomu poděkovat například za to, že Vás poslouchal při předběžné prezentaci či za přečtení a kontrolu této zprávy a nalezení některých chyb, tak to tady uveďte.

References

- [1] A. Balvínová a M. Bílý. Textové informační systémy, sazecí systém \LaTeX , cvičení. Skripta ČVUT-FEL, 1995.
- [2] V. Dančo. Kapesní průvodce počítačovou typografií. Nakladatelství Labyrint, Praha 1995.
- [3] J. Rybička. \LaTeX pro začátečníky. Nakladatelství Konvoj, 1999.
- [4] D. Gruber. Kdo to má všechno číst, Nakladatelství Gruber, 1991. (ISBN 80-900680-1-4).
- [5] CSTUG: Úvod do \LaTeX u. <http://www.cstug.cz/latex/lm/frames.html>. Stránky z roku 2009.
- [6] CSTUG: Dokumentace a manuály k \LaTeX u. <http://www.cstug.cz/documentation/index.html>