

# **Processamento Paralelo de Vídeo: Um Estudo de Caso com Conversão para Preto e Branco**

**Lucas R. Silva, Luan Medrado Moreira**

Departamento de tecnologia – Centro Universitário Euro-americano (Unieuro)  
Av. das Castanheiras, s/n - Lote 3700 - Águas Claras, Brasília - DF, 70297-400

lucas60899@unieuro.com.br, luan61449@unieuro.com.br

## **Abstract**

This paper presents a study on parallel video processing, focusing on the conversion of videos to black and white. The main objective was to investigate techniques to optimize performance and achieve a speedup of 2.0 in video transformation, using multiple threads. A high-resolution video and a computer with a Ryzen 7 5700G processor and 16GB of RAM were used. The study explored different approaches, from processing the entire video to splitting it into smaller parts and using threads to parallelize the conversion. The results showed that splitting the video into parts and using 8 threads provided the best performance, achieving a significant speedup over sequential processing.

## **Resumo**

Este trabalho apresenta um estudo sobre o processamento paralelo de vídeo, com foco na conversão de vídeos para preto e branco. O objetivo principal foi investigar técnicas para otimizar o desempenho e alcançar um speedup de 2.0 na transformação de vídeos, utilizando múltiplas threads. Para isso, foi utilizado um filme e um computador com processador Ryzen 7 5700G e 16GB de RAM. O estudo explorou diferentes abordagens, desde o processamento do vídeo completo até a divisão em partes menores e o uso de threads para paralelizar a conversão. Os resultados mostraram que a divisão do vídeo em partes e o uso de 8 threads proporcionaram o melhor desempenho, alcançando um speedup significativo em relação ao processamento sequencial.

## **1. Introdução**

O processamento de vídeo é uma área desafiadora da computação, com aplicações em diversas áreas, como edição de vídeo, streaming ao vivo e visão computacional. A crescente demanda por vídeos de alta resolução e a necessidade de processamento rápido exigem soluções eficientes e otimizadas. A paralelização, através do uso de múltiplas threads, é uma técnica promissora para acelerar o processamento de vídeo, aproveitando o poder de processamento de múltiplos núcleos.

Neste trabalho, exploramos o processamento paralelo de vídeo, com foco na conversão de vídeos para preto e branco. A escolha dessa tarefa se deve à sua simplicidade e representatividade, permitindo uma análise clara do impacto da paralelização no desempenho. O objetivo principal foi investigar técnicas para otimizar o desempenho e alcançar um speedup de 2.0 na transformação de vídeos.

## 2. Metodologia

Para realizar o estudo, utilizamos um vídeo (1 hora e 30 minutos) do filme Resident Evil 2. O computador utilizado possuía um processador Ryzen 7 5700G, 16GB de RAM e um SSD. A linguagem de programação Python foi escolhida para implementar as diferentes abordagens de processamento, devido à sua facilidade de uso e bibliotecas eficientes para manipulação de vídeo, como OpenCV e MoviePy.

### 2.1 Métodos com resultados não satisfatório

Inicialmente, exploramos o processamento do vídeo completo, convertendo cada quadro para preto e branco sequencialmente. No entanto, essa abordagem se mostrou ineficiente devido ao alto consumo de memória. Em seguida, investigamos um método que processava cada quadro individualmente e armazenava os resultados em uma fila, utilizando threads para paralelizar o processo. Essa abordagem apresentou resultados promissores, mas não atingiu o speedup desejado ficando com 1.6 de speedup.

### 2.2 Método eficiente

Finalmente, chegamos à abordagem final, que consistia em dividir o vídeo em 8 partes menores antes do processamento. Cada parte era então processada em paralelo por uma thread, Essa abordagem permitiu um melhor aproveitamento dos recursos do computador e resultou em um desempenho significativamente superior em relação às abordagens anteriores.

## 3. Resultados

A Tabela 1 apresenta os resultados obtidos com as diferentes abordagens de processamento, incluindo o tempo de execução para o processamento sequencial (sem threads / 1 thread) e para diferentes números de threads.

**Tabela 1**

Número de Threads	Tempo de Execução (min:seg)	Speedup
1	10:29	1.0
2	6:05	1.7
4	4:12	2.5
6	4:25	2.4
8	3:43	2.9

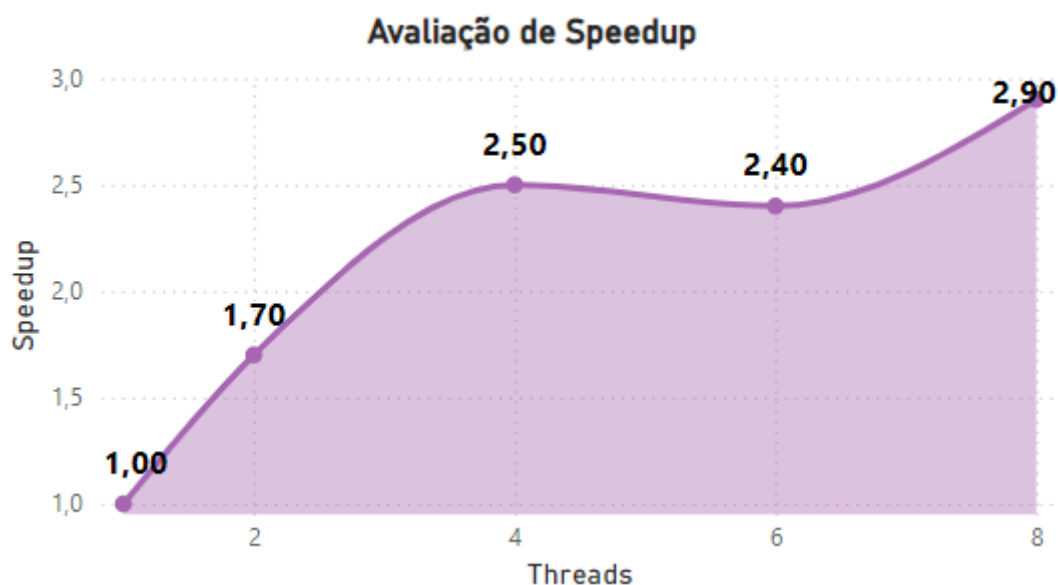
Observamos que a divisão do vídeo em partes e o uso de múltiplas threads resultaram em um speedup significativo em relação ao processamento sequencial. O melhor desempenho foi alcançado com 8 threads, atingindo um speedup de 2.9, ultrapassando o objetivo de 2.0.

### 3.1 Resultados de forma gráfica

#### 3.1.1 Análise de Desempenho do Processamento Paralelo com Threads

O Gráfico 1 ilustra a evolução do speedup em relação ao aumento do número de threads. Observa-se um aumento significativo no speedup ao utilizar 2 threads, atingindo um fator de 1.7x em comparação ao processamento sequencial. Com 4 threads, o speedup chega a 2.5x, demonstrando um ganho considerável em desempenho. No entanto, ao adicionar mais threads, o aumento no speedup se torna menos expressivo, atingindo 2.9x com 8 threads. Esse comportamento sugere que o sistema pode estar se aproximando de um limite na capacidade de paralelização da tarefa, onde a sobrecarga na criação e gerenciamento das threads, juntamente com a contenção por recursos compartilhados, começam a impactar o desempenho.

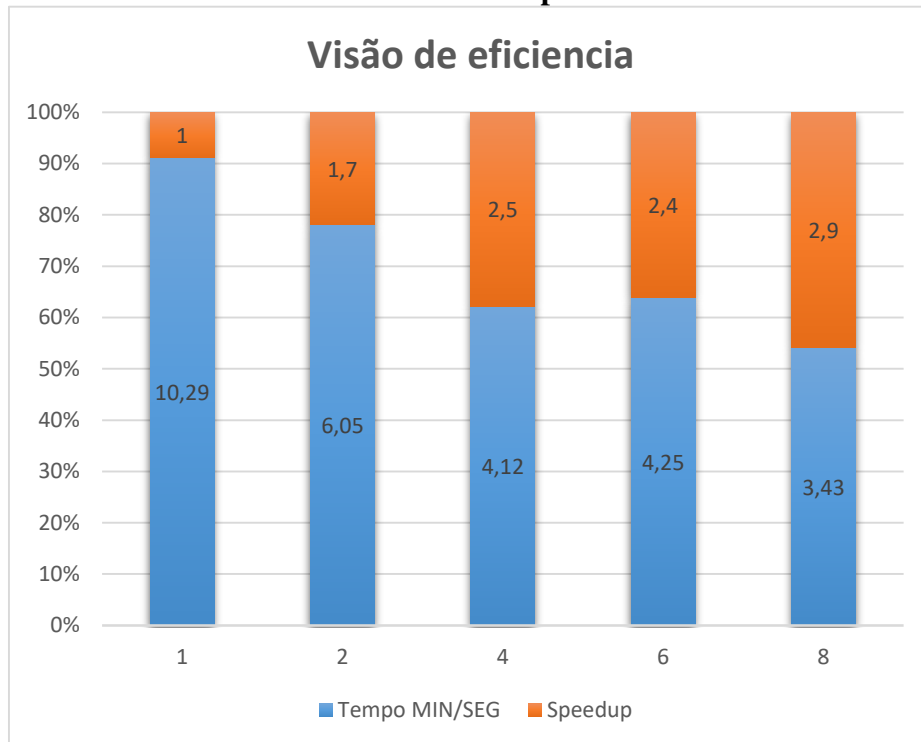
**Gráfico 1 Speedup por número de Threads**



#### 3.1.2 Relação entre Eficiência e Tempo de Execução

O gráfico 2 de colunas 100% empilhadas evidencia a relação inversa entre o tempo de execução e a eficiência do processamento. À medida que o tempo de execução diminui, a eficiência aumenta, indicando que o sistema está utilizando os recursos de forma mais otimizada. No entanto, é importante notar que a eficiência não atinge 100%, mesmo com tempos de execução menores, o que sugere que existem fatores limitantes, como a sobrecarga na comunicação entre as threads e a natureza da tarefa em si, que podem impedir a utilização total dos recursos disponíveis.

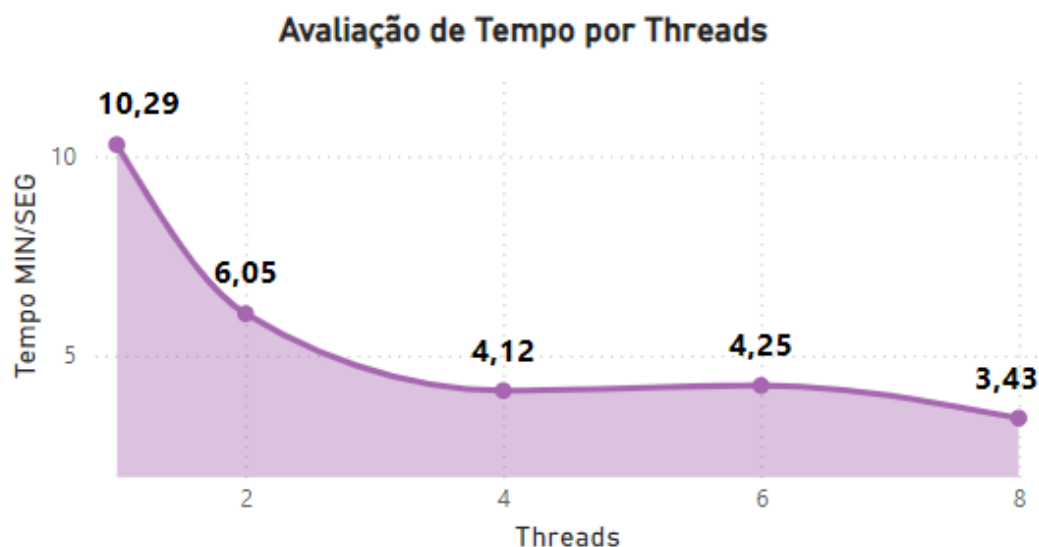
**Gráfico 2 visão da eficiência com o tempo**



### 3.1.3 Redução do Tempo de Execução com o Aumento do Número de Threads

O gráfico que ilustra a diminuição do tempo de execução com o aumento do número de threads demonstra claramente o impacto positivo da paralelização na performance do sistema. Observa-se uma redução significativa no tempo de execução ao utilizar 2 threads, passando de 10 minutos e 29 segundos para 6 minutos e 5 segundos. Com 4 threads, o tempo de execução cai ainda mais, chegando a 4 minutos e 10 segundos. No entanto, a partir desse ponto, a redução no tempo de execução se torna menos acentuada, indicando que o sistema está se aproximando de um limite na capacidade de paralelização da tarefa. Essa análise reforça a importância de encontrar um equilíbrio entre o número de threads e o ganho de desempenho, considerando as características da tarefa e os recursos disponíveis.

**Gráfico 3 linha do tempo por threads**



#### **4. Discussão**

Os resultados demonstram o potencial do processamento paralelo para acelerar o processamento de vídeo. A divisão do vídeo em partes menores e o uso de threads permitiram um melhor aproveitamento dos recursos do computador, resultando em um desempenho superior. No entanto, é importante observar que o speedup não aumenta linearmente com o número de threads, indicando que existem limitações na paralelização da tarefa, como a sobrecarga na criação e gerenciamento das threads e a contenção por recursos compartilhados.

#### **5. Conclusão**

Este estudo explorou o processamento paralelo de vídeo, com foco na conversão para preto e branco. Os resultados demonstraram que a divisão do vídeo em partes e o uso de múltiplas threads podem acelerar significativamente o processamento, ultrapassando o speedup desejado de 2.0. No futuro, pretendemos investigar outras técnicas de paralelização e otimização, como o uso de GPUs e a aplicação de filtros mais complexos em vídeos ao vivo.