

**Avaliação de desempenho de um módulo de *telestration* para  
sistemas de telemedicina em tempo real**

Angelo Artur Vilani (PIBITI/CNPq/Unioeste), Huei Diana Lee (Orientador), Weber Takaki, Alexandre Peiter Ferraz, Wu Feng Chung, e-mail: [angelo.vilani@unioeste.br](mailto:angelo.vilani@unioeste.br)

Universidade Estadual do Oeste do Paraná/Centro de Engenharias e  
Ciências Exatas/Foz do Iguaçu, PR

**Área/subárea:** Ciências Exatas e da Terra/Ciência da Computação

**Palavras-chave:** Telementoria, eHealth, Desempenho.

**Resumo**

O telestrator, ferramenta que permite realizar anotações gráficas sobre vídeo, é um recurso relevante para a telementoria em sistemas de telemedicina, favorecendo a comunicação em tempo real e o ensino à distância. A necessidade de alto desempenho da aplicação, de modo a não comprometer a fluidez da transmissão, caracteriza-se como um desafio técnico de implementação dessa ferramenta. Neste trabalho, é apresentada a avaliação do desempenho de duas implementações de um módulo de telestration (v1 e v2), com o objetivo de validar sua adequação a sistemas de telemedicina em tempo real. Um protocolo experimental automatizado foi proposto para comparar o consumo de processamento e o número de quadros por segundo sob baixa e alta carga de anotações. A versão v2 mostrou-se mais eficiente e, portanto, mais adequada para integração em sistemas de telemedicina em tempo real.

**Introdução**

A telemedicina, definida como a prestação de serviços de saúde à distância por meio de tecnologias da informação, amplia o alcance da assistência médica e da educação continuada (Fernandes et al., 2023). Nesse contexto, a telementoria — orientação remota de procedimentos — tem impacto na formação de especialistas, especialmente em técnicas cirúrgicas minimamente invasivas (Tanaka et al., 2024).

Entre as tecnologias que sustentam essa prática destaca-se o módulo de telestration, que permite marcações gráficas sobre o vídeo do procedimento,

facilitando a comunicação entre mentor e aprendiz. Embora estudos indiquem ganhos na precisão cirúrgica e na eficácia do ensino (Fernandes et al., 2023; Tanaka et al., 2024), a implementação desse recurso em navegadores web ainda enfrenta desafios de desempenho, sobretudo no uso de CPU e na latência de renderização.

Grande parte das publicações aborda aspectos clínicos e pedagógicos, mas há escassez de estudos sobre eficiência computacional e escalabilidade em tempo real. Assim, este trabalho avalia o desempenho de duas implementações de um módulo de telestratagem integrável ao Sistema de Monitoramento Médico, Acompanhamento Remoto de Procedimentos e Telementoria (SMMAR-T Mentor) (Coy et al., 2022), com o objetivo de identificar a arquitetura mais eficiente quanto ao consumo de CPU e à taxa de quadros por segundo (FPS).

O diferencial do estudo está na abordagem experimental automatizada, que utiliza ferramentas de análise de desempenho (Firefox Profiler) e automação (AutoHotKey) para mensurar o impacto das estratégias de renderização, contribuindo para o desenvolvimento de módulos mais leves e estáveis para aplicações em telementoria.

## Material e Métodos

O desenvolvimento foi realizado em um notebook Lenovo Legion 5i, equipado com processador Intel® Core™ i7-13700H, 16 GB de memória DDR5 de 5200 MT/s e sistema operacional Windows 11. Foram utilizadas as linguagens JavaScript, HTML5 e CSS3 (Mozilla, 2024a) para o desenvolvimento do protótipo. Para a avaliação experimental, empregaram-se o software OBS Studio (OBS Project, 2024) para geração de um fluxo de vídeo padronizado, o Firefox Profiler (Mozilla, 2024b) para coleta de dados e a ferramenta de automação AutoHotKey (AutoHotKey, 2024).

Duas versões do módulo foram implementadas com as seguintes características:

- Versão v1 (Renderização Direta): Utilizou um único `<canvas>`, no qual, a cada quadro, o canvas era limpo e tanto o vídeo quanto todos os objetos gráficos eram redesenhados sequencialmente;
- Versão v2 (Renderização com Cache): Utilizou um segundo `<canvas>` auxiliar, fora da tela, para armazenar uma imagem única de todas as anotações. O ciclo de animação foi simplificado para desenhar apenas o quadro de vídeo e, sobre ele, a imagem do *cache*.

Um protocolo experimental em ambiente controlado foi proposto para validação dos protótipos. As métricas avaliadas foram o percentual de uso da CPU e a taxa de quadros por segundo (FPS). Foram definidos três cenários de teste: **BAS** – linha de base, sem anotações; **DES** – desenho contínuo; e **CAR** – carga de 100 desenhos automatizados. Cada caso de teste foi repetido três vezes durante 20 segundos, e os dados obtidos foram agrupados por cenário de teste e versão do módulo, produzindo seis conjuntos de dados: BAS-v1, BAS-v2, DES-v1, DES-v2, CAR-v1 e CAR-v2. Ao final, os grupos foram comparados entre si (todos contra todos) utilizando o teste de Kruskal-Wallis e o pós-teste de Dunn, com intervalo de confiança de 95%.

## Resultados e Discussão

A análise da taxa de quadros por segundo (FPS) demonstrou que ambas as versões mantiveram fluidez visual em todos os cenários, com média de 62,30 FPS, sem diferença estatisticamente significativa ( $p > 0,9999$ ) entre os grupos. Em relação ao uso de CPU, observaram-se variações relevantes entre os casos de teste. A Tabela 1 apresenta os valores de p obtidos na análise estatística, destacando em negrito as diferenças estatisticamente significativas, enquanto a Tabela 2 resume a média, o desvio-padrão e a mediana de consumo de CPU para cada versão e cenário.

**Tabela 1** – Comparação estatística para uso de CPU (Valores de p).

|                 | BAS - v2 | DES - v1          | DES - v2          | CAR - v1          | CAR - v2          |
|-----------------|----------|-------------------|-------------------|-------------------|-------------------|
| <b>BAS - v1</b> | >0,9999  | <b>&lt;0,0001</b> | <b>&lt;0,0001</b> | <b>&lt;0,0001</b> | >0,9999           |
| <b>BAS - v2</b> | –        | <b>&lt;0,0001</b> | <b>&lt;0,0001</b> | <b>&lt;0,0001</b> | 0,6202            |
| <b>DES - v1</b> | –        | –                 | <b>&lt;0,0001</b> | <b>&lt;0,0001</b> | <b>&lt;0,0001</b> |
| <b>DES - v2</b> | –        | –                 | –                 | <b>&lt;0,0001</b> | <b>&lt;0,0001</b> |
| <b>CAR - v1</b> | –        | –                 | –                 | –                 | <b>&lt;0,0001</b> |

**Tabela 2** - Consumo de CPU (Média, Desvio-Padrão, Mediana).

| Cenário de Teste | Média  | Desvio-Padrão | Mediana |
|------------------|--------|---------------|---------|
| <b>BAS - v1</b>  | 15,26% | 28,21%        | 0,22%   |
| <b>BAS - v2</b>  | 14,20% | 27,46%        | 0,22%   |
| <b>DES - v1</b>  | 21,11% | 32,37%        | 2,53%   |
| <b>DES - v2</b>  | 18,37% | 30,82%        | 1,96%   |
| <b>CAR - v1</b>  | 18,37% | 31,10%        | 1,31%   |

| Cenário de Teste | Média  | Desvio-Padrão | Mediana |
|------------------|--------|---------------|---------|
| <b>CAR - v2</b>  | 14,35% | 27,50%        | 0,29%   |

No cenário de linha de base (BAS), o comportamento das duas versões foi equivalente, sem diferença estatisticamente significativa. Em situação de desenho contínuo (DES), a v2 demonstrou menor utilização do processador em relação à v1, apresentando diferença estatisticamente significativa ( $p < 0,0001$ ).

A diferença mais evidente surgiu sob carga intensiva (CAR), quando a primeira implementação demandou mais processamento (18,37%) que a segunda (14,35%), também com diferença estatisticamente significativa ( $p < 0,0001$ ). Além disso, o desempenho da v2 manteve-se estável entre sua linha de base e o cenário de carga ( $p = 0,6202$ ), sem diferença estatisticamente significativa, indicando resistência ao aumento de anotações. Em contraste, a v1 apresentou acréscimo proporcional na demanda de CPU conforme a complexidade dos testes, confirmando as limitações da renderização direta.

O desempenho da v2, caracterizado por baixo consumo de CPU e FPS estável, é um pré-requisito para a baixa latência, aspecto fundamental em telementoria (Takemasa et al., 2023). Um telestrator ineficiente torna-se um gargalo de processamento que gera atrasos na transmissão, independentemente da latência da rede. Nesse sentido, a v2, ao manter o uso de CPU reduzido e constante, contribui para a fluidez geral do sistema — um requisito técnico essencial para viabilizar sua aplicação em telementoria e favorecer a utilização da ferramenta em dispositivos com diferentes capacidades de processamento.

## Conclusões

A versão com Renderização com Cache (v2) da ferramenta de *telestration* apresentou melhor desempenho, tendo sido superior no cenário de múltiplos desenhos, o que sugere que a funcionalidade pode ser incorporada sem comprometer o desempenho geral do sistema SMMAR-T Mentor.

## Agradecimentos

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

## Referências

AutoHotKey. (2024). *AutoHotKey*. <https://www.autohotkey.com>. Acesso em 25 de setembro de 2025.

Coy, C. S. R., Ayrizono, M. L. S., Leal, R. F., Maletzke, A. G., Chung, W. F., Lee, H. D., Fagundes, J. J., Machado, R. B., Maciel, J. N. & Voltolini, R. F. (2012). Patente BR 10 2012 033125 0. Método de Telemedicina para o Acompanhamento Remoto e em Tempo Real de Procedimentos Médicos. Universidade Estadual do Oeste do Paraná (UNIOESTE) / Universidade Estadual de Campinas (UNICAMP).

Coy, C. S. R., Lee, H. D., Chung, W. F., Spolaôr, N., Ensina, L. A. & Takaki, W. S. R. (2022). Sistema de Monitoramento Médico, Acompanhamento Remoto de Procedimentos e Telementoria - SMMAR-T Mentor (Registro BR512022002022-0). Universidade Estadual do Oeste do Paraná (UNIOESTE) / Universidade Estadual de Campinas (UNICAMP).

Fernandes, R. D., Ghasroddashti, A., Sorefan-Mangou, F., Williams, E., Choi, K., Fasola, L., Szasz, P. & Zevin, B. (2023). Educational effectiveness of telementoring as a continuing professional development intervention for surgeons in practice: A systematic review. *Annals of Surgery Open* 4, e341.

Mozilla. (2024a). *JavaScript*. <https://developer.mozilla.org/pt-BR/docs/Web/>. Acesso em 17 de setembro de 2025.

Mozilla. (2024b). *Firefox Profiler*. <https://profiler.firefox.com>. Acesso em 17 de setembro de 2025.

OBS Project. (2024). *OBS Studio*. <https://obsproject.com>. Acesso em 17 de setembro de 2025.

Takemasa, I., Okuya, K., Okita, K., Akizuki, E., Miyo, M., Ishii, M., Miura, R., Ichihara, M., Takahiro, K., Oki, E., Takatsuki, M., Eguchi, S., Ichikawa, D., Kitagawa, Y., Sakai, Y. & Mori, M. (2024). Tele-proctoring for minimally invasive surgery across Japan: An initial step toward a new approach to improving the disparity of surgical care and supporting surgical education. *Annals of Gastroenterological Surgery* 8, 356–364.

Tanaka, Y., Abe, D., Inaji, M., Hara, S., Sakai, R. & Maehara, T. (2024). Intraoperative telestration system in endoscopic transsphenoidal surgery contributes to improved surgical safety and efficient surgical education. *World Neurosurgery* 190, e348–e354.