

E-motion: Uma Abordagem de Visão Computacional para Inclusão de Pessoas com Limitações Motoras em Jogos Digitais

André L. R. Cabral¹, Douglas N. S. Gomes¹, João P. D. Estevão¹,
Ricardo S. P. G. Cerqueira¹, Victor S. Lima¹,
Felipe D. Cunha¹, Matheus B. Pereira¹, Matheus A. Souza¹

¹Instituto de Ciências Exatas e Informática
Pontifícia Universidade Católica de Minas Gerais (PUC Minas)

{alrcabral, dnsngomes, joao.estevao,
ricardo.cerqueira, victor.lima.1478515}@sga.pucminas.br,
{felipe, matheusbarros, matheusalcantara}@pucminas.br

Abstract. *Digital games can support leisure, social participation and rehabilitation, but many interfaces still assume fine motor control of the hands. This paper presents e-motion, a webcam-based virtual controller for turn-based games that uses MediaPipe Hands to map hand position to virtual zones, avoiding continuous hand-shape gestures. The prototype was evaluated through an exploratory performance benchmark with real camera input and a functional observation with users without diagnosed motor impairments. Across 10 runs from four participants, the full pipeline reached 61.22 FPS with one worker and 106.89 FPS with four workers in the strong scalability experiment, with no frame drops. The study does not claim clinical validation.*

Resumo. *Jogos digitais podem apoiar lazer, participação social e reabilitação, mas muitas interfaces ainda dependem de controle fino das mãos. Este trabalho apresenta o e-motion, um controle virtual por webcam para jogos em turnos, que usa o MediaPipe Hands para mapear a posição das mãos em zonas virtuais, evitando gestos contínuos de formato da mão. O protótipo foi avaliado em benchmark exploratório de desempenho com entrada real de câmera e em observação funcional com usuários sem diagnóstico de limitação motora. Em 10 execuções com quatro participantes, o pipeline completo obteve 61,22 FPS com um worker e 106,89 FPS com quatro workers na escalabilidade forte, sem descarte de frames. O estudo não reivindica validação clínica.*

1. Introdução

Jogos digitais fazem parte da rotina de lazer de muitos usuários e também aparecem em contextos de educação, saúde e reabilitação. A literatura descreve benefícios sociais e emocionais associados ao jogo [Granic et al. 2014], além de aplicações de Jogos Sérios em reabilitação física [Vieira et al. 2021], saúde mental [Dewhirst et al. 2022] e educação [de Carvalho and Coelho 2022]. No entanto, gamepads e teclados ainda exigem destreza manual, força de prensão e movimentos repetidos, o que pode excluir pessoas com limitações motoras nos membros superiores.

No Brasil, 14,4 milhões de pessoas declararam alguma deficiência no Censo de 2022 [Instituto Brasileiro de Geografia e Estatística 2023]. Esse dado não define diretamente o público do sistema, mas ajuda a dimensionar a importância de interfaces de entrada mais flexíveis.

Embora existam controles adaptativos e recursos de acessibilidade em alguns jogos, essas soluções nem sempre estão disponíveis, podem depender de hardware específico e precisam ser ajustadas ao perfil motor do usuário [Brown and Anderson 2021]. Este trabalho investiga uma alternativa baseada em webcam comum, no qual o usuário move as mãos sobre uma superfície de apoio, e o sistema converte a posição delas em comandos de teclado. O recorte é intencionalmente limitado a usuários que preservam algum controle dos antebraços e conseguem manter as mãos no campo de visão da câmera.

O objetivo deste trabalho é desenvolver e avaliar um sistema de controle virtual por gestos baseado em visão computacional que permita a interação com jogos digitais através de movimentos simples sobre uma superfície plana. O *e-motion* foca em configuração simples, baseada em câmera monocular, que possa ser executada em computadores convencionais e que evite um controle físico e uso gestos finos com os dedos, exigências comuns em outras soluções.

A pergunta que orienta a avaliação é: um mapeamento posicional por visão computacional consegue manter vazão suficiente e baixa perda de frames para controlar um jogo em turnos em ambiente exploratório com webcam comum? Neste estudo, considerou-se suficiente manter taxa média acima de 30 FPS no pipeline completo, sem descarte médio de frames nas execuções agregadas. As contribuições são: (i) uma arquitetura de calibração e mapeamento bimanual por zonas; (ii) uma avaliação exploratória com frames reais de câmera; e (iii) uma discussão explícita das limitações antes de testes com o público-alvo real.

O artigo está organizado da seguinte forma: A Seção 2 posiciona a proposta em relação a visão computacional e acessibilidade em jogos. A Seção 3 descreve a implementação, a calibração, o mapeamento das zonas e a atuação sobre o jogo. A Seção 4 apresenta o protocolo, os resultados e as ameaças à validade. A Seção 5 conclui o trabalho.

2. Trabalhos Relacionados

A acessibilidade em jogos digitais tem sido abordada por três caminhos principais: adaptar o controle físico tradicional, trocar o dispositivo de entrada por hardware especializado e reconhecer sinais corporais por visão computacional sem contato [Yuan et al. 2011]. O *e-motion* se aproxima da terceira linha, mas com um recorte específico, baseado em posição das mãos em vez de forma das mãos. Esta seção organiza os trabalhos correlatos em duas frentes e termina com uma comparação direta entre o protótipo proposto e abordagens consolidadas na literatura, considerando hardware, custo, precisão e carga física.

2.1. Detecção de Gestos por Visão Computacional

Estudos recentes mostram que redes neurais profundas reconhecem gestos manuais com taxas altas de acerto em condições controladas. [Sharma and Singh 2021] propuseram

uma CNN otimizada que atingiu 99,96% na Língua de Sinais Indiana e 100% na Americana, com robustez sob rotação e mudança de escala. Esses resultados confirmam a viabilidade da classificação de gestos com modelos enxutos, mas dependem de poses bem definidas e de mãos com anatomia próxima à norma do conjunto de treino.

Para aplicações interativas em hardware comum, [Zhang et al. 2020] apresentaram o MediaPipe Hands, um pipeline de dois estágios que combina um detector de palma e um modelo de landmarks. A solução roda em tempo real em GPUs móveis a partir de uma única imagem RGB e foi adotada neste trabalho como base de detecção. O *e-motion* usa apenas a posição do *Landmark 9* para localizar a mão dentro de uma área calibrada. [Gu et al. 2022] mostram, ainda, que o MediaPipe mantém robustez razoável em mãos com deformidades causadas por danos nervosos, o que sustenta seu uso para usuários com alterações anatômicas leves a moderadas. O sistema utiliza como critério apenas a posição da mão em uma zona da grade, sem analisar a forma, já que poses com punho fechado ou mão aberta podem ser tão pesadas quanto pressionar botões pequenos para quem tem dor articular, rigidez ou tremor.

2.2. Acessibilidade em Jogos e Interfaces de Entrada Alternativas

A interação com jogos comerciais ainda assume motricidade fina dos dedos. Em pacientes com Artrite Reumatoide, por exemplo, rigidez articular e dor crônica reduzem a destreza necessária para operar gamepads e teclados convencionais [Armanini et al. 2015]. Padrão semelhante aparece em outras condições que afetam preensão, força e movimentos rápidos. O levantamento de [Yuan et al. 2011] indica que adaptações de acessibilidade em jogos comerciais ainda são escassas, mesmo quando o público potencial é amplo.

A literatura de tecnologias assistivas oferece respostas distintas para esse problema. O *Camera Mouse* [Betke et al. 2002] rastreia pontos da face em vídeo monocular para movimentar o cursor, sem qualquer hardware especializado, e influenciou várias interfaces sem contato posteriores. Em jogos, [Taheri et al. 2021] avaliaram um controlador acionado por expressões faciais com indivíduos com deficiências motoras severas e relataram alta aceitação. [Vickers et al. 2013] demonstraram o uso de eye trackers comerciais para tarefas de locomoção em jogos imersivos, com bom desempenho, mas dependência de hardware dedicado e calibração frequente. No lado do controle físico, o Xbox Adaptive Controller tornou-se referência ao manter o gamepad como base e permitir conectar botões e *switches* de baixa força, beneficiando usuários com força reduzida sem dispensar equipamento adicional.

Esses trabalhos cobrem perfis motores diferentes, e nenhum atende a todos. Tratar acessibilidade como propriedade única da interface é, talvez, o principal erro a evitar. Um sistema sem contato funciona para um usuário com dor nas mãos e pode ser inadequado para outro que não consegue sustentar os braços. O *e-motion* se inscreve nessa lógica complementar.

2.3. Posicionamento do *e-motion* frente ao Estado da Arte

A Tabela 1 resume a comparação entre o *e-motion* e cinco abordagens citadas, considerando o sinal usado, o hardware exigido, o custo aproximado de aquisição (sem contar o computador), a precisão reportada na literatura original e a carga física típica. As precisões correspondem aos contextos originais de cada estudo e não são diretamente comparáveis entre si; servem para localizar a ordem de grandeza de cada solução.

Tabela 1. Comparação entre o *e-motion* e abordagens correlatas.

Abordagem	Sinal	Hardware	Custo	Precisão	Carga física
Gamepad/teclado padrão	Toque e preensão	Controle ou teclado	Baixo	Alta	Preensão fina, movimentos rápidos dos dedos
Xbox Adaptive Controller	Toque adaptado	Console + <i>switches</i> externos	Médio-alto	Alta	Apertar botões grandes ou <i>switches</i>
CNN para sinais [Sharma and Singh 2021]	Forma da mão	Câmera RGB	Baixo	99,96% ISL	Sustentar poses dos dedos
Camera Mouse [Betke et al. 2002]	Pontos da face	Câmera RGB	Baixo	Não informada	Movimento contínuo da cabeça
Eye tracking em jogos [Vickers et al. 2013]	Olhar	Eye tracker dedicado	Médio-alto	Não informada	Fixação visual sustentada
Controle por face [Taheri et al. 2021]	Expressões faciais	Câmera RGB	Baixo	Não informada	Mímica facial sustentada
e-motion (este trabalho)	Posição das mãos	Webcam comum + apoio para os antebraços	Baixo	>61 FPS, sem descarte médio	Translação dos antebraços, sem poses finas

O *e-motion* se diferencia em três pontos. O sinal de entrada é a posição cartesiana das mãos em uma grade calibrada, e não a forma da mão, o rosto ou o olhar. O sistema usa apenas uma webcam padrão e bibliotecas abertas (MediaPipe, OpenCV, PyAutoGUI), sem hardware dedicado ou *switches* adicionais. E o apoio dos antebraços sobre a mesa reduz a necessidade de sustentar os membros suspensos durante a sessão de jogo.

Para tornar o efeito sobre o custo concreto, a Tabela 2 compara o investimento mínimo de entrada de soluções que o *e-motion* pode complementar ou substituir parcialmente, com valores coletados em maio de 2026 em varejistas brasileiros (loja oficial PlayStation, Logitech Store BR, Microsoft, Mercado Livre, Amazon BR) e fornecedores especializados (Miotec, Mercado Adaptado, Civiam).

Tabela 2. Custo de entrada estimado no Brasil (maio de 2026) para soluções que o *e-motion* pode complementar ou substituir.

Solução	Hardware obrigatório	Custo (R\$)	Adaptação oferecida
PlayStation Access (Sony)	PS5 Digital + Access Controller + 2 <i>switches</i> externos	4.300–5.560	Botões grandes intercambiáveis, perfis personalizáveis
Xbox Adaptive + Logitech G Adaptive Kit	Xbox Series S + XAC + Kit Logitech (12 <i>switches</i>); ou PC + XAC + Kit (sem console)	5.200 com console; 2.000 só PC	<i>Switches</i> de baixa força, motricidade fina reduzida
Miotec <i>New Miotool</i> + Biogames	PC dedicado + sensor EMG + eletrodos descartáveis + <i>software</i> Biogames	6.000–9.000	EMG de superfície; jogos terapêuticos proprietários, não comerciais
e-motion (este trabalho)	PC comum + webcam (R\$ 50–300, frequentemente já disponível)	0–300 marginal sobre PC existente	Translação dos antebraços, sem preensão fina

Hardware obrigatório: equipamentos sem os quais a solução não funciona, ao menor preço oficial encontrado.

Custo: soma do hardware obrigatório (R\$), sem jogos, assinaturas ou serviços recorrentes.

Conversão: valores importados em R\$ 5,2/USD com impostos típicos quando aplicáveis.

A diferença de ordem de grandeza acontece porque soluções baseadas em consoles dedicados partem de R\$ 4.000 só para começar a jogar, equipamentos clínicos como o *New Miotool* ultrapassam R\$ 6.000 e cobrem apenas jogos terapêuticos próprios, enquanto o *e-motion* parte do hardware que o usuário já possui. Entretanto, o sistema exige que o usuário mantenha pelo menos uma das mãos no campo de visão e mova os antebraços sobre uma superfície de apoio, não atendendo tetraplegia alta, movimentos involuntários amplos ou dependência exclusiva de controle por olhar. Para esses perfis,

eye trackers, controladores faciais e o Xbox Adaptive Controller seguem sendo respostas mais adequadas. A proposta é complementar, não substitutiva.

3. Metodologia

O *e-motion* foi implementado em Python com OpenCV para manipulação de imagem, MediaPipe Hands para rastreamento e PyAutoGUI para envio de comandos ao sistema operacional. O desenvolvimento partiu de uma versão com gestos manuais e emulação de controle físico, mas essa abordagem adicionava dependências e exigia poses de mão, como mão aberta ou fechada. A versão final substituiu esses gestos por mapeamento posicional em zonas virtuais.

Essa mudança foi motivada por uma decisão de acessibilidade. Gestos baseados no formato da mão são expressivos, mas podem exigir força, extensão dos dedos e estabilidade articular. Já o deslocamento dos antebraços sobre uma superfície de apoio usa movimentos mais amplos. Ainda há esforço físico, e ele precisa ser medido em estudos futuros, mas a interação deixa de depender de apertar botões pequenos ou manter poses finas.

Para garantir a reprodutibilidade deste trabalho e incentivar o desenvolvimento contínuo da ferramenta, o código-fonte completo do *e-motion* foi disponibilizado publicamente no repositório do projeto no GitHub¹.

3.1. Arquitetura e Calibração

O fluxo do sistema possui quatro etapas. Primeiro, a câmera captura o frame e aplica os ajustes de orientação necessários. Em seguida, o usuário ou um assistente delimita uma região de interesse (ROI) pela seleção de dois pontos na imagem. Depois, o MediaPipe Hands detecta os landmarks das mãos dentro da ROI. Por fim, a posição das mãos é convertida em comandos discretos de teclado.

Na inicialização, o usuário ou um assistente delimita uma região de interesse (ROI) pela seleção de dois pontos na imagem. A ROI é dividida em três colunas com proporções 0,27, 0,44 e 0,27. As colunas laterais são divididas verticalmente em duas zonas; a coluna central usa divisão 40%/60%, deixando a região inferior como área de repouso. Esses valores foram definidos por ajuste manual em testes piloto e não devem ser interpretados como ótimos. A Figura 1 ilustra a grade resultante e o conjunto completo de combinações entre as mãos esquerda (Z_E) e direita (Z_D) que disparam cada comando do GBA.

¹Acesse em: <https://github.com/ricardospgc/emotion-controle-virtual-via-visao-computacional>

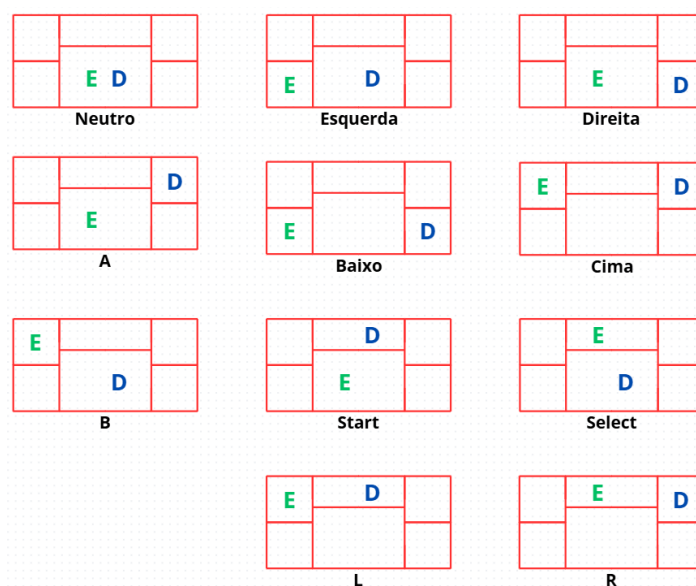


Figura 1. Combinações entre a posição da mão esquerda (E) e da mão direita (D) sobre a grade calibrada e os comandos correspondentes do GBA. A configuração “Neutro” deixa o sistema em repouso e não envia eventos ao emulador.

3.2. Rastreamento e Atuação

O sistema rastreia o *Landmark 9* do MediaPipe, próximo ao centro visual da mão, e determina a zona ativa da mão esquerda (Z_E) e da direita (Z_D). A combinação das duas zonas define o comando enviado ao jogo, permitindo acionar direcionais, A, B, L, R, Start e Select sem exigir movimentos finos dos dedos. Para reduzir disparos repetidos, o protótipo usa atraso de mudança de estado de 0,45s, cooldown de combo de 0,60s e pressões discretas de tecla com duração de 0,1s. O ambiente de demonstração foi o jogo *Pokémon Emerald*, executado em emulador de *Game Boy Advance*, por ser baseado em turnos e usar poucos botões simultâneos.

O mapeamento bimanual foi escolhido para ampliar o número de comandos sem aumentar excessivamente a área de alcance. Um comando pode depender, por exemplo, de a mão esquerda estar em uma zona lateral e a mão direita em uma zona superior. Essa combinação permite representar botões do GBA sem exigir que uma única mão percorra toda a ROI. A versão implementada permite ajustar a tabela de comandos conforme o jogo-alvo, mas o estudo manteve o foco no conjunto de entradas do GBA. A zona de repouso é importante para reduzir comandos involuntários: quando as mãos permanecem na região neutra, o sistema não envia eventos ao emulador, criando um estado previsível para pausas durante o jogo. O protótipo também permite que o usuário mantenha os antebraços apoiados sobre a mesa, o que reduz a necessidade de suspensão contínua dos membros superiores.

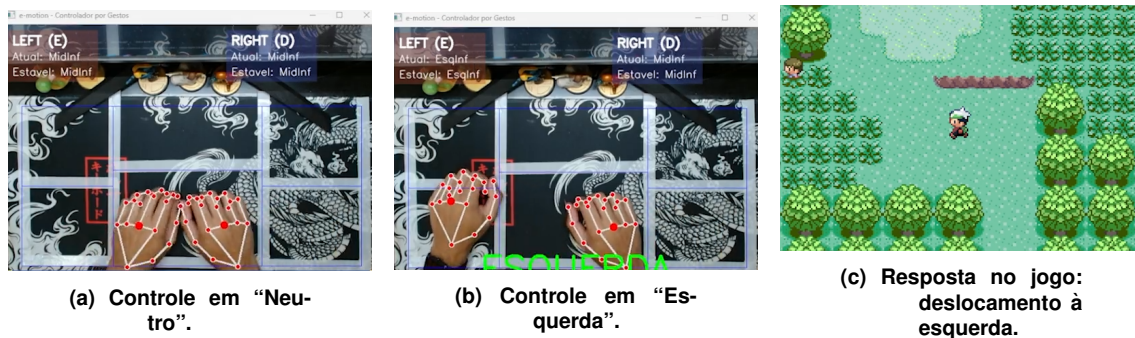


Figura 2. Demonstração do protótipo em *Pokémon Emerald*. Em (a), as mãos esquerda (E) e direita (D) estão na zona inferior central e o sistema permanece sem enviar comandos. Em (b), a mão esquerda desce para a zona inferior lateral, configurando o comando "Esquerda". Em (c), o personagem responde no emulador, deslocando-se um *tile* à esquerda.

4. Avaliação Preliminar e Discussão

A avaliação tem caráter preliminar. Foram consolidadas 10 execuções de benchmark geradas por quatro integrantes do grupo, todas com entrada *local_webcam*, isto é, frames capturados de câmera real. As métricas analisadas foram tempo total, FPS médio, speedup, eficiência, latência média por worker, custo da etapa de detecção e frames descartados. Os artefatos disponíveis não registraram, de forma padronizada, modelo de CPU, câmera, resolução, tempo de calibração ou falsos comandos; por isso, esses itens são tratados como limitação do estudo.

Como os testes foram executados em máquinas e câmeras diferentes, as médias devem ser interpretadas como caracterização exploratória do grupo, não como comparação controlada de hardware. Na escalabilidade forte, o número total de frames permaneceu fixo enquanto o número de workers variou. Na escalabilidade fraca, a carga cresceu junto com o número de workers. Linhas com tempo inferior a 0,05s foram descartadas por serem incompatíveis com o processamento de 100 ou mais frames.

4.1. Protocolo e Métricas

Cada execução produziu resultados para dois tipos de escalabilidade e para diferentes perfis do pipeline. Na escalabilidade forte, a carga total de frames foi mantida fixa, permitindo observar se o aumento do número de workers reduz o tempo total. Na escalabilidade fraca, a quantidade de trabalho cresce junto com o número de workers, permitindo observar o custo de coordenação quando o volume processado aumenta. Além disso, os perfis *detect*, *detect_zones*, *detect_zones_gestures* e *full* isolaram partes do pipeline para identificar onde estava o maior custo.

O critério de sucesso adotado foi simples: o pipeline completo deveria manter média acima de 30 FPS e não apresentar descarte médio de frames nas execuções agregadas. Esse limiar foi escolhido por ser compatível com jogos em turnos, nos quais o usuário decide comandos discretos e não precisa responder a eventos em frações muito pequenas de segundo. Dois pontos qualitativos observados durante o uso, ainda sem medição padronizada, aparecem como prioridade para a próxima rodada: a necessidade de uma calibração curta antes do jogo e a sensibilidade do rastreamento à iluminação ambiente.

Tabela 3. Resultados consolidados do pipeline completo nos experimentos de escalabilidade forte e fraca.

Exp.	Workers	N	Tempo (s)	FPS	Speedup/Ef.	Lat. (ms)
Forte	1	10	2,08 ± 0,83	61,22	1,00 / 1,00	19,43
Forte	2	10	1,35 ± 0,43	82,84	1,55 / 0,77	20,71
Forte	4	10	1,12 ± 0,43	106,89	2,00 / 0,50	26,02
Fraca	1	9	2,27 ± 0,64	47,62	1,00 / 1,00	19,70
Fraca	2	10	2,52 ± 1,10	96,05	0,81 / 0,40	22,72
Fraca	4	10	3,35 ± 1,53	147,66	0,62 / 0,16	29,18

Exp.: forte = carga total fixa; fraca = carga proporcional ao número de *workers*.
N: execuções válidas agregadas. **Tempo:** média ± desvio padrão.
Speedup: T_1/T_p . **Eficiência:** $speedup / workers$.
Lat.: latência média por frame (ms). Maior FPS e menor latência são melhores.

Na escalabilidade forte, o pipeline completo atingiu 61,22 FPS com 1 worker e 106,89 FPS com 4 workers, sem descarte médio de frames nas rodadas agregadas. A latência média por worker variou entre 19,43 ms e 26,02 ms. Nos perfis de pipeline com 4 workers, o perfil de detecção isolada obteve 108,18 FPS, enquanto o pipeline completo obteve 106,93 FPS; a etapa de detecção concentrou cerca de 25 ms por frame. A variação entre participantes foi relevante: no cenário forte com 4 workers, as médias variaram de 63,92 FPS a 180,92 FPS, reforçando o efeito do ambiente de execução.

Tabela 4. Perfis do pipeline com 4 workers, isolando o custo de cada etapa.

Perfil	N	Tempo (s)	FPS	Lat. (ms)	Detecção (ms)
detect	10	1,09 ± 0,41	108,18	25,33	24,57
detect_zones	10	1,12 ± 0,43	106,11	25,99	25,19
detect_zones_gestures	10	1,17 ± 0,48	104,87	27,00	26,17
full	10	1,14 ± 0,48	106,93	26,52	25,70

detect: apenas detecção MediaPipe.
detect_zones: adiciona mapeamento da posição em zonas.
detect_zones_gestures: adiciona a lógica bimanual.
full: inclui o envio de comandos via PyAutoGUI.
Detecção: tempo médio na chamada do MediaPipe; a diferença entre *detect* e *full* mede o *overhead* da lógica acima do detector.

A Tabela 4 indica que a etapa de detecção de mãos domina o custo do protótipo. A diferença entre o perfil de detecção isolada e o pipeline completo foi pequena, sugerindo que o cálculo das zonas e a lógica de comando acrescentam pouco overhead em comparação com o MediaPipe. Isso é importante para a evolução do sistema: melhorias de desempenho devem priorizar resolução de entrada, frequência de inferência, reaproveitamento de landmarks e configuração do detector.

Tabela 5. Variação por participante na escalabilidade forte com 4 workers.

Participante	N	Tempo (s)	FPS
André	3	0,55	180,92
Douglas	3	1,56	63,92
João	1	1,08	92,63
Ricardo	3	1,25	80,56

N: execuções válidas registradas por aquele participante.

Ambiente: cada participante rodou em sua própria máquina (CPU, câmera e iluminação distintas).

Os valores não são comparáveis como referência de hardware, mas evidenciam a faixa de desempenho na amostra do grupo.

Os resultados por participante reforçam que os números não devem ser lidos como benchmark controlado de hardware. Mesmo assim, todos os valores médios ficaram acima do limiar de 30 FPS. O resultado mais baixo no cenário forte com 4 workers foi 63,92 FPS, ainda compatível com a proposta de uso em RPG por turnos. Essa margem sugere que a abordagem tem folga para executar a lógica de zonas e envio de comandos em paralelo à detecção.

A observação funcional foi feita em cenário simulado com usuários sem diagnóstico de limitação motora. Os participantes restringiram punhos e dedos e usaram principalmente a translação dos antebraços para navegar entre zonas. A lógica posicional foi mais simples de manter que gestos manuais complexos, pois não exigia sustentar poses dos dedos. A observação não substitui testes com o público-alvo: o estudo não mediu esforço muscular, fadiga, conforto físico ou taxa formal de falsos comandos. Durante a demonstração foi observado que demarcar fisicamente os limites das zonas na mesa, com fita ou marcador tátil, reduz a dependência do overlay da câmera e pode acelerar a aprendizagem do espaço de comandos, hipótese a ser testada com participantes do público-alvo.

4.2. Discussão

Os resultados indicam que o gargalo principal não está na lógica de comando, mas na detecção das mãos. Isso torna a arquitetura modular uma vantagem prática, já que melhorias futuras podem trocar parâmetros de câmera, reduzir a frequência de inferência ou ajustar a resolução sem alterar a tabela de comandos. Também sugere que a interação por zonas é computacionalmente barata em comparação com a etapa de visão computacional.

A escolha por jogos de turnos deve ser interpretada como parte do desenho experimental, não como limitação acidental. Em jogos com combate em tempo real, mira contínua ou necessidade de pressionar vários botões em sequência, o mesmo resultado de FPS pode não ser suficiente. Em um RPG por turnos, por outro lado, o sistema precisa principalmente selecionar menus, confirmar ações e navegar pelo mapa. Essa diferença justifica o critério de sucesso usado no estudo.

Outro ponto é que a ausência de frames descartados nas médias agregadas não significa ausência de erros de interação. Um frame pode ser processado corretamente e ainda assim gerar um comando indesejado se a mão cruzar uma borda de zona ou se o usuário estiver em fase de aprendizagem. Por isso, taxa de falsos comandos, tempo de calibração e esforço percebido foram mantidos como métricas obrigatórias para trabalhos futuros.

4.3. Ameaças à Validade

- **Validade interna:** Os benchmarks usaram máquinas e câmeras diferentes, e parte dos runs foi gerada por versões anteriores do protótipo. A agregação removeu execuções incompatíveis com o volume esperado de frames, mas não corrige diferenças de CPU, iluminação, câmera ou resolução.
- **Validade externa:** O sistema foi avaliado em *Pokémon Emerald*, um jogo em turnos; os resultados não se generalizam para jogos de ação, mira contínua ou múltiplos comandos simultâneos.
- **Validade de construto:** FPS, latência e observação qualitativa aproximam fluidez e usabilidade, mas não medem conforto físico, esforço percebido ou eficácia clínica. Trabalhos futuros devem incluir voluntários do público-alvo, protocolo controlado de tarefas, tempo de calibração, falsos comandos, fadiga percebida e comparação com outras formas de controle adaptativo.

5. Conclusão

Este trabalho apresentou o *e-motion*, um sistema de controle virtual por visão computacional voltado a jogadores com mobilidade parcial dos membros superiores. A proposta desloca a interação do formato das mãos para sua posição em uma grade calibrada, evitando a exigência de prensão fina e poses sustentadas dos dedos que persiste tanto em controladores físicos convencionais quanto em sistemas que classificam gestos manuais. Ao operar sobre hardware comum (webcam e PC), o projeto se posiciona como alternativa de baixo custo ao ecossistema atual de acessibilidade em jogos, em que soluções consolidadas partem da casa dos milhares de reais em equipamentos dedicados.

A principal contribuição é metodológica: a documentação de um caminho de projeto que combina uma biblioteca aberta de detecção de mãos com um modelo bimanual de zonas, reproduzível em qualquer ambiente que disponha de câmera e processamento convencional. A avaliação preliminar mostrou que a arquitetura é computacionalmente viável em equipamentos comuns e que o gargalo principal está na detecção, não na lógica de comando, abrindo espaço para evoluir a interação sem custo adicional de hardware.

O estudo não substitui validação clínica, e o recorte cobre apenas parte dos perfis motores que se beneficiariam de interfaces sem contato. Os próximos passos envolvem ensaios com pessoas do público-alvo, medição de esforço percebido e fadiga, avaliação em jogos com demandas distintas das de um RPG em turnos e investigação de adaptações para usuários com perfis motores não atendidos pelo recorte atual, ampliando o alcance da contribuição para além da configuração explorada aqui.

Referências

- Armanini, K. K., Weber, F. M., Muraro, C. F., Borges Junior, N. G., Domenech, S. C., and Gevaerd, M. d. S. (2015). Evaluation of manual dexterity in individuals with rheumatoid arthritis. *Acta Fisiatrica*. doi:10.5935/0104-7795.20150032.
- Betke, M., Gips, J., and Fleming, P. (2002). The camera mouse: Visual tracking of body features to provide computer access for people with severe disabilities. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 10(1):1–10. doi:10.1109/TNSRE.2002.1021581.

- Brown, M. and Anderson, S. L. (2021). Designing for disability: Evaluating the state of accessibility design in video games. *Games and Culture*, 16(6):702–718. First published online 2020; doi:10.1177/1555412020971500.
- de Carvalho, C. V. and Coelho, A. (2022). Game-based learning, gamification in education and serious games. *Computers*, 11(3):36. doi:10.3390/computers11030036.
- Dewhirst, A., Laugharne, R., and Shankar, R. (2022). Therapeutic use of serious games in mental health: Scoping review. *BJPsych Open*, 8(2):e37. doi:10.1192/bjo.2022.4.
- Granic, I., Lobel, A., and Engels, R. C. M. E. (2014). The benefits of playing video games. *American Psychologist*, 69(1):66–78. doi:10.1037/a0034857.
- Gu, F., Fan, J., Cai, C., Wang, Z., Liu, X., Yang, J., and Zhu, Q. (2022). Automatic detection of abnormal hand gestures in patients with radial, ulnar, or median nerve injury using hand pose estimation. *Frontiers in Neurology*, 13:1052505. doi:10.3389/fneur.2022.1052505.
- Instituto Brasileiro de Geografia e Estatística (2023). Censo demográfico 2022: Pessoas com deficiência. IBGE.
- Sharma, S. A. and Singh, S. (2021). Vision-based hand gesture recognition using deep learning for the interpretation of sign language. *Expert Systems with Applications*, 182:115657. doi:10.1016/j.eswa.2021.115657.
- Taheri, A., Weissman, Z., and Sra, M. (2021). Design and evaluation of a hands-free video game controller for individuals with motor impairments. *Frontiers in Computer Science*, 3:751455. doi:10.3389/fcomp.2021.751455.
- Vickers, S., Istance, H., and Hyrskykari, A. (2013). Performing locomotion tasks in immersive computer games with an adapted eye-tracking interface. *ACM Transactions on Accessible Computing*, 5(1):2:1–2:33. doi:10.1145/2514856.
- Vieira, C., Ferreira da Silva Pais-Vieira, C., Novais, J., and Perrotta, A. (2021). Serious game design and clinical improvement in physical rehabilitation: Systematic review. *JMIR Serious Games*, 9(3):e20066. doi:10.2196/20066.
- Yuan, B., Folmer, E., and Harris Jr., F. C. (2011). Game accessibility: A survey. *Universal Access in the Information Society*, 10(1):81–100. doi:10.1007/s10209-010-0189-5.
- Zhang, F., Bazarevsky, V., Vakunov, A., Tkachenka, A., Sung, G., Chang, C.-L., and Grundmann, M. (2020). Mediapipe hands: On-device real-time hand tracking. In *CVPR Workshop on Computer Vision for Augmented and Virtual Reality*. arXiv:2006.10214.