Codificação Áudio e Vídeo

Trabalhos Práticos de
Codificação de Vídeo

Filipe Manco | 45500
<filipe.manco@ua.pt>

Frederico Honório | 42914
<fredericohonorio@ua.pt>

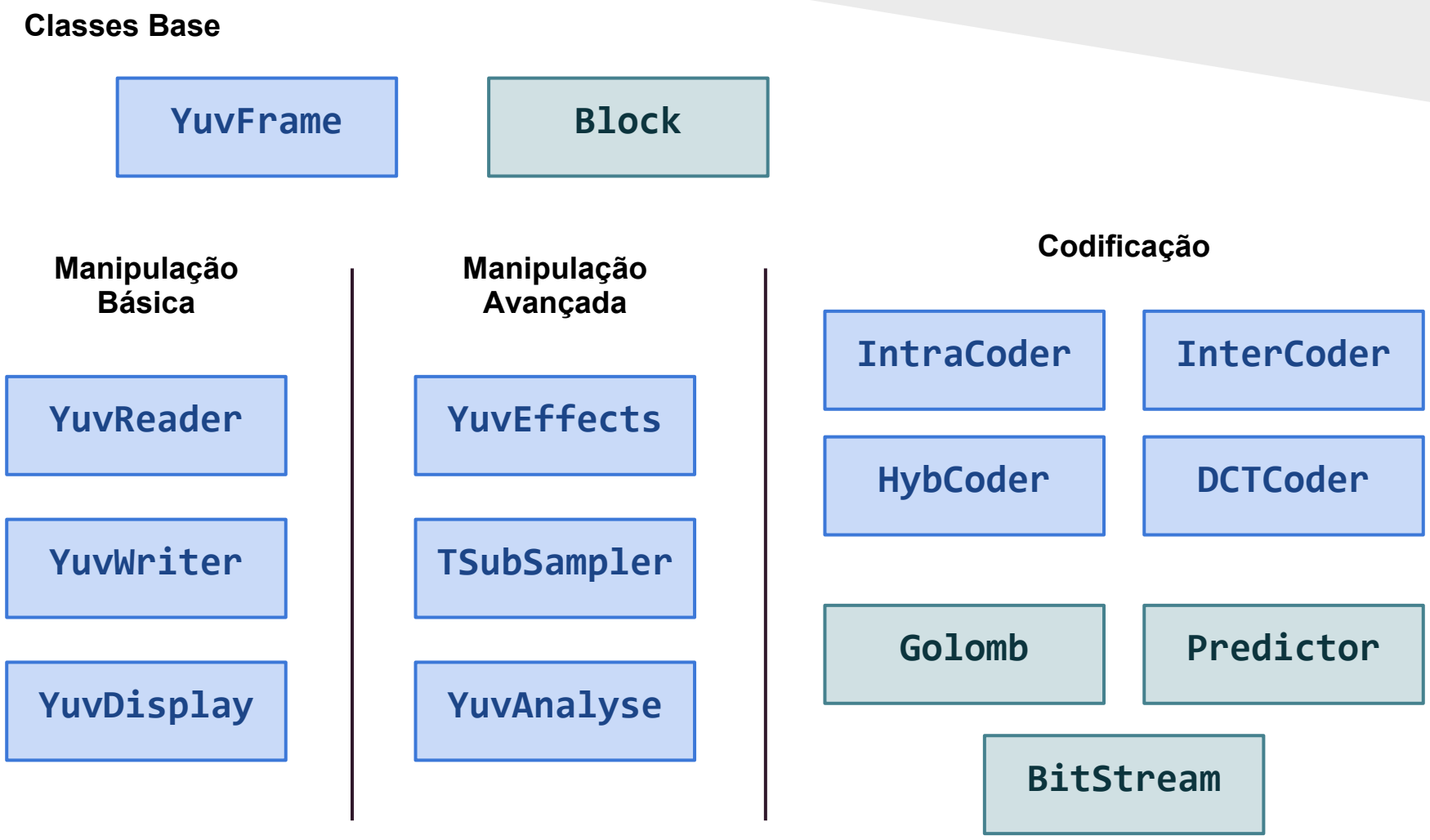
Arquitectura do Software

Primeira implementação

- A arquitectura não foi bem definida
  - Apesar dos algoritmos estarem bem implementados

Abordagem:

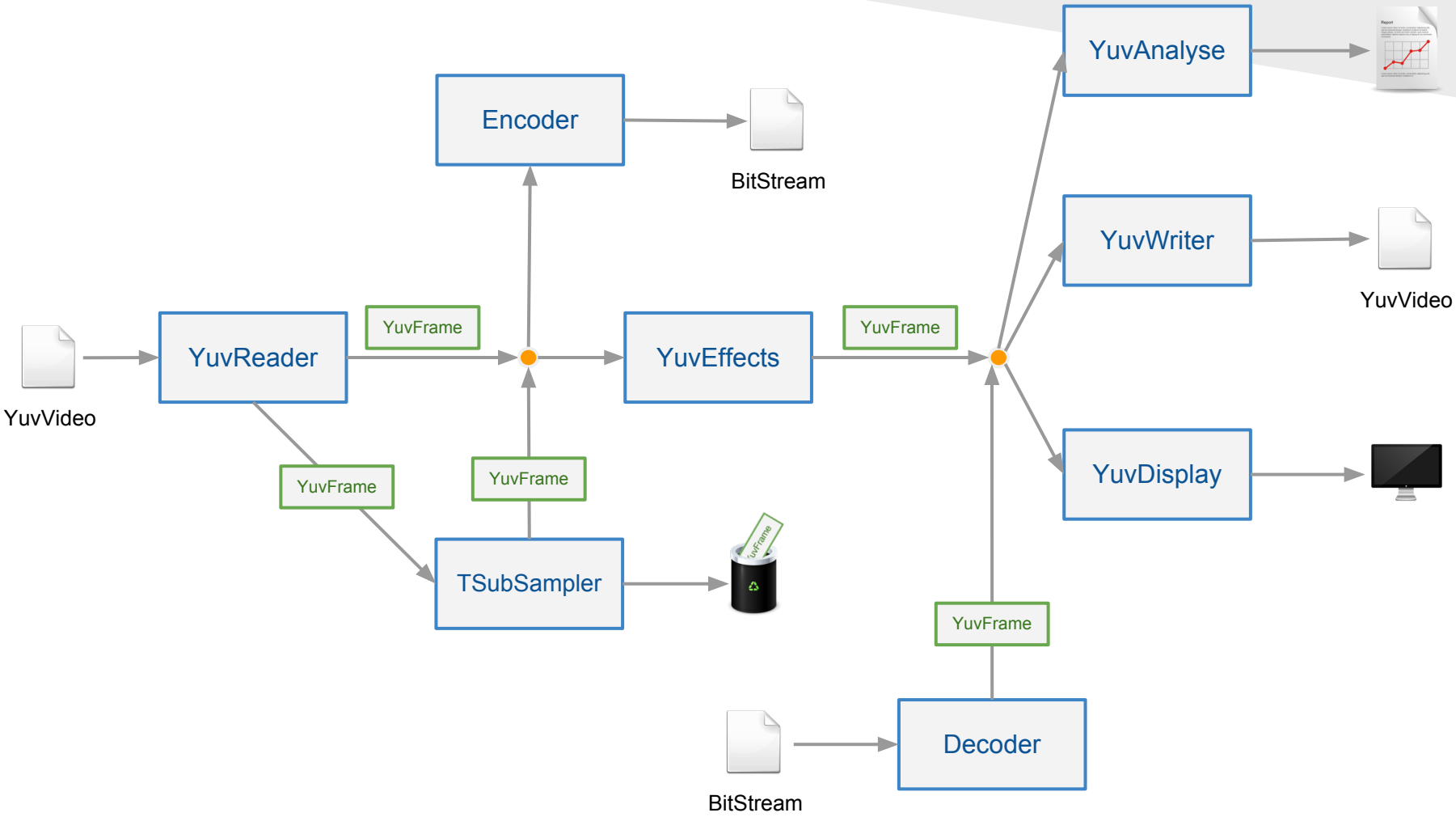
Nova Arquitectura Base



YuvFrame

- Armazena os dados de **uma** frame
- Permite converter entre formatos
  - YUV444
  - YUV422
  - YUV420
- Escrita e leitura em qualquer dos formatos é transparente ao utilizador
- Operação ao nível:
  - do buffer
  - do bloco
  - do píxel

Workflow


YuvAnalyse
Encoder
BitStream
YuvWriter
YuvVideo
YuvVideo
YuvReader
YuvFrame
YuvEffects
YuvFrame
YuvDisplay
YuvFrame
YuvFrame
TSubSampler
YuvFrame
Decoder
BitStream

Implementação

- Algumas classes não foram portadas:
  - YuvEffects
  - TSubSampler
- Apesar das interfaces estarem definidas não eram necessárias para esta iteração

Codificação: Algoritmos auxiliares

Codificação de entropia

- Golomb
  - Utilizando códigos de Rice ($m=2^k$)
  - Melhor[1] k, para a média μ, está entre k_{min}, k_{max}:

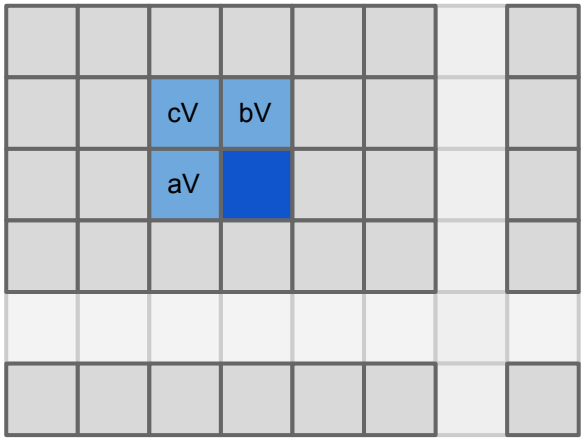
$$k_{min}(\mu) = max\{0, \lfloor log_2(2/3(\mu+1)) \rfloor\}$$

$$k_{max}(\mu) = max\{0, \lceil log_2 \mu \rceil\}$$

[1]: Kiely, A. (2004). Selecting the Golomb Parameter in Rice Coding, 2, 2-3.

Preditores

- 7 Preditores lineares do JPEG
- 1 Preditor não linear do JPEG-LS



JPEG-LS - Non Linear Predictor

```
int v    = 0;
int max_ = max(aV, bV);
int min_ = min(aV, bV);

if (cV >= max_) {
     v = min_;
} else if (cV <= min_) {
     v = max_;
} else {
     v = aV + bV - cV;
}
```

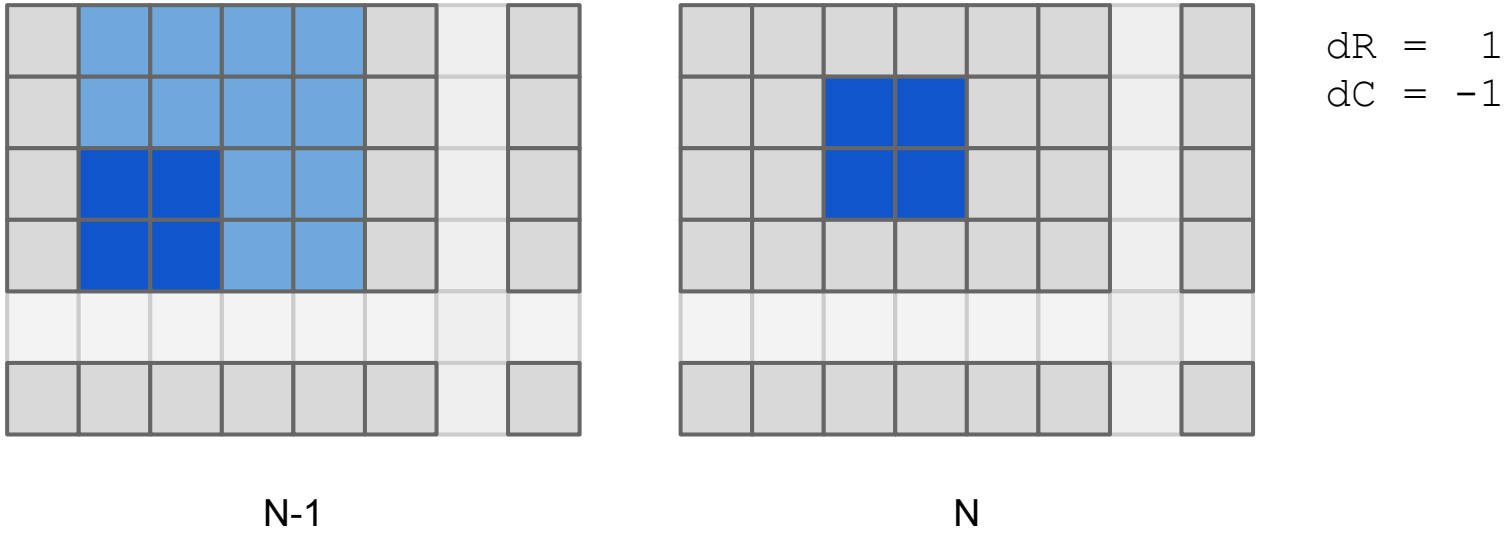
Codificadores

Codificação intra-frame com preditores

- Para cada frame
  - Para cada pixel
    - O preditor estima um valor do pixel
    - O erro entre a estimativa e o valor atual é codificado para o ficheiro resultante
      - Este erro pode ser quantizado.

Codificação inter-frame

- Compensação de movimento



- Os buffers (Y, U, V) são tratados independentemente

Codificação intra-frame com DCT

- Transforma uma função numa soma de cosenos a diferentes frequências
- São guardados os coeficientes produzidos
- Utilidade?
  - Coeficientes são valores reais, mas mais pequenos.
    - Codificação lossless é ineficiente (espaço).
    - Arredondamento introduz perdas.
  - Os primeiros coeficientes são mais importantes
  - Podem ser quantizados ou até omitidos
    - Depois de quantizados, pelo menos metade são zero.
    - Standard JPEG define matrizes de quantização para luminância (Y) e crominância (U/Cb, V/Cr)

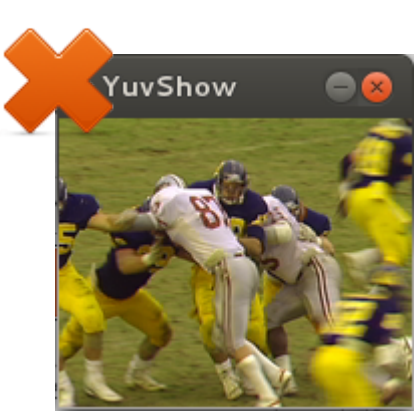
Resultados

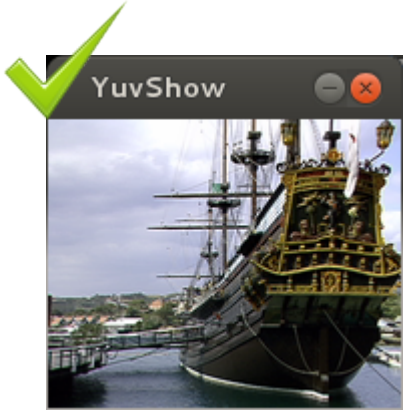
Efeito dos preditores

- O preditor não-linear revelou-se o melhor
- São bons para frames com blocos de cor
- O melhor preditor depende das características da frame

Codificador híbrido

- Compensação de movimento
  - Eficaz para situações de continuidade de cena
  - Má para mudanças de cena
  - Má para cenas com muito movimento

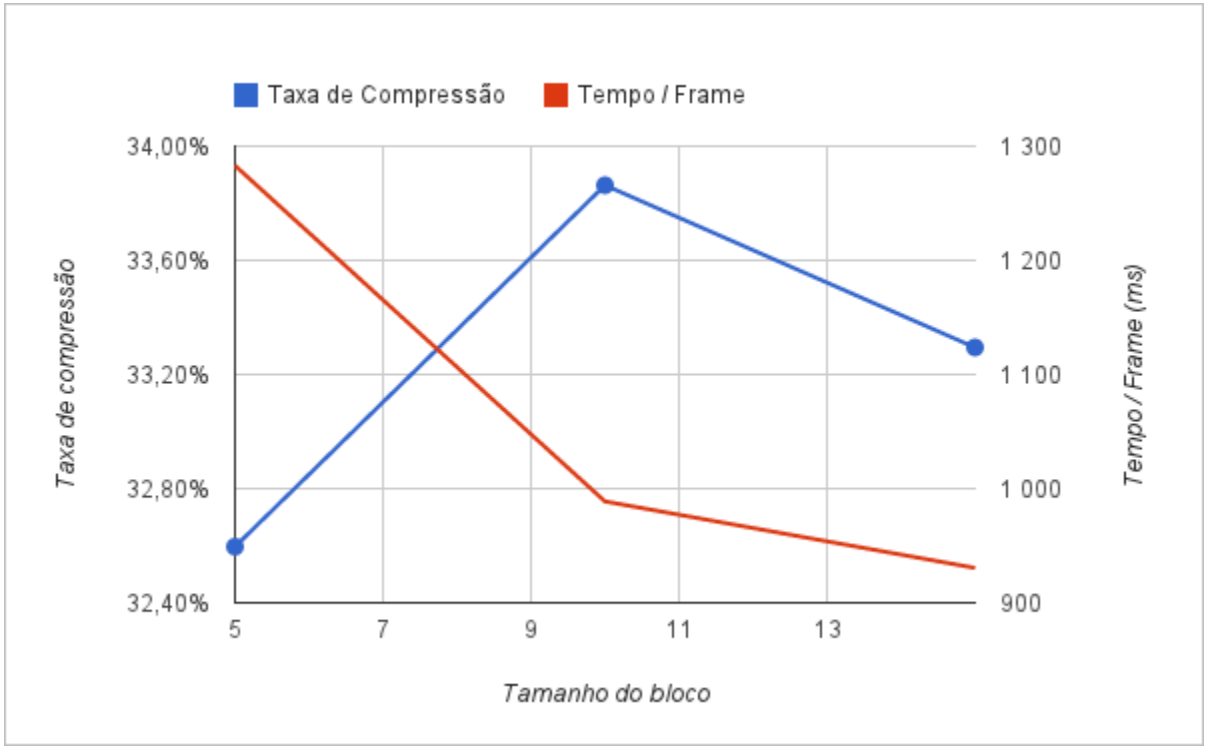




- Melhor altura para introduzir uma keyFrame
  - Seleccionada automaticamente com base no erro

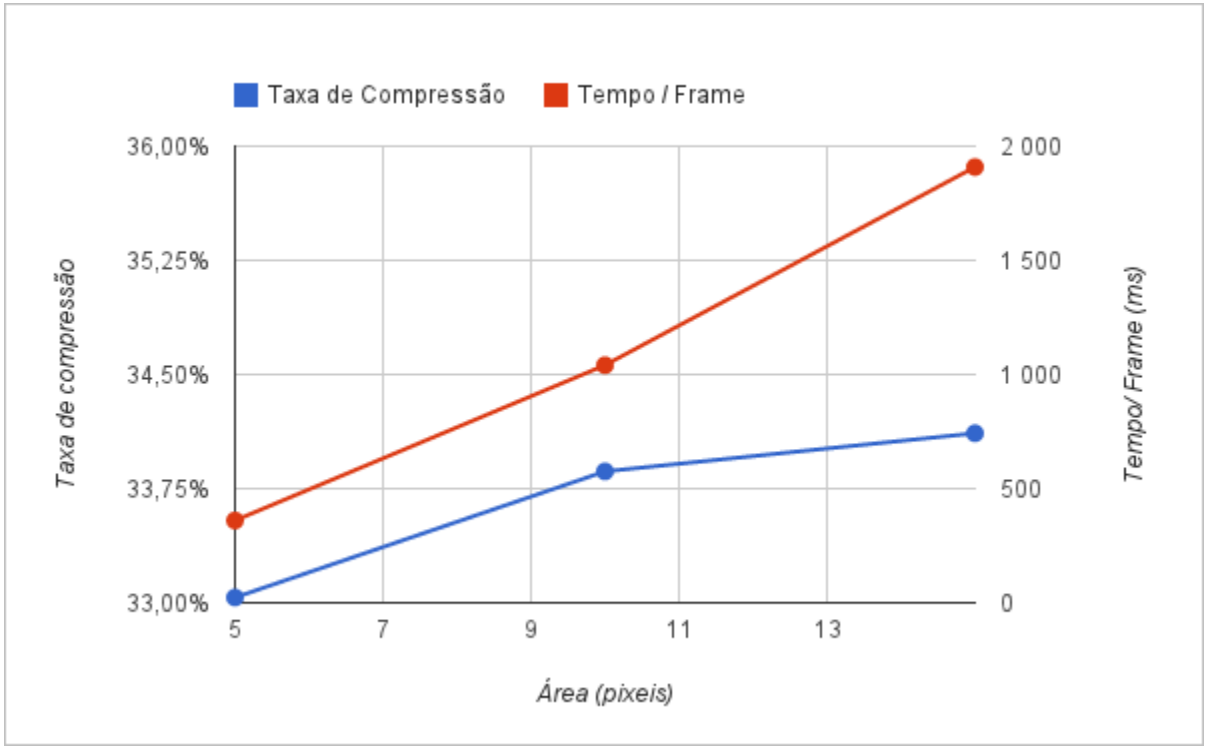
Compensação de movimento (1)

- Efeito do tamanho do bloco



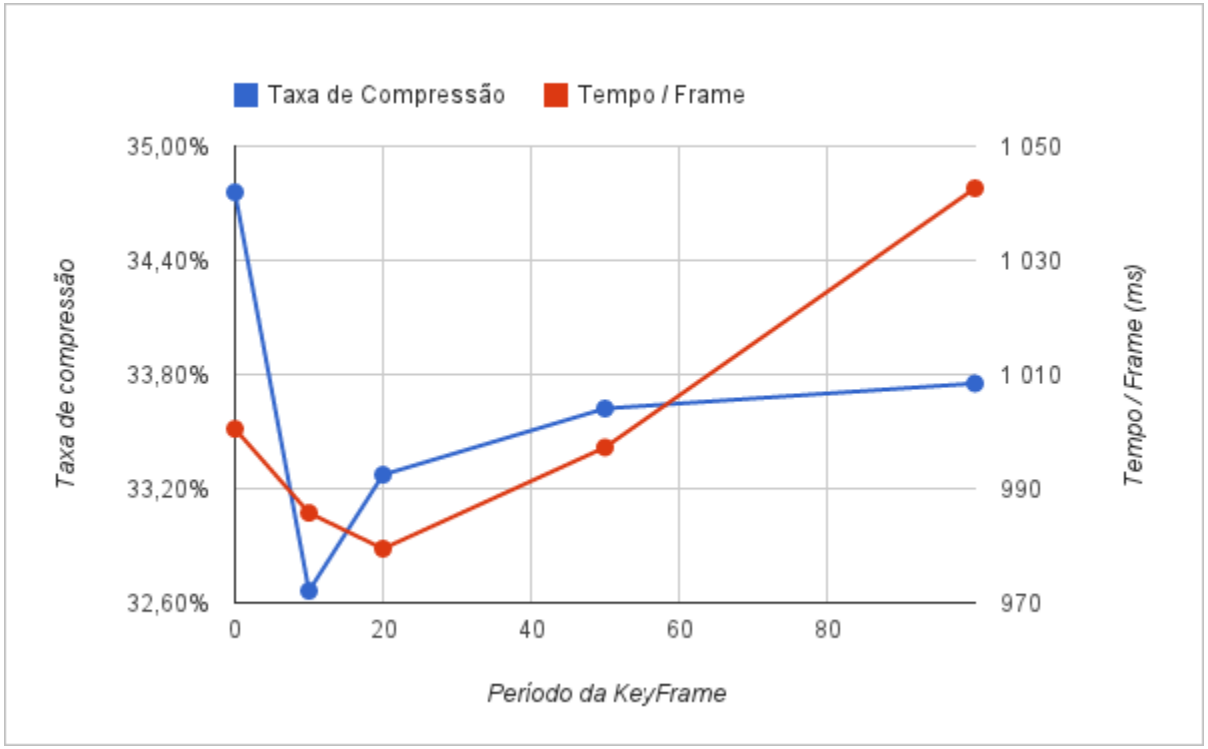
Compensação de movimento (2)

- Efeito da área de pesquisa



Compensação de movimento (3)

- Efeito do período da keyFrame



DCT

- Devia produzir melhor taxa de compressão
- No entanto:

	PNSR	Compressão
Intra quantized(4, 8, 8)	39,55	45%
DCT quantized(1.0)	34,82	48%
Intra quantized(8,16,16)	33,56	48%
DCT quantized(2.5)	32,40	50%

- A nossa implementação não usa RLE
  - Cada zero é codificado com 4 bits.

Conclusões

- Golomb
  - Bom para codificar valores com pouco variância
  - É importante escolher um bom M
  - Codificador aritmético adaptativo daria melhores ganhos
    - Implementação é mais complexa
- DCT
  - Melhor taxa de compressão (quando bem implementado)
  - Lossy
  - Introduz alguns artefactos