

Consumo de recursos de algoritmos de subtração de fundo

Resource consumption of background subtraction algorithms

RESUMO

Fagner Henrique Santos Onesko
fagneronesko@alunosutfpr.edu.br
Universidade Tecnológica Federal
do Paraná, Cornélio Procopio,
Paraná, Brasil.

Silvio Ricardo Rodrigues
Sanches
silviosanches@utfpr.edu.br
Universidade Tecnológica Federal
do Paraná, Cornélio Procopio,
Paraná, Brasil.

Este trabalho apresenta uma análise do consumo de recursos computacionais de algoritmos de subtração de fundo, considerando diferentes resoluções de quadros de vídeo. O objetivo é identificar a proporção que o aumento da resolução afeta o consumo de recursos. Para este estudo foram utilizados 2 vídeos, inicialmente com 10 resoluções cada, e 38 algoritmos de subtração de fundo cujos códigos-fontes estão disponíveis *online*. As resoluções foram trabalhadas até se atingir o limite computacional da máquina utilizada, analisando o tempo de execução, o consumo de memória e o consumo de processador. Como resultado desta pesquisa, através da mediana entre os 38 algoritmos, agrupando-os por resolução, observou-se um comportamento exponencial de consumo desses recursos.

PALAVRAS-CHAVE: Segmentação. Resolução. Consumo. Recursos. Computacionais.

Recebido: 19 ago. 2020.

Aprovado: 01 out. 2020.

Direito autorial: Este trabalho está licenciado sob os termos da Licença Creative Commons-Atribuição 4.0 Internacional.



ABSTRACT

This work presents an analysis of the consumption of computational resources of background subtraction algorithms, considering different resolutions of video frames. The goal is to identify the proportion that increasing the resolution affects resource consumption. For this study, 2 videos were used, initially with 10 resolutions each, and 38 background subtraction algorithms whose source codes are available online. The resolutions were worked on until the computational limit of the machine used was reached, analyzing the execution time, memory consumption and processor consumption. As a result of this research, through the median among the 38 algorithms, grouping them by resolution, an exponential consumption behavior of these resources was observed.

KEYWORDS: Segmentation. Resolution. Consumption. Resources. Computational.

INTRODUÇÃO

A segmentação de uma imagem é a tarefa de subdividi-la em suas regiões (ou objetos) constituintes (GONZALEZ; WOODS; EDDINS, 2002). Algumas dessas regiões são formadas por *pixels* que pertencem a elementos que são de interesse para uma aplicação específica. Um elemento de interesse pode ser, por exemplo, uma pessoa em aplicações como *people tracking* (WREN et al., 1997). Portanto, as regiões da imagem que contêm *pixels* que pertencem às pessoas são as que interessam para a aplicação. Por outro lado, em aplicações como *vehicle accident detection* (PARSA et al., 2020) os veículos são os elementos de interesse.

A subtração de fundo é uma abordagem muito utilizada na segmentação de imagens, onde subtrai-se da imagem atual o objeto de interesse a partir de um fundo estático, podendo utilizar da imagem atual para atualizar o fundo (FERREIRA, 2012). Nas aplicações de vigilância por vídeo, por exemplo, em que a segmentação do quadro de vídeo é a primeira etapa do processo, as câmeras que capturam o vídeo normalmente possuem baixa ou média resoluções. A utilização de câmeras com resoluções mais altas que as utilizadas atualmente podem melhorar o desempenho dos algoritmos de subtração de fundo ainda que mais recursos computacionais sejam necessários.

No decorrer desse trabalho será analisado como a resolução de quadros de vídeo podem impactar no consumo de recursos (tempo de processamento, uso de memória e uso de processador) dos algoritmos de subtração de fundo que representam o estado-da-arte.

MATERIAL E MÉTODOS

Primeiramente, foram utilizados dois vídeos, cada qual com objetos de interesses distintos. O primeiro exhibe pessoas em um ambiente externo, capturado na UTFPR - Câmpus Cornélio Procópio, e o segundo exhibe veículos transitando em uma rodovia, capturado na cidade de Marília. Ambos os vídeos foram gravados com resolução de 4K (3840 x 2160) *pixels*, têm duração de 16 segundos e possuem 51 quadros.

Para a subtração de fundo desses quadros foram utilizados 38 algoritmos, todos eles obtidos da biblioteca *BGSLibrary*, que está implementada na linguagem C++, e é compatível com a biblioteca OpenCV. O código fonte da biblioteca está sob licença do MIT, que permite uso livre para todos os usuários (SOBRAL, 2013).

Os vídeos em 4K foram divididos em 51 frames, e após isso, a resolução foi reduzida para 90%, 80% e assim sucessivamente até se obter 10% do valor original. Com isso, para cada vídeo, obteve-se 10 resoluções, sendo essas: (3840 x 2160) *pixels*, (3456 x 1944) *pixels*, (3072 x 1728) *pixels*, (2688 x 1512) *pixels*, (2304 x 1296) *pixels*, (1920 x 1080) *pixels*, (1536 x 864) *pixels*, (1152 x 648) *pixels*, (768 x 432) *pixels*, (384 x 216) *pixels*.

A partir dos vídeos, até se atingir o limite computacional da máquina Core i7-4500U (1.8 GHz), 8GB RAM (*dual channel*), Linux Debian 9 "Stretch" (64 bits), realizou-se a segmentação. O limite da máquina foi 30% das resoluções, nesse

caso: (384 x 216) *pixels*, (768 x 432) *pixels* e (1152 x 648) *pixels*. Durante esse trabalho, essas resoluções serão citadas como 1, 2 e 3, respectivamente.

Por fim, foi realizado o levantamento do consumo de recursos de cada algoritmo, para cada um dos vídeos e para cada uma das 3 resoluções trabalhadas. Desse modo, gerou-se dados para cada um dos algoritmos, e para as resoluções em que foi possível realizar a segmentação.

Para informação de tempo de execução foi utilizado a classe "high_resolution_clock" do C++. Já para o consumo de processador, foi utilizado o arquivo "stat" do Linux, localizado no "/proc/stat", o qual exibe informações de todos os núcleos, tais como tempo de operação do processador, e tempo em que este permaneceu parado (aguardando alguma informação ou até mesmo ocioso). Nesse trabalho será apresentado somente o processamento total ativo, já que este resultado mostrará a soma do consumo de todos os núcleos, e que de fato estavam processando algo. Por fim o consumo de memória, foi obtido a partir do arquivo "stat" localizado no diretório "/proc/self/stat". Segundo (KERRISK, 2020) esse arquivo fornece *status* tanto da "VM" quanto da "RSS", que indicam o consumo de memória virtual em *Bytes*, e a memória residente ou real, também em *Bytes*. Para analisar-se os resultados, fez-se uso de um *script* em Linguagem R 3.3.3, de autoria própria.

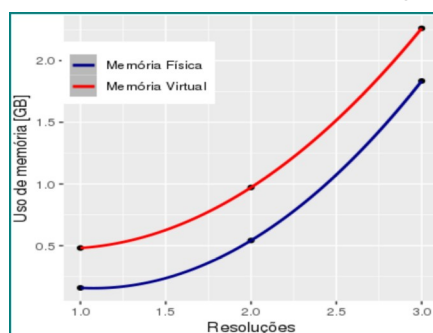
Para o tempo, a unidade de medida utilizada será segundos, para a memória *Gigabytes* e para o processamento *User Hz*. Uma unidade do *kernel* Linux para indicar o consumo de processador em *Heartz*, definindo um valor padrão inicial e evitar anomalias que ocorriam em versões antigas.

Como nesses CSVs, observou-se discrepância dos dados, ou seja, alguns algoritmos possuem elevados valores de consumo quando comparados a outros, para evitar que esses valores interfiram no resultado, optou-se por utilizar a mediana, já que nesse caso tende a ser uma medida mais justa (JUNIOR, 2015).

RESULTADOS E DISCUSSÃO

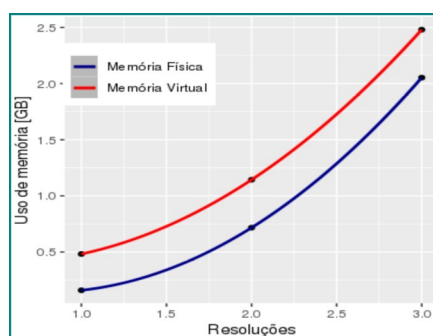
Para o consumo de memória, obteve-se as Figuras 1 e 2, as quais mostram os dados referentes a memória Física e Virtual, para o vídeo gravado em Cornélio Procópio e para o de Marília, respectivamente.

Figura 1 – Mediana do consumo de memória para Vídeo 1



Fonte: elaborada pelo autor.

Figura 2 – Mediana consumo de memória para Vídeo 2

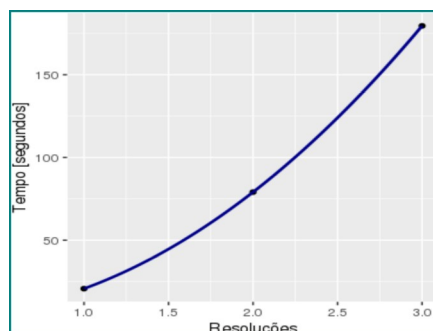


Fonte: elaborada pelo autor.

Se observa um comportamento exponencial no consumo de memória, tanto para a memória física, quanto para a memória virtual, nos dois vídeos, observando um consumo ainda maior para a memória virtual, já que a mesma se refere ao espaço alocado, e não necessariamente utilizado. Nesse caso, para o consumo de memória física, o espaço de memória que foi realmente utilizado, chegou a quase 2 Gigabytes para o vídeo 1, e chegou a ultrapassar para o vídeo 2.

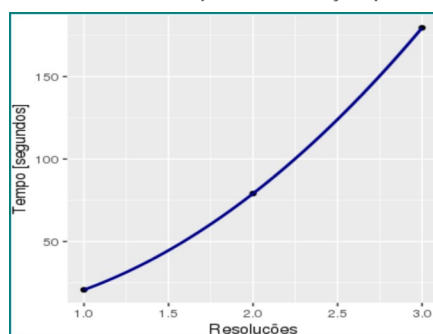
Para o tempo de execução, obteve-se as Figuras 3 e 4, as quais mostram os dados levantados, para os dois vídeos dessa pesquisa.

Figura 3 – Mediana tempo de execução para Vídeo 1



Fonte: elaborada pelo autor.

Figura 4 – Mediana tempo de execução para Vídeo 2

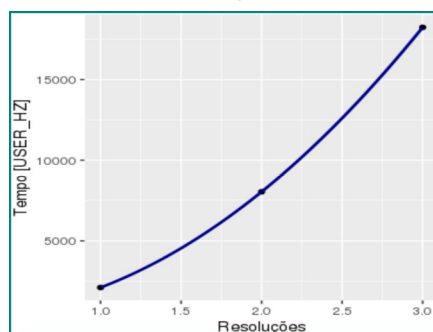


Fonte: elaborada pelo autor.

Se observa para o tempo de processamento, um comportamento exponencial, quando se tem a mediana. Se nota, pelo gráfico que para futuras resoluções esse tempo poderia se tornar extremamente elevado, e quando em conjunto com as outras métricas, justifica a saturação do hardware em questão.

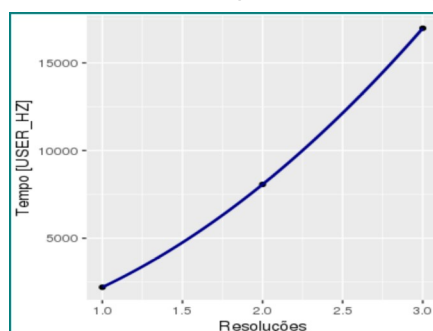
Para o processamento, obteve-se o consumo total do processador, ou seja, a soma do consumo de todos os 4 núcleos presentes na máquina em *User Hz*, e este resultado está ilustrado nas Figuras 5 e 6.

Figura 5 – Mediana consumo de processador total para Vídeo 1



Fonte: elaborada pelo autor.

Figura 6 – Mediana consumo de processador total para Vídeo 2



Fonte: elaborada pelo autor.

Para o consumo de processador ativo total, ou seja, o consumo deste hardware, quando utilizado de fato pela aplicação, em sua totalidade também tem-se um comportamento exponencial.

CONCLUSÃO

Para todas as análises de consumo: uso de memória, tempo de execução e processamento, observa-se um comportamento exponencial de acordo com o aumento da resolução. Isso indica, que resoluções muito altas, como por exemplo a de 4K podem tornar a subtração de fundo inviável, exigindo alto recurso computacional para ser realizada. Em aplicações que não possuem uma máquina com grande poder computacional, resoluções muito altas para subtração de fundo, devem ser evitadas.

Antes de se escolher uma resolução, deve-se levar em conta os algoritmos que serão utilizados, e os recursos disponíveis para a aplicação. Como sabe-se recursos computacionais são finitos, portanto adotar resoluções não tão altas, que permitem uma boa qualidade de segmentação para a aplicação, é crucial para se ter uma aplicação com qualidade, tempo e consumo de recursos aceitáveis.

Portanto, observar os recursos computacionais disponíveis, assim como a qualidade de segmentação que a resolução pode oferecer, e achar uma qualidade que equilibre tais funções é de extrema relevância.

Neste trabalho, não foi comparado o impacto que a resolução do vídeo causa na qualidade da segmentação com o consumo de recursos computacionais. Como essa análise pode fornecer uma resolução que fique dentro dos parâmetros de qualidade de segmentação aceitável com consumo de recursos aceitável, trabalhos futuros podem

AGRADECIMENTOS

Agradeço ao meu orientador Silvio Ricardo Rodrigues Sanches, por toda ajuda no desenvolvimento dessa pesquisa. O presente trabalho foi realizado com o apoio da Universidade Tecnológica Federal do Paraná/Brasil, por meio do Programa Institucional de Voluntariado em Iniciação Científica da UTFPR (PVICT).

REFERÊNCIAS

FERREIRA, C. S. Implementação do algoritmo de subtração de fundo para detecção de objetos em movimento, usando sistemas reconfiguráveis, 2012.

JUNIOR, M. para análise de dados. A B, 2015.

GONZALEZ, R. C.; WOODS, R. E.; EDDINS, S. Image segmentation. Digital image processing, Prentice-Hall, Inc, v. 2, 2002.

KERRISK, M. Linux Programmer's Manual. [S.l.: s.n.], abr. 2020. <http://man7.org/linux/man-pages/man5/proc.5.html>.

PARSA, A. B. et al. Toward safer highways, application of XGBoost and SHAP for real-time accident detection and feature analysis. Accident Analysis & Prevention, Elsevier, v. 136, p. 105405, 2020

SOBRAL, A. BGSLibrary: An OpenCV C++ Background Subtraction Library. In: IX Workshop de Visão Computacional (WVC'2013). Rio de Janeiro, Brazil: [s.n.], jun. 2013. Disponível em: <https://github.com/andrewssobral/bgslibrary>

WREN, C. R. et al. Pfnder: Real-time tracking of the human body. IEEE Transactions on pattern analysis and machine intelligence, IEEE, v. 19, n. 7, p. 780–785, 1997.

WREN, C. R. et al. Pfinder: Real-time tracking of the human body. IEEE
Transaction on pattern analysis and machine intelligence, IEEE, v. 19, n. 7, p. 780-
785, 1997.