



UNIVERSIDADE FEDERAL DO PIAUÍ
CENTRO DE CIÊNCIAS NATURAIS
DEPARTAMENTO DE COMPUTAÇÃO
DISCIPLINA: PROCESSAMENTO DIGITAL DE IMAGENS
PROFESSOR: KELSON AIRES

SAMUEL SANTOS ARAUJO

ATIVIDADE PRÁTICA 04

1. Ferramentas Utilizadas

Os programas foram desenvolvidos utilizando a linguagem Python 3 e como suporte foram utilizadas as seguintes bibliotecas: OpenCV 2 e Numpy. Para a construção do vídeo foi utilizado o site [FlexClip](#).

2. Objetivos

A partir dos resultados da Atividade Prática 3 (imagens), montar um vídeo ilustrativo com todas as operações envolvidas. Além disso implementar um algoritmo de codificação (dentre os apresentados em sala de aula) para comprimir cada quadro do vídeo.

3. Especificações do Vídeo

- Tempo** : 47 segundos.
- Quantidade de Quadros**: 1199 quadros.
- Quadros/segundo**: 25.02 quadros/segundo.
- Tamanho**: 1.22 MB (1.286.144 bytes).
- Resolução**: 480x852.

4. Organização do Vídeo

O vídeo se dispõe da seguinte maneira, primeiro é apresentado uma imagem com texto relacionada a operação da atividade 03 e em seguida é apresentado os resultados obtidos, cada um destes durando dois segundos de exibição . Como exemplo as Figuras 1 e 2.



Figura 1. Imagem informando a operação realizada



Figura 2. Resultado da Operação

5. Algoritmo de Compressão

O algoritmo escolhido foi de Run-Length(RLE), pois codifica sem perda e se concentra na remoção de redundâncias espaciais. O algoritmo de compressão só foi aplicado nas imagens obtidas da atividade anterior. Como a maioria delas a redundância espacial é bem grande, concluiu-se que a codificação RLE seria a que melhor se encaixaria para esta aplicação. A figura 3 apresenta o código utilizado.

```

import cv2

#Lista com o endereço das imagens para leitura
imagens = ["frames_video\com_Fecho_Convexo.png",
           "frames_video\com_Fecho_Convexo_na_Original.png",
           "frames_video\Esqueleto do fecho convexo.png",
           "frames_video\esqueleto_vermelho.png",
           "frames_video\sem_Buracos.png",
           "frames_video\sem_pontos.png",
           "frames_video\ fecho_vermelho.png",
           "frames_video\ vermelho.png"]

j = 1, #contador para salvar as imagens
for i in imagens:
    nome = "frames_video\img" + str(j) + ".png" #variavel que guarda nome da imagem comprimida

    img = cv2.imread(i,1) #lendo a imagem original

    cv2.imwrite(nome,img,[16,3]) #salvando a imagem no formato png com a compressão RLE
    j+=1

```

Figura 3. Código Utilizado para Compressão

6. Taxa de Compressão

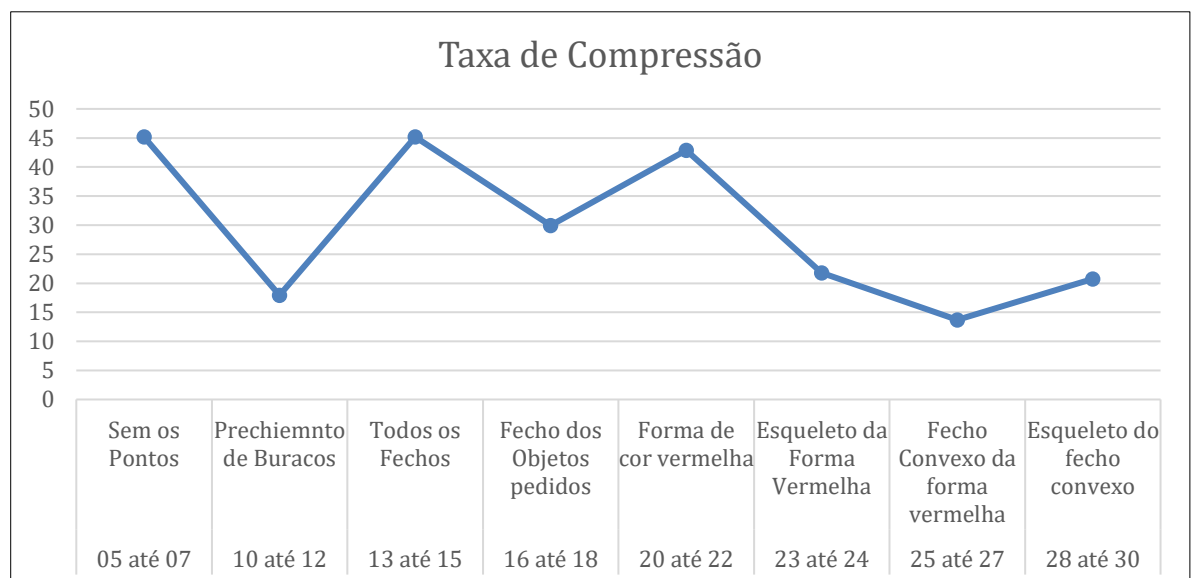


Gráfico 1. Taxa de Compressão ao longo do vídeo

O método utilizado é uma forma simples de compressão onde sequências longas de valores repetidos são armazenadas como um único valor e sua contagem no lugar de sua sequência original. Como as imagens apresentam grandes áreas contínuas de uma mesma cor a menor taxa alcançada foi de 13,69 na imagem que contém o fecho convexo da forma vermelha e a maior de 45,17 na imagem com que apresenta todos os fechos convexos encontrados e na imagem sem os pontos pretos.

No Gráfico 2 apresenta uma comparação entre cada imagem original e sua versão comprimida para ficar visível a discrepância de tamanho entre elas.

As ultimas imagens até parecem que o tamanho é bem próximo mas isso se dá pela escala pois esta diferença é tão grande que fica difícil representar os valores menores.

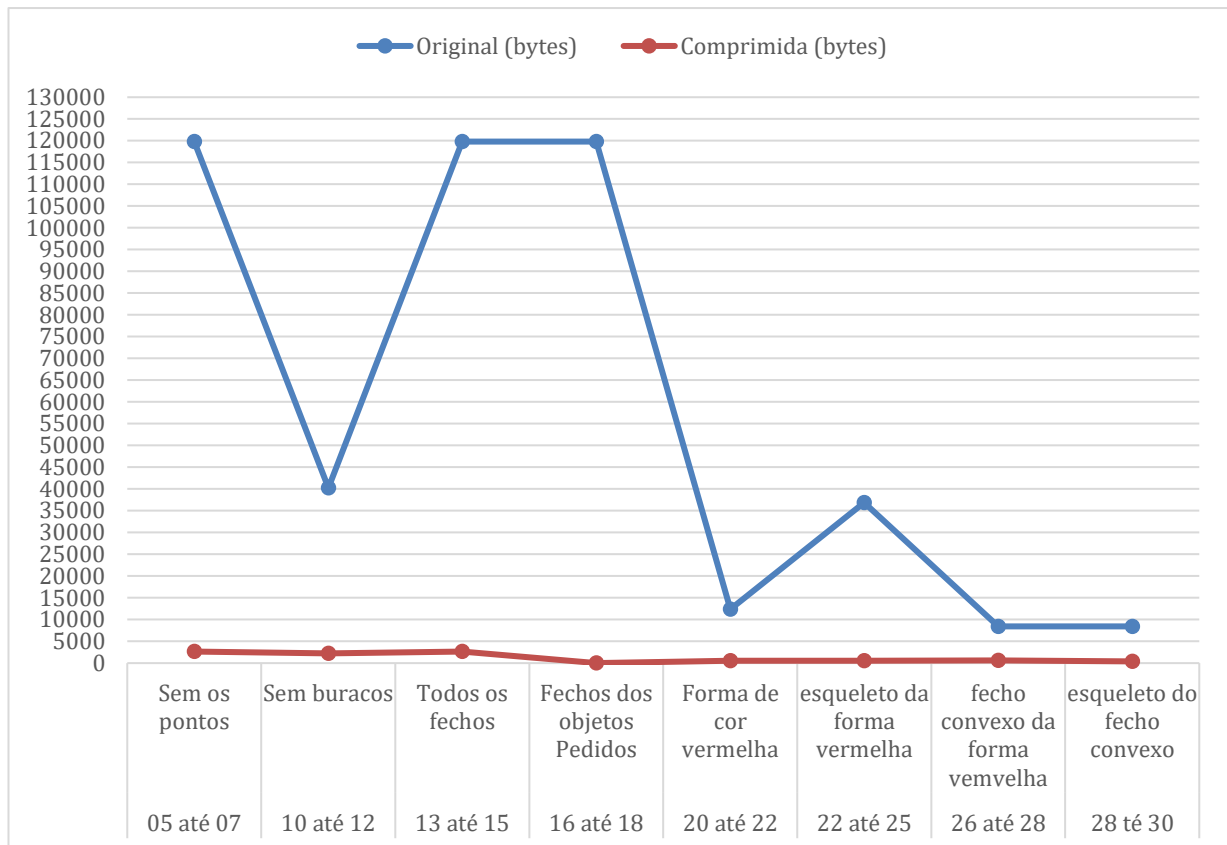


Gráfico 2. Comparação das quantidades bytes da Imagem Original com a Imagem comprimida