INSTITUTO DE INFORMÁTICA UFRGS

VVCPy: Um Framework para Análise e Visualização de dados para o padrão Versatile Video Coding

Luís Eduardo Pereira Mendes
Aluno de Engenharia de Computação - UFRGS
Cláudio Machado Diniz
Orientador

UFRGS
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Crescimento da demanda por processamento de vídeo

- Aumento do número de dispositivos de vídeo
- Aumento da resolução dos vídeos produzidos

**Cisco Annual Internet Report (2018–2023) White Paper*

Codificação de Vídeo

- Uso de técnicas de redução de redundância para diminuir a quantidade de dados enviada para representar informação de vídeo;
- Possibilita o armazenamento e transmissão de vídeo;

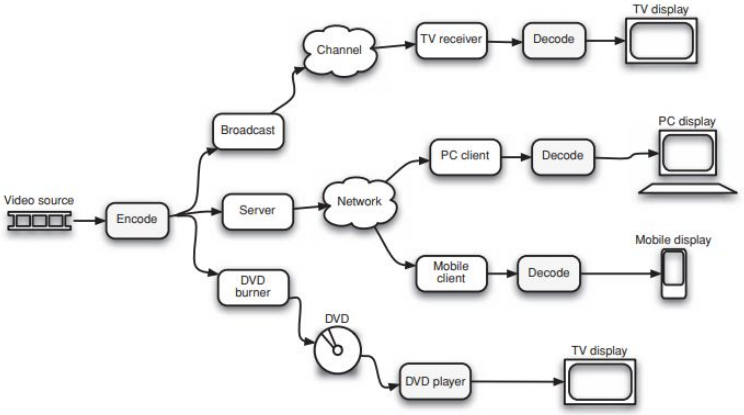


Figure 1.1 Video coding scenarios, one-way

Figure 3.4 Frame 1

Figure 3.5 Frame 2

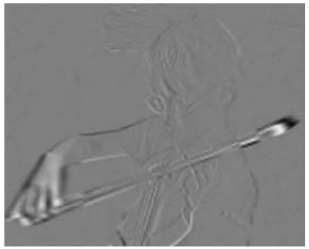


Figure 3.6 Difference

Versatile Video Coding (VVC/H.266)

- Padrão de codificação de vídeo;
- Sucessor do padrão High Efficiency Video Coding (HEVC/H.265);
- Realiza compressões de até 50% mais do que o HEVC;
- Possui um custo computacional de até 30x o seu antecessor

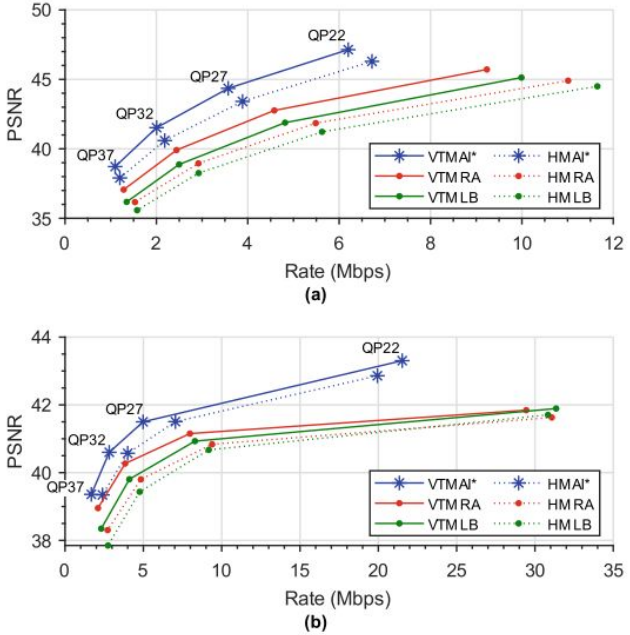


FIGURE 2. PSNR RD curves. (a) *RitualDance* (1920 × 1080). (b) *Tango* (4096 × 2160). (*) The actual AI coding rate is eight times as high due to sequence subsampling.

Aumento da complexidade do codificador

- Predição Intra e Inter quadros, Estimação de movimento e outras técnicas são muito custosas computacionalmente
- Exige o desenvolvimento de Hardware especializado

Cenário para possibilidade de implementação do VVC

- Aceleração de hardware usando módulos especializados em computar funções de codificação
- Aplicação de técnicas de computação aproximada

Demanda por agilidade no processo de desenvolvimento

- Para que seja possível realizar mudanças substanciais que tragam melhorias para o VVC, é necessário o uso de ferramentas de automação de processos

Demanda por uma grande quantidades de dados

- A aplicação de alterações voltadas para o software do codificador necessitam de testagem e validação

Framework para automatizar o processamento de vídeos para o VVC

- Surge a necessidade de um software que seja capaz de automatizar:
  - O processo de obtenção de dados do VVC para um determinado conjunto de vídeos
  - O processo de manipulação e obtenção de métricas a partir dos dados gerados
  - A geração de gráficos, tabelas e informações visuais sobre o conjunto de vídeos
  - A comparação de diferentes execuções do codificador
- Ferramenta código-aberto (open-source) livre para modificação e incrementação pela comunidade

Estudo da Arquitetura do Sistema

- A partir da análise de trabalhos recentemente publicados se estabeleceu os requisitos para o desenvolvimento do framework
- Escolha das ferramentas utilizadas para o desenvolvimento da aplicação baseada no ambiente científico Python
  - Pandas, Numpy, Matplotlib, Scipy

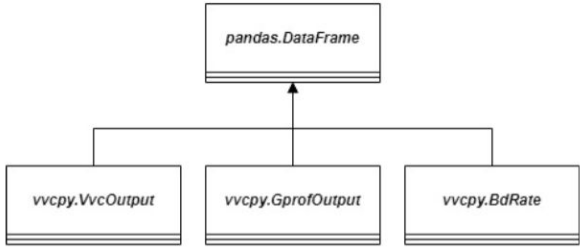


Fig. 1: Data Structures of VVCPy

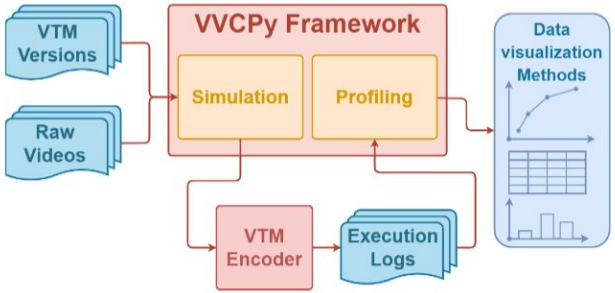


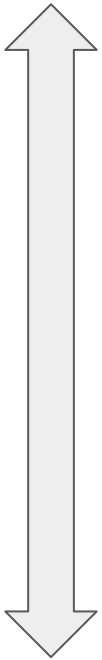
Fig. 2: VVCPy Framework System Diagram

Implementação e validação

- Implementação realizada em Python usando herança e polimorfismo de classes do framework Pandas
- Verificação realizada por meio de testes unitários e testes de sistema
- Validação realizada pela utilização pelo grupo de pesquisa

Uso do VVCPy

- O framework pode ser usado em diferentes níveis de abstração
  - provê uma interface de funções de alto nível, que realiza as operações mais comuns de obtenção de dados e métricas gerais
  - provê uma interface de baixo nível, com métodos mais simples que podem ser usados para construir extensões para usos específico
- Foco na realização de simulações automatizadas de execuções do codificador VVC



```
vvcpy.Simulation()
vvcpy.Output.plot()
vvcpy.calc_bdr()
```

```
vvcpy.run_vtm()
vvcpy.find_file()
```

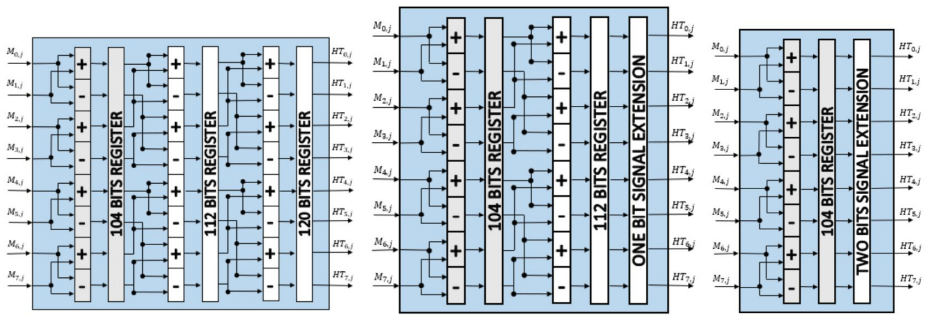
Explorando caso de estudo: Descrição do experimento

- O experimento conta com a redução de camadas de somador no hardware dedicado à transformação de Hadamard para unidades de codificação 8x8
- A remoção é simulada no software do codificador, testando funcionalmente o resultado

$$SATD = \sum_{i,j} |HT_{(i,j)}| \tag{1}$$

$$HT = H \cdot W \cdot H^T \tag{2}$$

$$H = \frac{1}{4} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{bmatrix} \tag{3}$$



(a) Precise (b) with One Column of adders/subtractors excluded (c) with Two Columns of adders/subtractor excluded.

Fig. 3: Block Diagram of the Vertical Transform Block.

Explorando caso de estudo: avaliação de um modelo de computação aproximada

- Para avaliar a utilização do framework, o cálculo da transformação de Hadamard foi aproximado no software do VVC, com a redução de até 80% as operações de soma realizadas durante esse método.
- Usando o script ao lado, foi possível testar um conjunto de 10 vídeos e gerar um gráfico a partir do conjunto de métricas obtido.

```
import vvcpy as vp

sim = vp.Simulation(
    n_frames =  32,
    encoder  =  ["RA", "AI", "LB"],
    qps      =  [22,27,32,37]
)
sim.set_environment("setup.yaml")
sim.change_version(
    new_file = "RdCost-8x8-SAD.cpp"
    old_file = "CommonLib/RdCost.cpp"
    version = "RdCost-8x8-SAD"
)

sim.run_exec()
```

Fig. 4: Running the experiment using a new version of VTM

```
vp.set_environment("setup.yaml")
bdr = prof.calc_bdr(
    output = vp.env.output,
    cfgs   =  ["AI", "LB", "RA"],
    qps    =  [22, 27, 32, 37]
)

bdr.plot_bdr_video("RaceHorses")
```

Fig. 5: Script to calculate BD-Rate.

Explorando caso de estudo: Resultados obtidos da execução dos scripts

- Ao fim da execução do script, o seguinte gráfico foi gerado, possibilitando uma análise sobre o impacto da aproximação na codificação do vídeo.
- O gráfico representa a comparação da métrica *Bjontegaard-Delta Rate (BD-Rate)*, quanto menor, menos o vídeo aproximado foi distorcido.

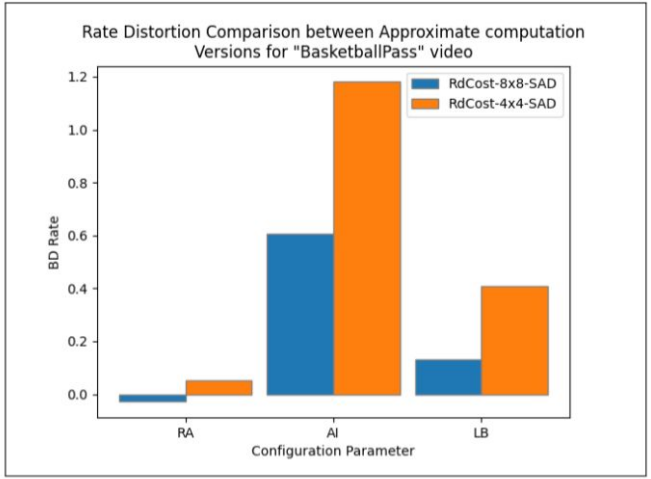


Fig. 7: BD-Rate comparison between different versions.

Version	Video	Config.	BD-rate
RdCostModif	BasketballPass	AI	0.028333
		RA	-0.131668
	RaceHorses	AI	0.027436
		RA	0.274682

TABLE VIII: BD-Rate comparison between the modified version and the precise version of VTM

Quando o Framework é útil?

- Após a avaliação feita pelo grupo de pesquisa, foi elencado os principais usos para o framework

Métodos de computação aproximada

- Avaliar quais mudanças ocasionam uma perda de qualidade significativa
- Avaliar diferentes versões de aproximações

Avaliação de complexidade de software

- Encontrar dentro do software as funções mais custosas
- Analisar o impacto de diferentes funções em diferentes tipos de vídeo

Obtenção de dados para treinamento de Inteligência Artificial

- Obter dados estruturados que podem ser utilizados como entrada para redes neurais

Próximos passos

- A utilização de um framework automatizado para realizar as tarefas de execução de simulações do codificador se mostrou eficaz para acelerar e facilitar o desenvolvimento de experimentos
- Entretanto existem aspectos a serem melhor trabalhados e características a serem adicionadas

Adição de suporte à outras métricas menos utilizadas, porém relevantes

Adição de suporte à outros codificadores, como o HEVC e o AVC

Adição de outros métodos de visualização de dados

Obrigado!

Instituto de Informática
Universidade Federal do Rio Grande do Sul - UFRGS

INSTITUTO DE INFORMÁTICA
UFRGS