

Avaliação de Desempenho de Novas Funcionalidades de *Telestration* para um Sistema de Telemedicina

Angelo Artur Vilani (PIBITI/CNPq)¹, Wu Feng Chung^{1,2}, Weber Shoity Resende Takaki^{1,3}, Huei Diana Lee (Orientadora)¹

Laboratório de Bioinformática (LABI)/Universidade Estadual do Oeste do Paraná (Unioeste)¹, Faculdade de Ciências Médicas (FCM)/Universidade Estadual de Campinas (Unicamp)², Universidade Federal da Integração Latino-Americana (Unila)³

angelo.vilani@unioeste.br

Objetivos

Avaliar o desempenho computacional de novas implementações das funcionalidades de *telestration* em um sistema de telemedicina.

Métodos e Procedimentos

A telemedicina integra tecnologias de informação e comunicação (TICs) à saúde. Nesse contexto, a ferramenta de *telestration* permite anotações gráficas em vídeos, em tempo real, para fins didáticos ou analíticos. O Sistema de Monitoramento Médico, Acompanhamento Remoto de Procedimentos e Tele-mentoria (SMMAR-T Mentor) [1] adota esta tecnologia; contudo, a viabilidade da implementação de novas funcionalidades nesse sistema requer uma avaliação de desempenho criteriosa de modo a não prejudicar a usabilidade e os requisitos de tempo real e de baixa latência, necessários nas transmissões de áudio e vídeo. Assim, foram implementadas duas versões de um programa web de *telestration* em JavaScript para avaliação e possível incorporação ao SMMAR-T Mentor. A versão base foi denominada v1, e a versão otimizada v2, à qual foi adicionado um *cache* para os desenhos. Os materiais utilizados nos experimentos foram: computador do tipo *notebook* equipado com processador Intel® Core™ i7-13700H, 16 GB de memória DDR5-5.200MHz, e sistema operacio-

nal Windows 11; linguagens de programação JavaScript e Python; navegador Mozilla Firefox versão 141.0; aplicativo OBS Studio para reprodução de vídeo de procedimento de videoendoscopia para simular aplicação real. A análise considerou o consumo de recursos e seguiu o protocolo de testes com os seguintes cenários: (1) base (BAS), caracterizado pela aplicação sem nenhuma ação adicional além da captura e apresentação de vídeo; (2) desenho contínuo (DES), caracterizado por uma única ação de desenho livre, durante 20 segundos; e (3) carga de desenhos (CAR), caracterizado pelo desenho de 100 retângulos no intervalo de 20 segundos (Figura 1). As variáveis monitoradas no teste foram o uso de CPU pelo programa de *telestration* e a taxa de quadros do vídeo (FPS) com as marcações de *telestration*. A coleta dos dados utilizou o Firefox Profiler [2]. Em seguida, cada cenário foi executado por um *script* de automação de *desktop* (AutoHotKey) que automatizou testes em ciclos de 34s, controlando ativação da aplicação, execução do teste e gravação do perfil. O perfil de desempenho obtido foi processado por *scripts* em Python para extração dos dados de uso de CPU, enquanto as informações de FPS foram coletadas automaticamente pela aplicação durante os testes e armazenadas em arquivos. Os resultados foram então organizados em seis grupos (BAS, DES e CAR, todos para v1 e v2).

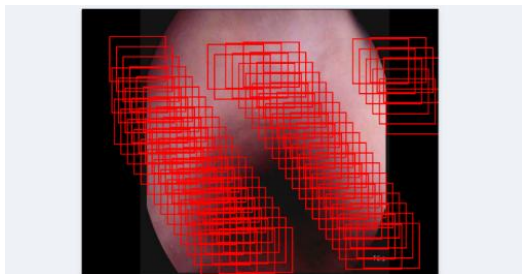


Figura 1: Exemplo de execução do cenário CAR

Os resultados foram analisados considerando intervalo de confiança de 95%.

Resultados

A comparação entre os seis cenários foi realizada usando o teste de Kruskal-Wallis, seguido do pós-teste de Dunn. Na Tabela 1 estão apresentadas as comparações de uso da CPU entre os grupos BAS-v1 (A), BAS-v2 (B), DES-v1 (C) e DES-v2 (D), CAR-v1 (E), CAR-v2 (F), indicando se houve diferença estatisticamente significativa entre os pares (em negrito) e o respectivo valor de p.

Tabela 1: Comparações entre os cenários experimentais com pós-teste de Dunn.

	B	C	D	E	F
A	>0,9999	<0,0001	<0,0001	<0,0001	>0,9999
B	–	<0,0001	<0,0001	<0,0001	0,6202
C	–	–	<0,0001	<0,0001	<0,0001
D	–	–	–	<0,0001	<0,0001
E	–	–	–	–	<0,0001

Na Tabela 2 estão apresentados os resultados de consumo percentual médio de CPU e respectivos desvios padrão para os seis cenários de teste. A terceira coluna apresenta, ainda, a mediana de cada grupo.

Tabela 2: Média e desvio-padrão do uso de CPU

Caso de Teste		Média (DP)	Mediana
BAS -	v1	15,26 (28,21)	0,22
	v2	14,20 (27,46)	0,22
DES -	v1	21,11 (32,37)	2,53
	v2	18,37 (30,82)	1,96
CAR -	v1	18,37 (31,10)	1,31
	v2	14,35 (27,50)	0,29

A diferença no uso da CPU entre as versões pode ser explicada pela adição do *cache* de

desenhos em v2, pois evita a necessidade de refazer os cálculos a cada quadro, mantendo as formas salvas em *cache*, operação esta com baixo custo ao ser processada.

Para os resultados referentes à taxa de quadros, não se observou diferença estatisticamente significativa entre os seis cenários ($p > 0,9999$). O valor médio se manteve em 62,30 FPS, com desvio padrão de 0,42. Assim, mesmo que se observe um resultado sem diferença estatisticamente significativa, no contexto de transmissão de vídeo em tempo real, v1 exigirá um processamento adicional, pois os desenhos e o quadro permanecem separados até antes de serem transmitidos. Desse modo, v2 apresenta vantagem em relação a v1 por não exigir processamento adicional, unindo os desenhos diretamente ao quadro no *cache*.

Conclusões

De acordo com as análises, a versão otimizada (v2) da ferramenta de *telestration* apresentou melhor desempenho no cenário de múltiplos desenhos, mostrando-se tão estável quanto a aplicação ociosa, o que sugere que a funcionalidade pode ser incorporada sem comprometer o desempenho geral do sistema. Essa versão será então integrada ao sistema de telemedicina.

Os autores atuaram em todas as etapas de confecção e escrita, aprovaram a versão final do resumo e declaram não haver conflito de interesses.

Agradecimentos

Ao CNPq pela bolsa PIC/PIBITI (Edital 03/2024 – PRPPG/Unioeste, Processo 154797/2024).

Referências

- [1] Coy, C., Lee, H.D., Ensina, L., Spolaôr, N., Takaki, W., & Wu, F.C. (2022). Sistema de Monitoramento Médico, Acompanhamento Remoto de Procedimentos e Telementoria - SMMAR-T Mentor (Registro BR512022002022-0).
- [2] Mozilla Foundation (2024). Firefox Profiler. Disponível em: <https://profiler.firefox.com/>. Acesso em: 26 de jul. de 2025.