

Universidade de Brasília 11-jun-2012

Meta 4: Transcodificação MPEG-2/H.264 em software

Coordenador: Prof. Ricardo Lopes de Queiroz, Ph.D.

Apresentação por:
Prof. João Luiz Azevedo de Carvalho, Ph.D.

departamento de engenharia elétrica
faculdade de tecnologia

PGEA
PÓS-GRADUAÇÃO EM ENGENHARIA DE SISTEMAS ELETRÔNICOS E DE AUTOMAÇÃO

GPDS

Universidade de Brasília 50

Equipe



Prof. Ricardo Queiroz, Ph.D.



Prof. João Luiz Carvalho, Ph.D.
(fev'09-jun'12)



Prof. Leonardo Aguayo, Dr.
(mai'11-jun'12)



Prof. Renan Utida Ferreira, M.Sc.
doutorando
(ago'10-jun'12)

Estudantes de graduação:

- Francisco Amorim (nov'08-mar'09)
- Antônio Carlos de Aquino (nov'08-mai'09)
- Mateus de Oliveira Ferreira (abr'09-jun'09)
- Daniel Saad N. Nunes (nov'08-jul'09)
- Luiz Raphael V. Santos (jun'09-ago'09)
- Pedro Rogério Vieira Dias (nov'08-out'09)
- Thacio Garcia Scandaroli (nov'08-jan'10)
- Daniilo G. A. Trindade (nov'08-jun'10)
- Raul A. Santana da Silva (abr'09-jun'10)
- Mayara R. Oliveira Santos (out'09-jun'10)

Universidade de Brasília 50

Motivação

- MPEG-2: comumente utilizado em transmissão de vídeo digital
- Sistema Brasileiro de TV Digital: H.264
- É necessário transcodificar de MPEG-2 para H.264 em tempo real

Universidade de Brasília 50

Objetivo

- Fazer transcodificação MPEG-2 / H.264
 - Em tempo real
 - Em software
 - Com a melhor qualidade possível

Universidade de Brasília 50

Premissa

- Evitar estimação de movimento
 - 40% a 90% do tempo de codificação
- Possibilidades na transcodificação
 - Não usar estimação de movimento
 - Search range igual a zero
 - Permite usar macroblocos menores
 - Somente vetores de movimento nulos
 - Utilizar os vetores de movimento do MPEG-2
 - Transcodificação bruta

Universidade de Brasília 50

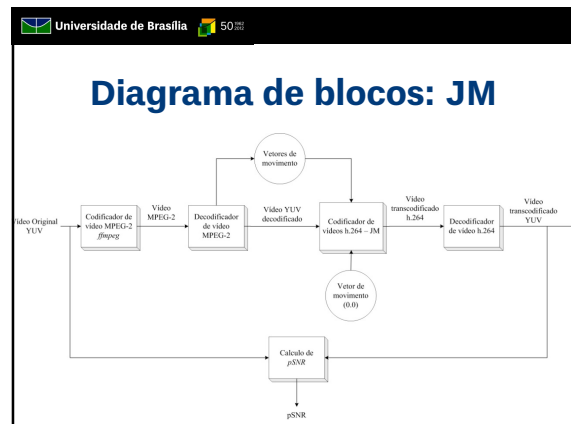
Primeira etapa: testes

- Levantar curvas
 - Qualidade (PSNR) × taxa de bits
 - Velocidade (FPS) × taxa de bits
- Codificadores H.264:
 - JM
 - x264
 - CUDA Encoder

Universidade de Brasília 50

JM (H.264/AVC Reference Software)

- Codificador mais completo para H.264
- Segue a norma H.264/AVC/MPEG-4 parte 10 à risca
- Lento!!!
- Usado apenas para avaliação de qualidade
- Modificado para importar vetores de movimento da sequência MPEG-2
 - Testa e compara com vetores de movimento nulos



Universidade de Brasília 50

x264

- Código aberto (GNU/GPL)
- Implementação otimizada: muito rápido!
- Parâmetros de codificação configuráveis
 - Tamanho da janela de busca, qualidade, etc.
- Vários presets com parâmetros pré-definidos
 - Very fast, pretty fast, medium, high quality, etc.

Universidade de Brasília 50

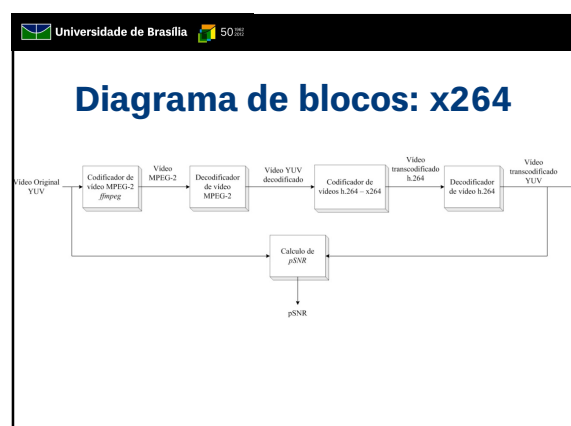
x264: Preset “Ultra-fast”

- Usa CAVLC como codificador de entropia;
- Estimação de movimento em diamante (losango);
- Seleção de referências apenas para partições maiores que 8x8;
- Usa apenas partições 16x16;
- Usa apenas um quadro de referência;
- Desabilita estimação de movimento em quarto-de-pixel;
- Desabilita predição ponderada explícita de quadros P;
- Desabilita controle de taxa baseado em antevisão de quadros;
- Desabilita controle de taxa baseado em árvore de partição de macroblocos;
- Desabilita quantização em treliça;
- Desabilita filtro de *deblocking*;
- Desabilita DCT 8x8;
- Desabilita Quantização Adaptativa;
- Desabilita uso de quadros-B;
- Desabilita decisão adaptativa de corte de cena.

Universidade de Brasília 50

x264: Preset “Super-fast”

- Usa CABAC como codificador de entropia;
- Estimação de movimento em diamante (losango);
- Seleção de referências apenas para partições maiores que 8x8;
- Usa apenas partições 16x16 para quadros-P e partições 16x16, 16x8, 8x16, 8x8, 8x4, 4x8 e 4x4 para quadros-I;
- Usa apenas um quadro de referência;
- Estimação de movimento em quarto-de-pixel em uma iteração e distorção medida em SAD (soma das diferenças absolutas);
- Predição ponderada explícita de quadros-P simples;
- Desabilita controle de taxa baseado em antevisão de quadros;
- Desabilita controle de taxa baseado em árvore de partição de macroblocos;
- Desabilita quantização em treliça.



Universidade de Brasília 50

CUDA Encoder

- Aproveita paralelismo em GPUs Nvidia
- À época: em estado beta
 - Muitas opções não funcionando corretamente
- Integrado ao programa Mediacoder
- Não foi possível isolar o codificador

Universidade de Brasília 50

Investigamos

- Influência do hardware na velocidade de codificação
- Influência do software na velocidade e qualidade de codificação
- Possibilidade de usar os vetores de movimento do MPEG-2 na codificação H.264

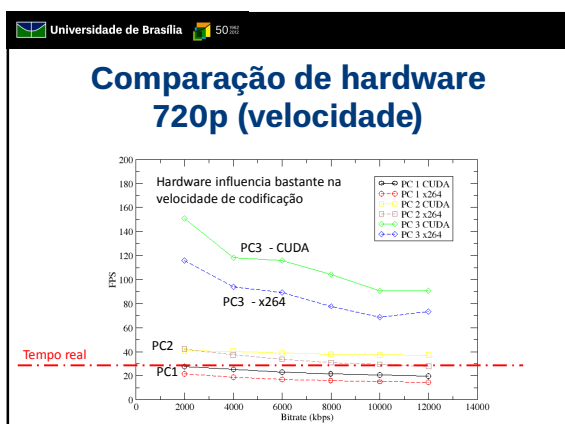
Universidade de Brasília 50

Influência do hardware na velocidade de codificação

Universidade de Brasília 50

Hardware de teste

- Computador 1:**
 - AMD Athlon 64 CPU 4000+, 2.41GHz, 2GB RAM
 - Placa de vídeo Nvidia Geforce GTX 280, PCI Express x16, 1GB RAM, 512 bits, 240 núcleos
- Computador 2:**
 - Intel Core 2 Duo P7350, 2.00GHz, 4GB RAM
 - Placa de vídeo Nvidia Geforce 9600M GT, PCI Express x1, 512MB RAM
- Computador 3:**
 - Intel Core 2 Quad Q9550, 2.83GHz, 4GB RAM
 - Placa de vídeo Nvidia Geforce GTX 275, PCI Express x16 Gen2, 896MB RAM, 240 núcleos



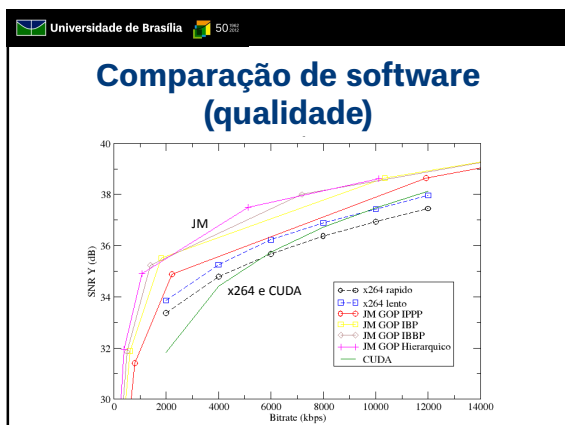
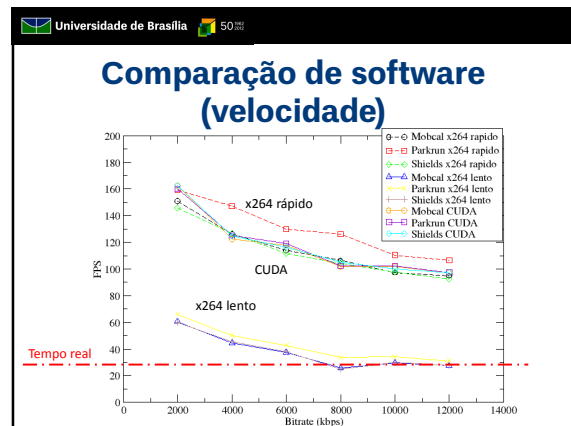
Universidade de Brasília 50

Influência do software na velocidade e qualidade de codificação

Universidade de Brasília

VÍDEOS DE TESTE

Nome do vídeo	Nº de frames	Resolução	Bitrate MPEG-2 (Mbits/s)
Stockholm	250	720p	12
Stockholm	250	720p	3
Mobcal	250	720p	12
Mobcal	250	720p	3
Pedestrian area	250	1080p	16
Pedestrian área	250	1080p	5
Blue sky	217	1080p	16
Blue sky	217	1080p	5

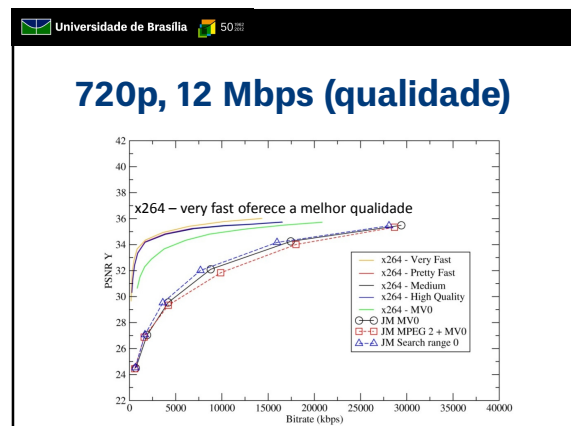
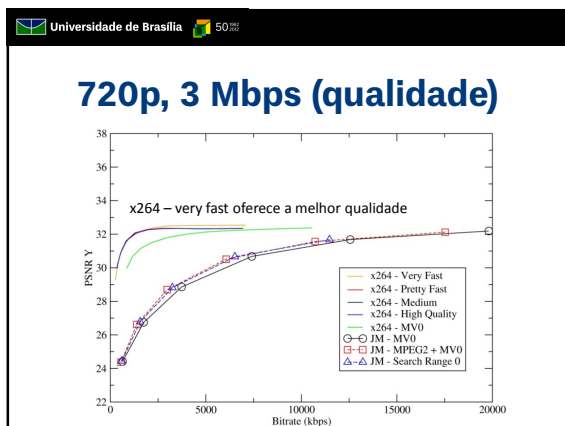


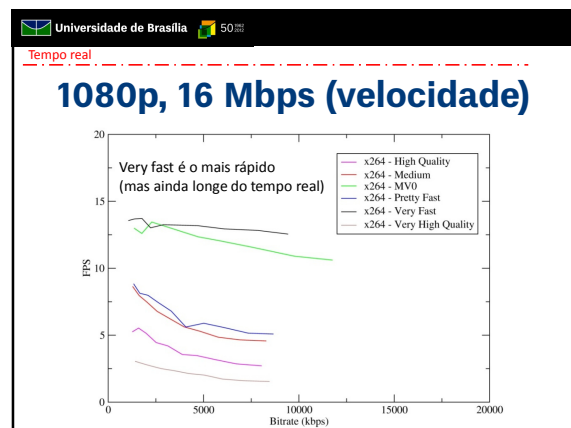
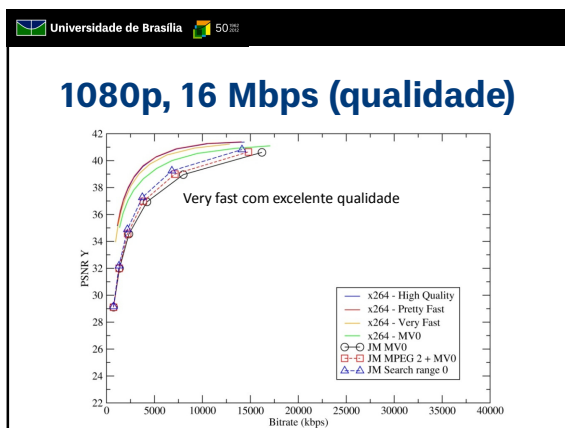
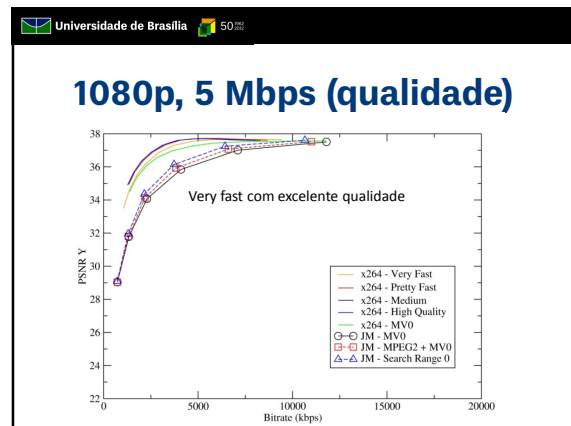
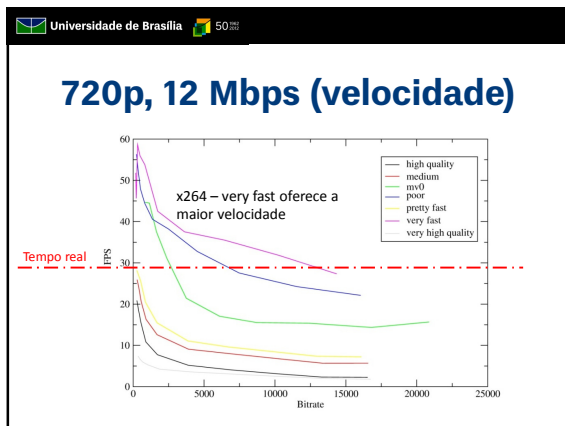
Universidade de Brasília

Vale a pena importar vetores de movimento do MPEG-2?

Resultados para:

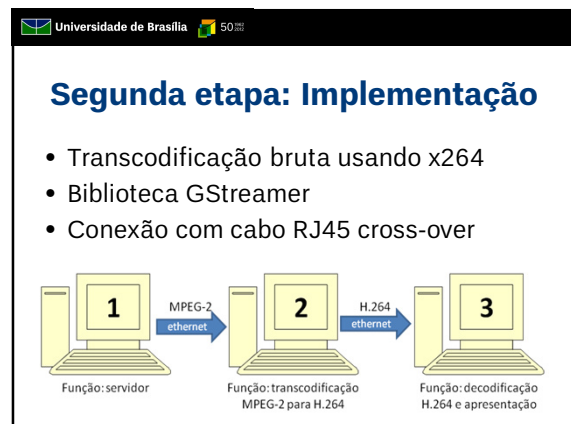
- 720p (Stockholm)
 - MPEG-2 de 3 Mbps
 - MPEG-2 de 12 Mbps
- 1080p (Pedestrian Area)
 - MPEG-2 de 5 Mbps
 - MPEG-2 de 16 Mbps





Conclusões da etapa de investigação

- Não é vantajoso utilizar os vetores de movimento do MPEG-2
- x264 em *hardware* adequado transcodifica 720p em tempo real
- Transcodificação bruta se mostrou robusta para vídeos em 720p
 - Altas taxas de FPS na transcodificação
 - PSNR maior que transcodificação que importa vetor de movimento do MPEG-2



Universidade de Brasília 50

Biblioteca GStreamer

- Código aberto
- Ferramentas e codecs pré-compilados: plugins
- Plugin "x264enc"
 - Implementa o codificador x264
 - Velocidade e eficiência
- Plugins "mpegtsmux" & "mpegtsdemux"
 - Implementam a camada de transmissão TS (*transport stream*)

Universidade de Brasília 50

Cascadeamento de plugins: "pipeline"

Gstreamer pipeline for a basic ogg player

Universidade de Brasília 50

Implementação no GStreamer

Função: servidor Função: transcodificação MPEG-2 para H.264 Função: decodificação H.264 e apresentação

Universidade de Brasília 50

Pipeline 1: RAW p/ MPEG-2

- Plugin "videorate"
 - Ajusta a taxa de quadros de 60 fps para 30 fps
- Plugin "timeoverlay"
 - Superimpõe o tempo de reprodução no video

Universidade de Brasília 50

Pipeline 1: RAW p/ MPEG-2

```
gst-launch -v \
filesrc location=vidyo3_720p_60.yuv \
! videoparse width=1280 height=720 \
  framerate=60/1 \
! videorate \
! video/x-raw-yuv,framerate=60/2 \
! timeoverlay ypad=0 xpad=46 \
! ffenc_mpeg2video bitrate=3000000 \
! avimux \
! filesink \
  location=vidyo3_720p_30_mpeg2.avi
```

Universidade de Brasília 50

Pipeline 2: MPEG-2 TS via TCP

- Envia stream MPEG-2 TS por TCP p/ o transcoder

Universidade de Brasília 50

Pipeline 2: MPEG-2 TS via TCP

```

while true; \
do \
gst-launch \
filesrc location=vidyo3_720p_30_mpeg2.avi \
! avidemux \
! queue \
! tee name=t \
! queue \
! decodebin \
! xvimagesink t. \
! queue \
! mpegtsmux \
! tcpserver sink sync-method=1 host=10.10.11.5 port=4000; done
    
```

Universidade de Brasília 50

Pipeline 3: RX + Transcod. + TX

- Recebe MPEG-2 TS, transcodifica e envia H.264 TS via TCP

Universidade de Brasília 50

Pipeline 3: RX + Transcod. + TX

```

gst-launch \
tcpclientsrc host=10.10.11.5 \
port=4000 \
! mpegtsdemux \
! decodebin \
! x264enc speed-preset=2 pass=4 \
quantizer=10 \
! mpegtsmux \
! tcpserver sink sync-method=1 \
protocol=none port=4000
    
```

Universidade de Brasília 50

Pipeline 4: Recebe H.264 e decodifica

- Recebe H.264 TS, decodifica e mostra na tela

Universidade de Brasília 50

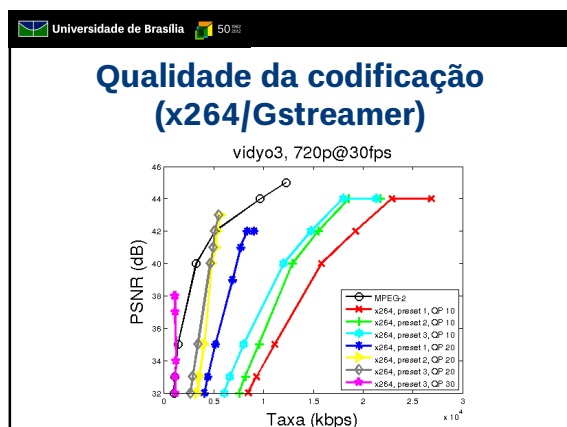
Pipeline 4: Recebe H.264 e decodifica

```

while true;
do \
gst-launch \
tcpclientsrc host=10.10.13.4 port=4000 \
! mpegtsdemux \
! decodebin \
! clockoverlay xpad=46 ypad=60 auto- \
resize=false \
! timeoverlay xpad=46 ypad=120 auto- \
resize=false \
! xvimagesink sync=true \;
done
    
```

Universidade de Brasília 50

Vídeo utilizado: vidyo3



Universidade de Brasília 50

Conclusões finais

- Formação de recursos humanos
 - 4 docentes (um doutorando)
 - 10 estudantes de graduação
- Testes de qualidade e velocidade
 - Diferentes hardwares (incluindo CPU e GPU)
 - Diferentes softwares (JM, x264 e CUDA)
 - Diferentes abordagens (bruta, MV=0, SR=0)
- Solução proposta:
 - transcodificação MPEG-2/H.264 bruta c/ x264
 - Implementada usando a biblioteca Gstreamer
 - Fácil desenvolvimento (pipelines) & software livre
 - Hardware comercial (quad core)
 - Vídeos 720p em tempo real