

清华大学本科生综合论文训练答辩

多视点视频的并行实时解码

本科生：	卿培
学号：	2006011291
指导老师：	孙立峰 副教授

- 任务概述 & 任务目标
- MVC解码器性能优化
- 3D播放器设计与实现
- 存在的问题 & 未来工作方向



任务概述 & 任务目标

- MVC解码器性能优化

- 3D播放器设计与实现

- 存在的问题 & 未来工作方向

任务概述



- ◆ 基于现有的MVC Decoder进行性能优化
- ◆ 基于NVIDIA 3D Vision眼镜实现有3D效果的播放器

任务目标

- ✓ 修正MVC解码器算法，使得解码结果与参考软件一致
- ✓ 优化解码器性能，CPU实时解码720x576两路视频
- ✓ 设计实现一个3D播放器



任务概述 & 任务目标



MVC解码器性能优化



3D播放器设计与实现



存在的问题 & 未来工作方向

MVC Decoder性能优化

- 性能分析
- 单核优化
- 并行优化
- 实验结果

性能分析

- Visual Studio 2008 - Analysis -> Profile
- Intel VTune Performance Analyzer

Visual Studio分析结果

表 3.1 VS2008 性能分析报告 (前 10 项记录)

Function Name	Exclusive Time ^①	Exclusive Time %	Number of Calls ^②
macroblockPredGetDataUV	341.01	14.41	2,632,478
macroblockPredGetDataY	321.37	13.58	1,316,239
idct4x4_c	316.86	13.39	5,028,960
macroblockGetPred_axb	181.29	7.66	1,316,239
macroblockGetPred_axb_Bi	133.42	5.64	571,944
macroblockGetHalfPel	132.47	5.6	148,552
Filter	101.92	4.31	6,215,308
addMacroblockdata	85.36	3.61	184,171
iquant4x4_c	84.48	3.57	5,028,960
macroblockPBPrediction	70.95	3	184,171

① 不包括函数体内调用其他函数的净运行时间

② 函数被调用次数

VTune分析结果

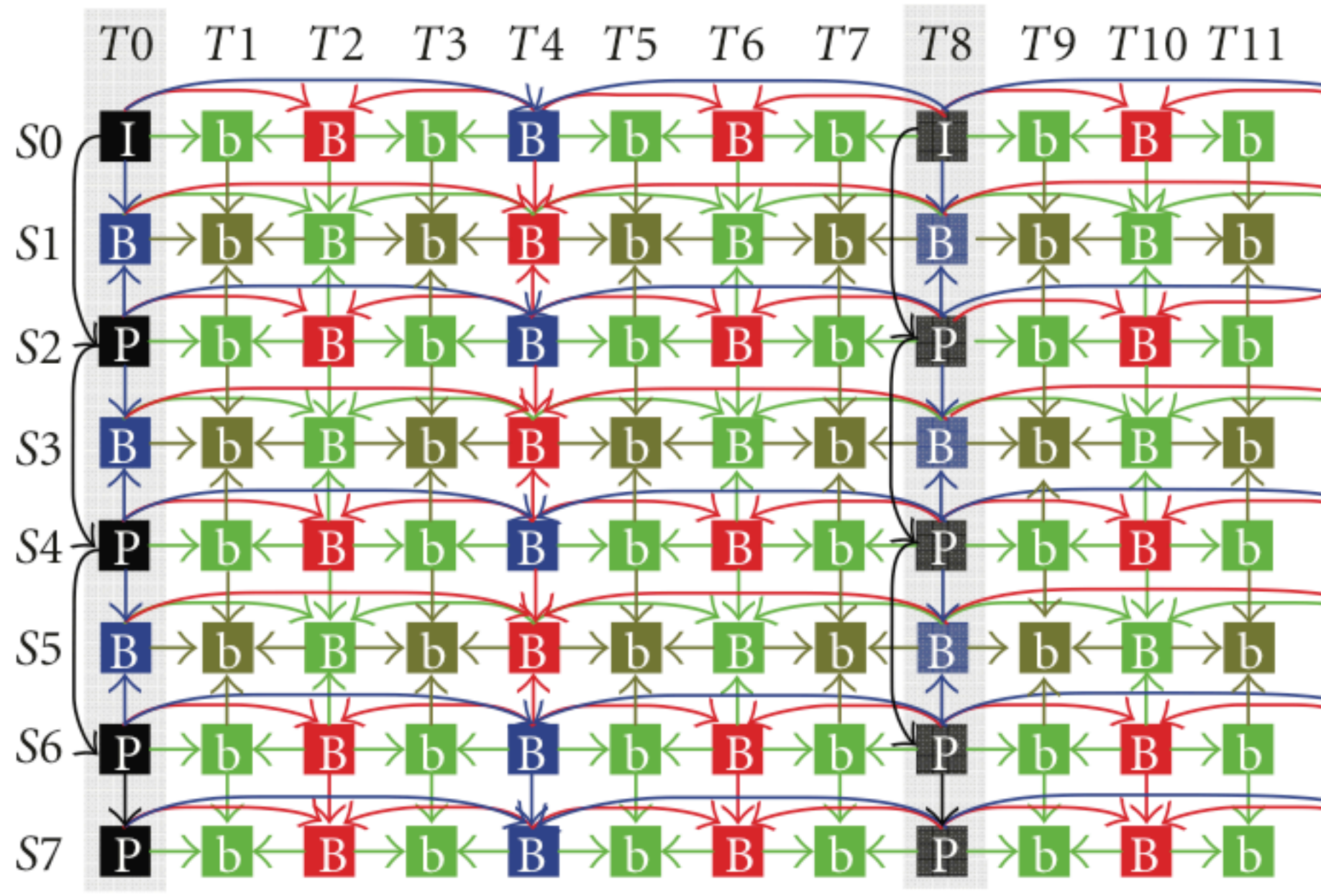
表 3.2 VTune 性能分析报告 (前 10 项记录)

Function	Calls	Self Time	% in function
macroblockPredGetDataUV	2632478	774863	0.61
macroblockPredGetDataY	1316239	702575	0.58
idct4x4_c	5028960	398076	1
macroblockInterDecode16x16_y	184171	328834	0.46
macroblockGetPred_axb	1316239	311730	0.11
FilterMB	210600	203506	0.21
Filter	6215308	191937	1
iquant4x4_c	5028960	157459	1
macroblockInterDecode_uv	184171	154697	0.48
macroblockPBPrediction	184171	150609	0.05

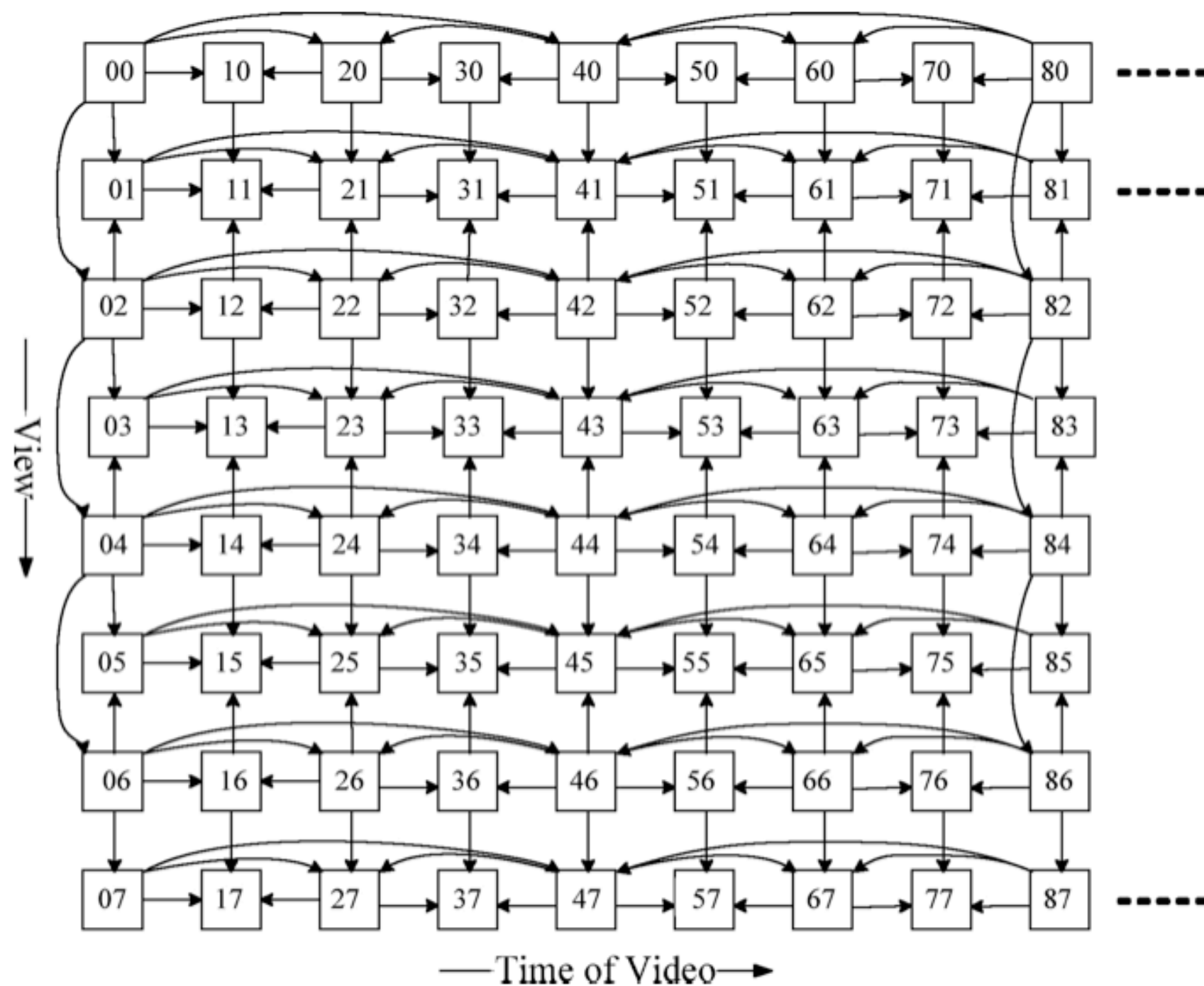
优化方案

- 重写函数逻辑
- 循环的优化
- 汇编优化
- 并行优化

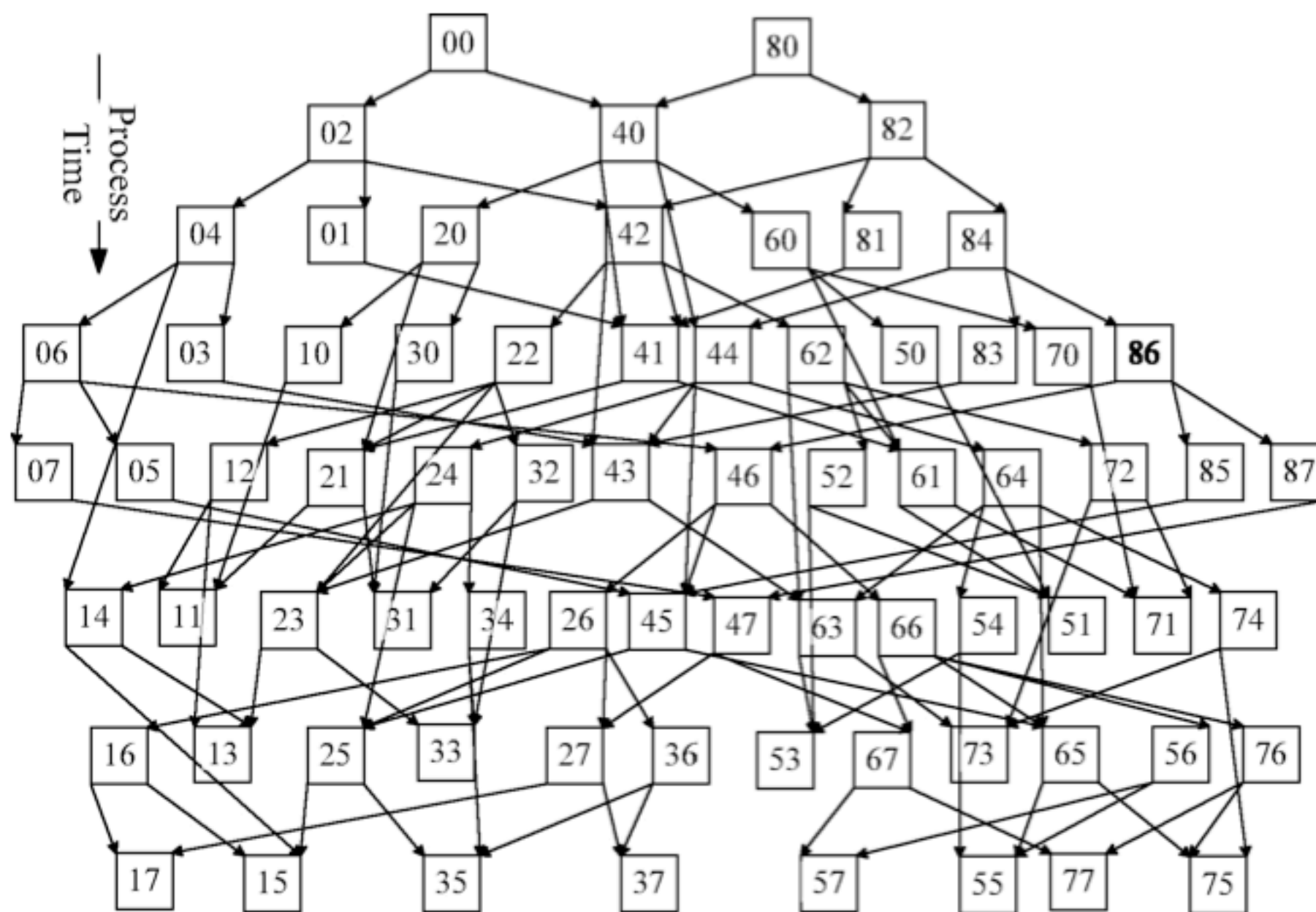
并行优化



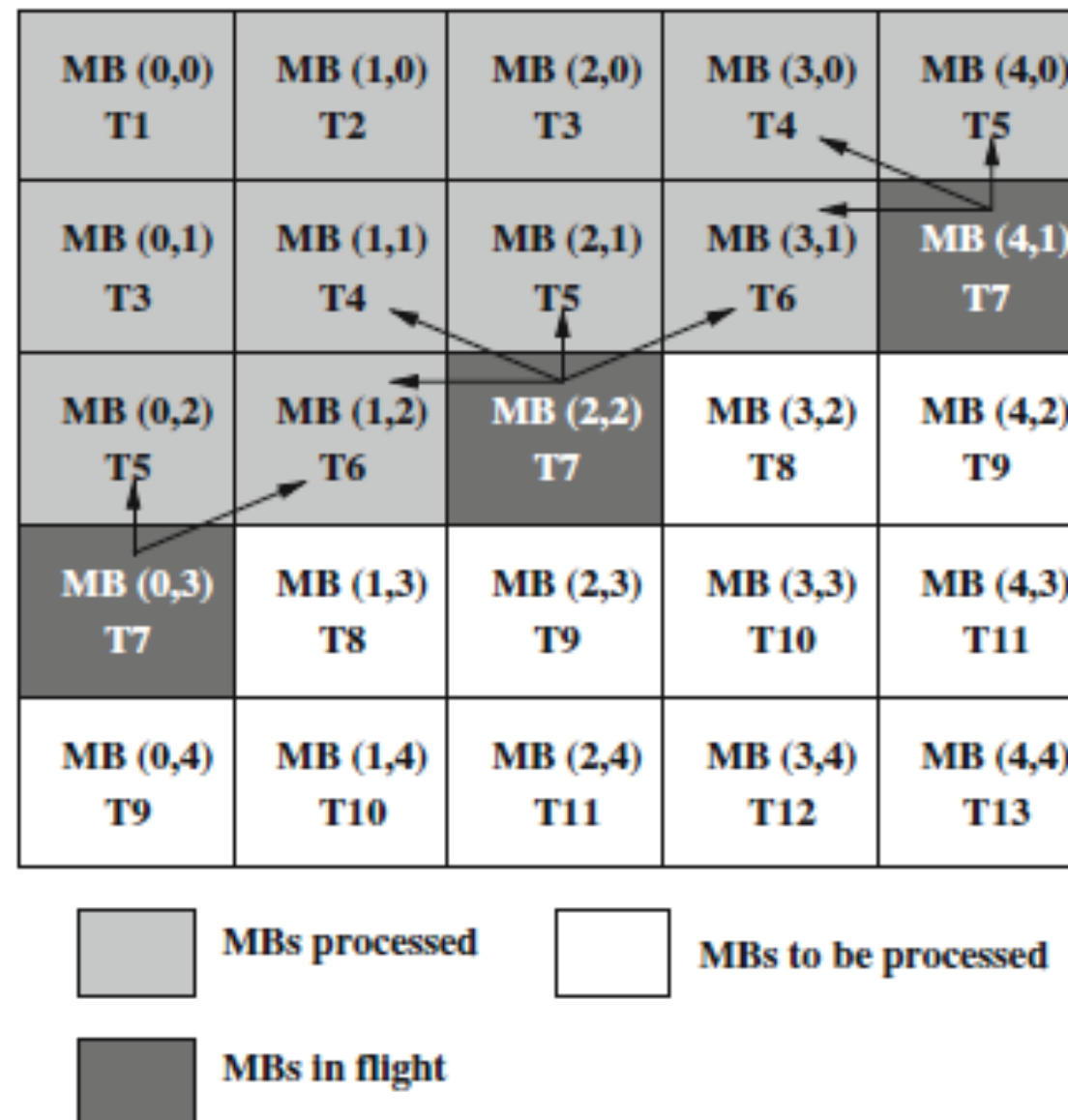
并行优化



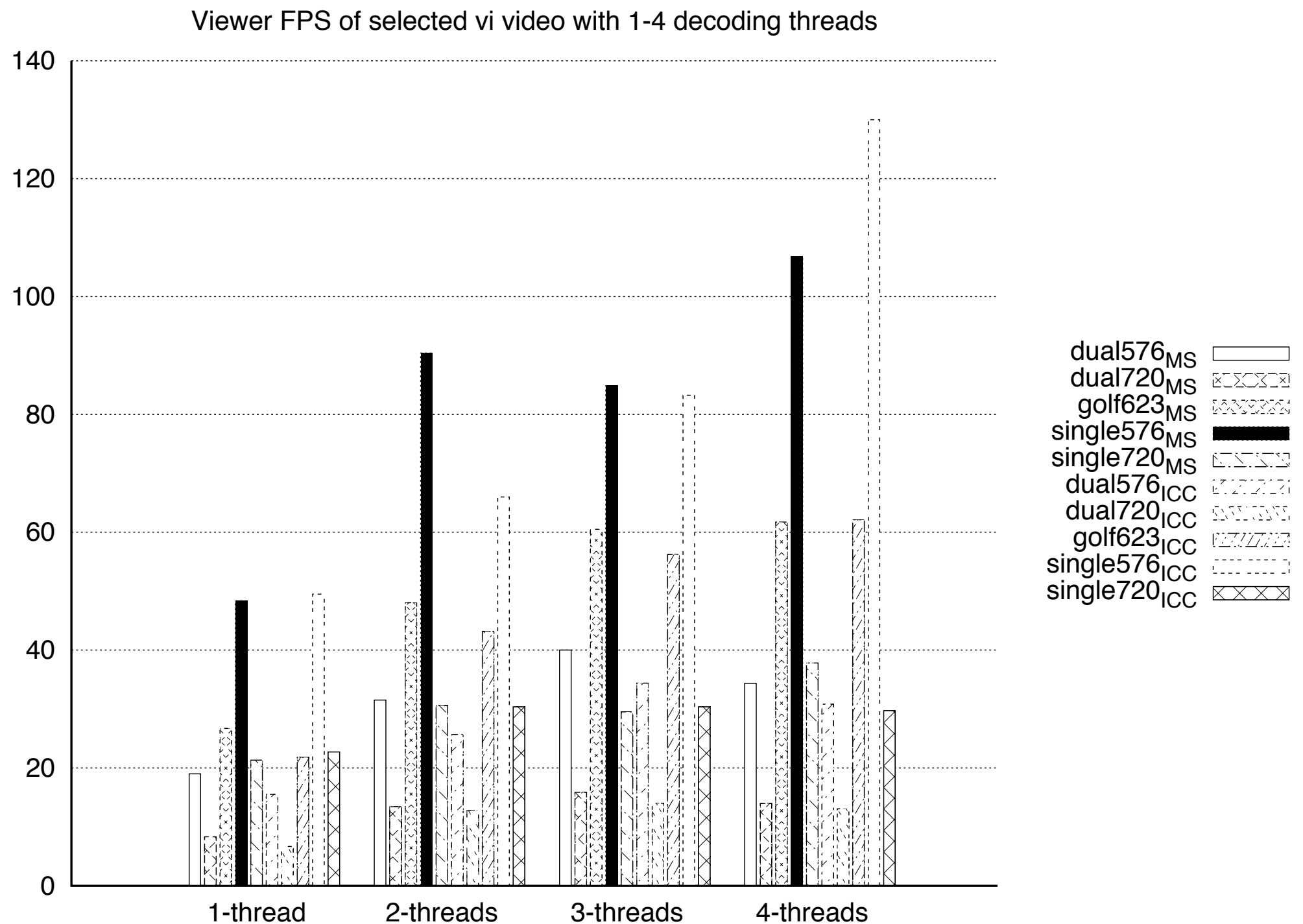
并行优化



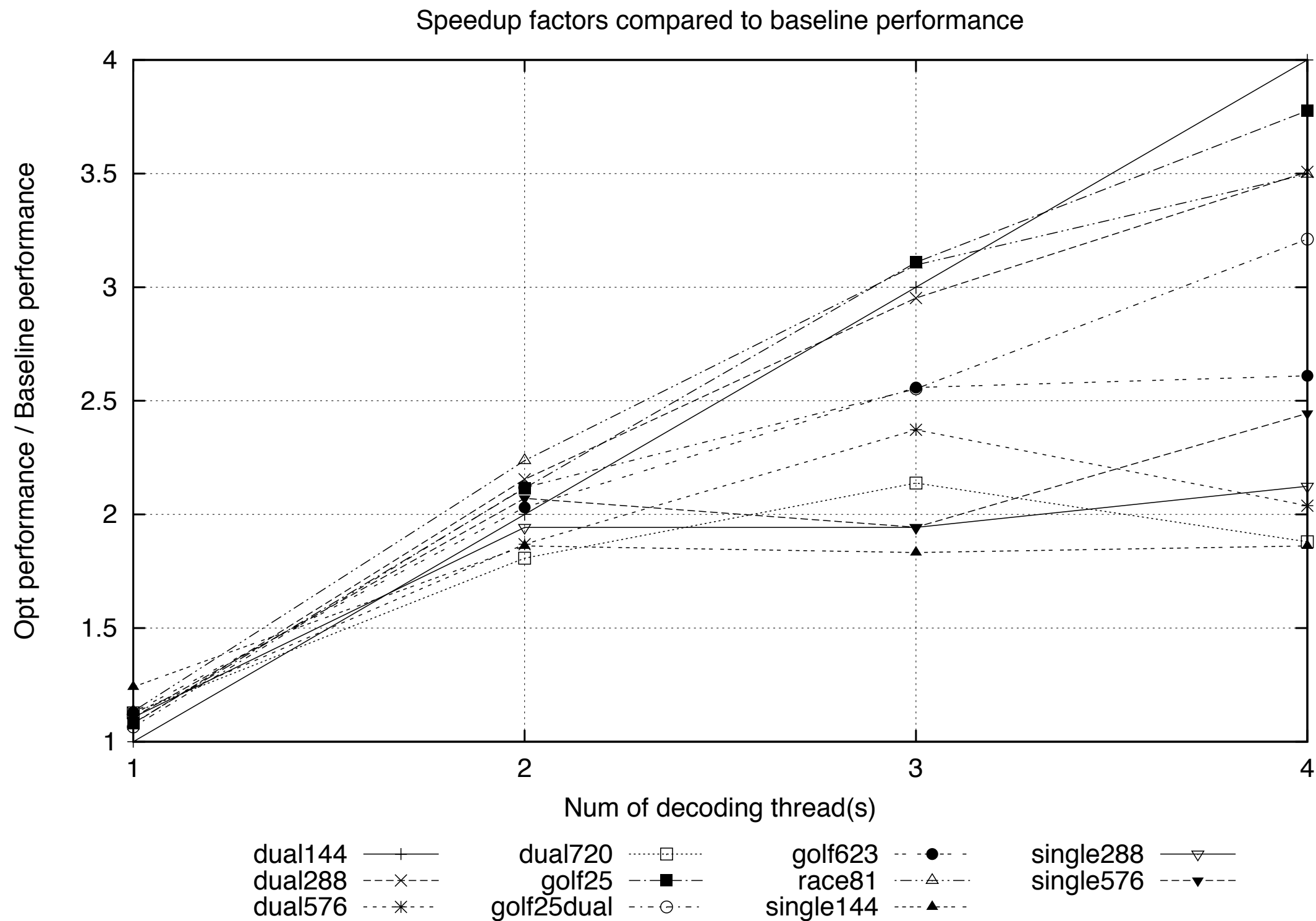
并行优化



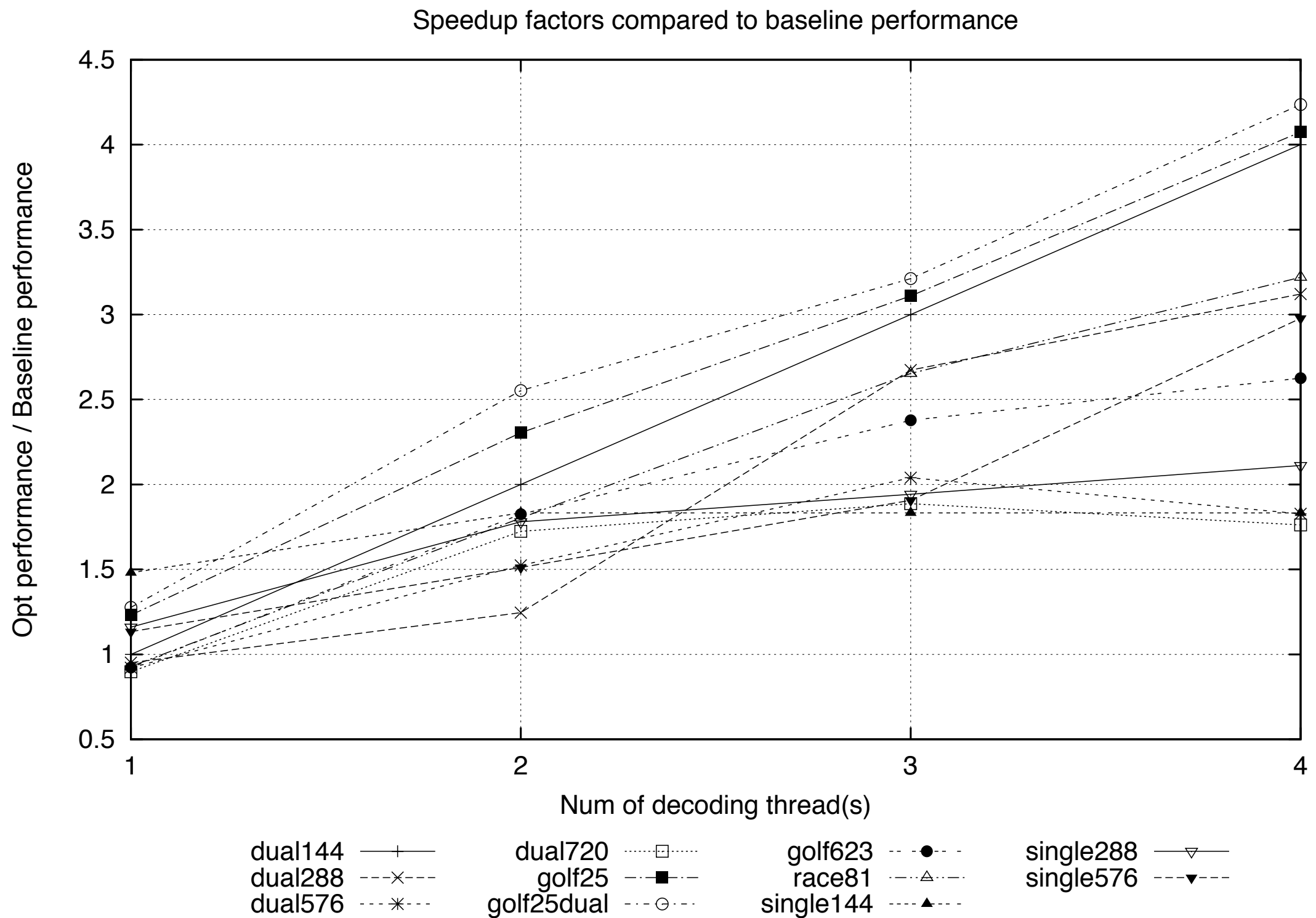
实验结果（解码帧率）



实验结果 (MS并行加速比)



实验结果（ICC并行加速比）



- ✓ 任务概述 & 任务目标
- ✓ MVC解码器性能优化
- ➡ 3D播放器设计与实现
- 存在的问题 & 未来工作方向

3D播放器设计与实现

- 硬件平台
- 可选方案
- 具体实现

硬件平台

- NVIDIA 3D Vision
- 快门式眼镜
- 120Hz显示器



可选方案

可选方案

- 查询NVAPI

可选方案

- 查询NVAPI

✗ 没有相应API Function

可选方案

- 查询NVAPI

✗ 没有相应API Function

- 以2xFPS播放+Enable3D程序运行

可选方案

- 查询NVAPI

 - ✗ 没有相应API Function

- 以2xFPS播放+Enable3D程序运行

 - ✗ 但见重影不见立体效果

可选方案

- 查询NVAPI

 - ✗ 没有相应API Function

- 以2xFPS播放+Enable3D程序运行

 - ✗ 但见重影不见立体效果

- 借助Direct3D

可选方案

- 查询NVAPI

 - ✗ 没有相应API Function

- 以2xFPS播放+Enable3D程序运行

 - ✗ 但见重影不见立体效果

- 借助Direct3D

 - ✓ 成功看到3D效果

具体实现

具体实现

IDirect3DSurface9

具体实现

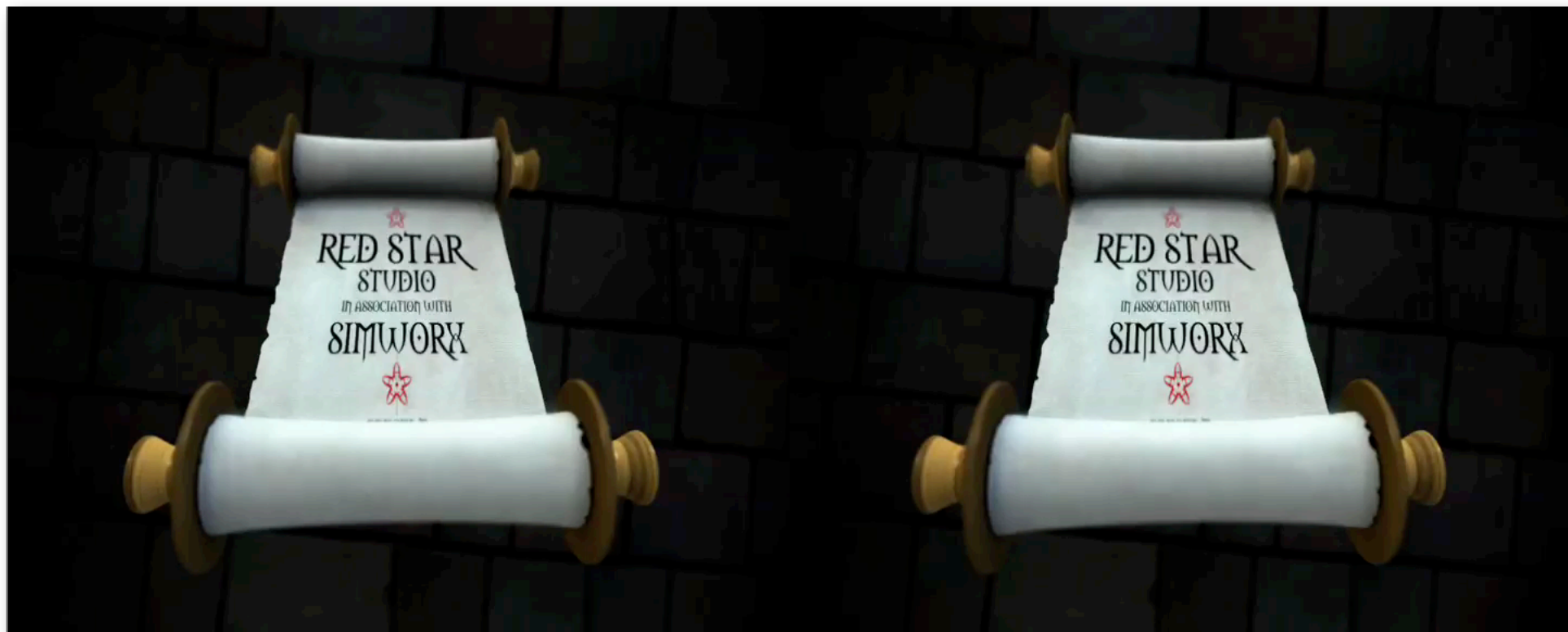
D3DXLoadSurfaceFromFile



具体实现

D3DXLoadSurfaceFromFile

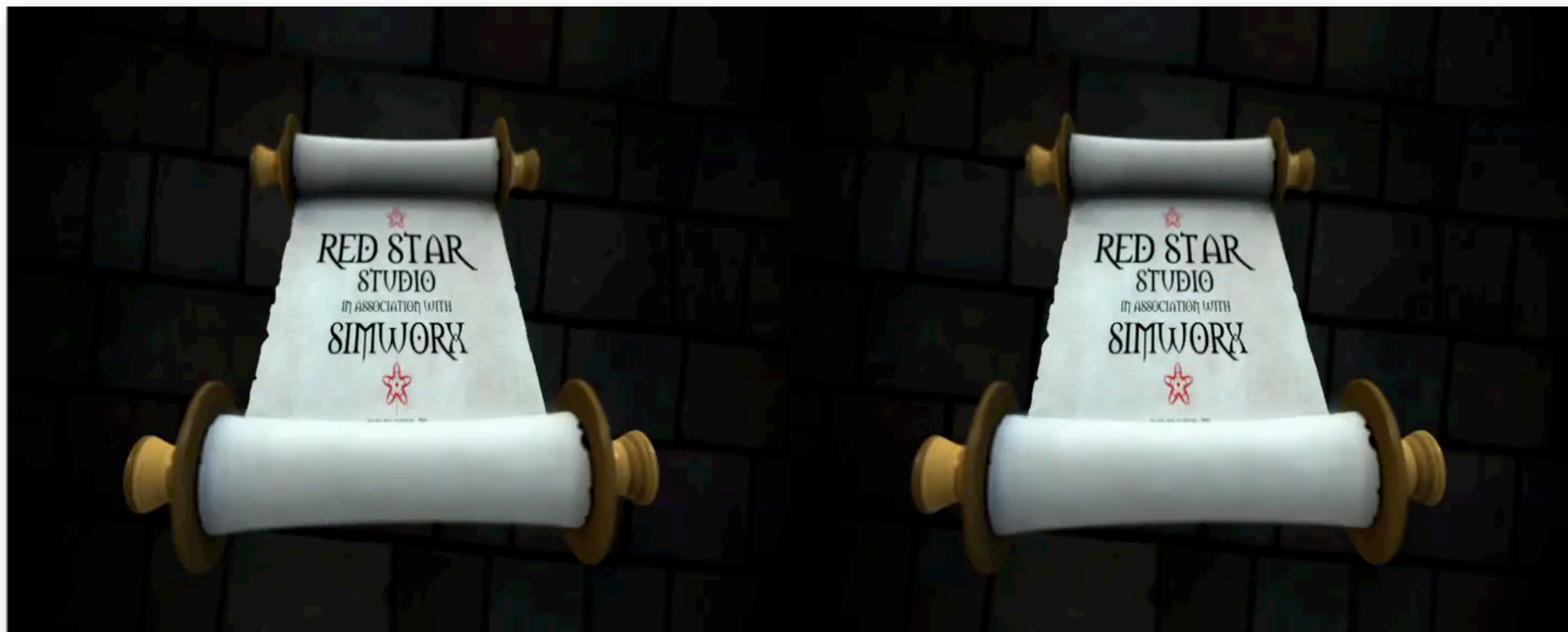
D3DXLoadSurfaceFromFile



具体实现

D3DXLoadSurfaceFromFile

D3DXLoadSurfaceFromFile



```
#define NVSTEREO_IMAGE_SIGNATURE 0x4433564e //NV3D
```

具体实现



具体实现



具体实现

具体实现

- Frame rate cap

具体实现

- Frame rate cap
 - Accurate timer

具体实现

- Frame rate cap
 - Accurate timer
 - QueryPerformanceCounter & QueryPerformanceFrequency

具体实现

- Frame rate cap
 - Accurate timer
 - QueryPerformanceCounter & QueryPerformanceFrequency
 - Retrieves the frequency of the high-resolution performance counter, if one exists. The frequency cannot change while the system is running. [Link to this doc](#)

具体实现

- Frame rate cap
 - Accurate timer
 - QueryPerformanceCounter & QueryPerformanceFrequency
 - Retrieves the frequency of the high-resolution performance counter, if one exists. The frequency cannot change while the system is running. [Link to this doc](#)
- $F_n = (t_n - t_0) * FPS$

具体实现

- Frame rate cap
 - Accurate timer
 - QueryPerformanceCounter & QueryPerformanceFrequency
 - Retrieves the frequency of the high-resolution performance counter, if one exists. The frequency cannot change while the system is running. [Link to this doc](#)
- $F_n = (t_n - t_0) * FPS$
- Frame skip

具体实现

Algorithm 1 Frame rate cap

```

1:  $\_start \leftarrow initialTime$ 
2:  $bufferd = FALSE$ 
3:  $lastRenderedFrame \leftarrow -1$ 
4: while  $TRUE$  do
5:    $\_current \leftarrow currentTime - \_start$ 
6:    $frameToRender \leftarrow \_current * FPS$ 
7:   if  $frameToRender + 1 > totalFrames$  then
8:      $break$  {break when reaching the last frame}
9:   end if
10:  if  $bufferd = FALSE$  then
11:    load  $Frame_{frameToRender+1}$  to memory
12:     $bufferd \leftarrow TRUE$ 
13:  end if
14:  if  $frameToRender \neq lastRenderedFrame$  then
15:    render  $Frame_{frameToRender}$ 
16:     $lastRendered \leftarrow frameToRender$ 
17:     $bufferd \leftarrow FALSE$ 
18:  end if
19: end while

```

实验结果

实验结果

- 功能

✓ 有3D效果

实验结果

- 功能

✓ 有3D效果

- 性能？

✗ 21fps

实验结果

- 功能

 - ✓ 有3D效果

- 性能?

 - ✗ 21fps

- 瓶颈

 - ➡ D3DXLoadSurfaceFromFile()

- ✓ 任务概述 & 任务目标
- ✓ MVC解码器性能优化
- ✓ 3D播放器设计与实现
- ➡ 存在的问题 & 未来工作方向

存在的问题

存在的问题

- 解码器
 - 多线程解码视频时的加速比不够稳定
 - 对 MVC 标准的支持尚不完整
 - 八路视频解码尚不能实时

存在的问题

- 解码器
 - 多线程解码视频时的加速比不够稳定
 - 对 MVC 标准的支持尚不完整
 - 八路视频解码尚不能实时
- 3D播放器
 - 图像序列以磁盘文件形式存储
 - 帧率达不到流畅要求

未来工作方向

未来工作方向

- 解码器
 - 多线程解码视频时的加速比不够稳定👉智能寻找最优线程数
 - 对 MVC 标准的支持尚不完整👉增加Main Profile和Stereo High Profile支持
 - 八路视频解码尚不能实时👉借助GPU进行transform运算

未来工作方向

- 解码器

- 多线程解码视频时的加速比不够稳定👉智能寻找最优线程数
- 对 MVC 标准的支持尚不完整👉增加Main Profile和Stereo High Profile支持
- 八路视频解码尚不能实时👉借助GPU进行transform运算

- 3D播放器

- 图像序列以磁盘文件形式存储👉内存数据、对外接口
- 帧率达不到流畅要求👉实现自行填充surface

致谢

- 衷心感谢导师孙立峰副教授和杨士强教授在毕业设计过程中对本人的精心指导。他们在研究方向和方法上给了我很多宝贵的建议。
- 在整个实验过程中,胡伟栋师兄在我遇到困难的时候一直给予帮助,对我完成实验有巨大帮助,在此对他表示感谢!
- 论文写作过程中用到了LATEX 和很多提供扩展功能的宏包,在此对这些开源软件的作者表示感谢。感谢清华大学高性能所薛瑞尼编写的ThuThesis,让我的论文写作避免了不少繁琐的格式调整,同时让我的论文格式规整了许多。论文中的图由gnuplot绘制,比起很多商业软件更加方便,图形也更美观,在此感谢gnuplot的开发人员。

主要参考文献

- I. Pang Y, Sun L, Wen J, et al. A framework for heuristic scheduling for parallel processing on multicore architecture: a case study with multiview video coding. IEEE Transactions on Circuits and Systems for Video Technology, 2009, 19(11):1658–1666
- II. Smolic A, Müller K, Stefanoski N, et al. Coding algorithms for 3DTV—a survey. IEEE transactions on circuits and systems for video technology, 2007, 17(11):1606–1620
- III. Merkel P, Smolic A, Müller K, et al. Efficient prediction structures for multiview video coding. IEEE Transactions on circuits and systems for video technology, 2007, 17(11):1461–1473

Q&A

谢谢!

卿培

FIT 1-512

edwardtoday@gmail.com