

REVISÃO SOBRE OTIMIZAÇÕES DE LAYOUTS DE TECLADOS VIRTUAIS APLICADOS A TECNOLOGIA ASSISTIVA

Luiz Fernando Batista Loja
Instituto Federal de Goiás
luiz@doutorado.ufu.br

Renato de Sousa Gomide
Universidade Federal de Uberlândia
renato@doutorado.ufu.br

Edna Lucia Flôres
Universidade Federal de Uberlândia
edna@ufu.br

Rodrigo Pinto Lemos
Universidade Federal de Goiás
lemos@emc.ufg.br

Daniel Vitor Lucena
Universidade Federal de Goiás
daniel.lucena@ifg.edu.br

Resumo—Pessoas que possuem a Síndrome do Encarceramento são privadas de todos os seus movimentos e da capacidade de fala. Entre os vários métodos e softwares que possibilitam a comunicação desse tipo de paciente destaca-se o teclado virtual. Entretanto, sabe-se que a performance de digitação dos teclados virtuais é baixa. Uma maneira de aprimorar a digitação por meio desses teclados é otimizando seus layouts. Vários estudos vem sendo desenvolvidos com a finalidade de otimizar esses teclados e torná-los mais eficientes. O objetivo desse trabalho é mostrar as pesquisas desenvolvidas para aprimorar o teclado virtual. Foram identificados diversos métodos que podem ser utilizados para a resolução parcial desse problema. A partir desse levantamento uma nova metodologia de otimização é proposta.

Palavras-Chaves - Comunicação Aumentativa e Alternativa, Otimização, Teclado Virtual, Tecnologia Assistiva, Revisão

REVIEW OF OPTIMIZATIONS VIRTUAL KEYBOARDS LAYOUTS APPLIED TO ASSISTIVE TECHNOLOGY

Resumo—People who have the Locked in Syndrome suffer of lack of movements and speech capabilities. Among the various methods and softwares that enable this type of patient to communicate is the virtual keyboard. However, it is known that the typing performance of virtual keyboards is low. One way to improve typing speed through keyboards is optimizing their layouts. Several studies have been developed in order to optimize these softwares and make them more efficient. The aim of this study is to show the researchs developed to improve the virtual keyboard. It was identified that many methods can be used to solve this problem partially. Based on this survey a new optimization methodology is proposed.

Key-Words - Augmentative and Alternative Communication, Optimization, Virtual Keyboard, Assistive Technology, Review

I. INTRODUÇÃO

Pessoas com restrições motoras e de fala ficam com a comunicação verbal e a linguagem corporal prejudicadas. Nos casos mais extremos o paciente é privado de todos os seus movimentos e da capacidade de fala. Essa situação é caracterizada como a síndrome do encarceramento (SE). Nesses casos o paciente deve recorrer a uma tecnologia que permita sua comunicação com o ambiente e as pessoas que o cercam. A Comunicação Aumentativa e Alternativa (CAA) reúne os métodos e as tecnologias desenhadas para auxiliar ou substituir a comunicação oral dessas pessoas [1].

Segundo [2], devido à crescente popularização dos computadores e à expansão da internet, vários trabalhos são desenvolvidos para auxiliar na comunicação de pacientes com afasia e tetraplegia. Os softwares computacionais implementados para auxiliar a comunicação dos indivíduos que possuem tais características são chamados de sistemas de CAA. Esses programas são diversificados e incluem desde teclados virtuais [3] até complexas planilhas de comunicação [4].

As planilhas de comunicação proporcionam ao usuário a capacidade de se expressar de maneira rápida. Essa funcionalidade se deve ao fato de que esses programas possuem comandos predefinidos. Esses comandos podem ser selecionados diretamente pelo usuário e permitem a eles expressarem seus sentimentos, desejos ou necessidades. Apesar da eficiência das planilhas de comunicação os usuários desse tipo de software necessitam de meios para se comunicarem de forma detalhada, formal e específica. Nesse caso os teclados virtuais ou soft keyboards representam uma alternativa para essa necessidade.

O teclado virtual é uma das alternativas mais primitivas de mecanismos de entrada de texto. Esse tipo de software substitui o teclado físico por uma representação gráfica do teclado na tela do computador [5]. Entretanto, em comparação ao seu correlato físico sabe-se que esse tipo de solução possui uma baixa performance de digitação [6], [7].

Essa performance é ainda pior quando os teclados virtuais



XIII CEEL - ISSN 2178-8308
12 a 16 de Outubro de 2015
Universidade Federal de Uberlândia - UFU
Uberlândia - Minas Gerais - Brasil

são utilizados por pessoas portadoras de SE. Pois, esses usuários estão aptos a transmitirem apenas um único tipo de estímulo, essa restrição faz com que esse tipo de usuário se comunique de maneira objetiva (*ativado, desativado*). Uma das estratégias utilizadas para possibilitar essa comunicação é a técnica de varredura.

Essa técnica consiste em percorrer o teclado virtual destacando suas teclas em sequência. Assim que a tecla desejada receber destaque o usuário transmite o estímulo. Esse estímulo é identificado pelo software e o programa seleciona a tecla ou opção em destaque [8]. Segundo [9], [8], esse tipo de método de interação limita a performance de escrita do usuário. Para maximizar a quantidade de palavras por minuto escritas por esses usuários é necessário que o teclado virtual seja otimizado.

O objetivo dessa otimização é identificar a *layout* ideal que aumente a performance de digitação e diminua o esforço do usuário durante a entrada de texto. Segundo [10] o problema do arranjo do teclado pode ser reduzido polinomialmente ao problema de atribuição quadrática, que é NP-completo [11]. Portanto, a solução desse problema não pode ser realizada computacionalmente em um tempo polinomial.

Vários trabalhos foram desenvolvidos na tentativa de obter um *layout* de teclado otimizado que aumente a performance de digitação e diminua o esforço do usuário. Este trabalho é uma revisão sobre os esforços de aprimoramento do *layout* do teclado virtual. A finalidade dessa revisão é identificar as possíveis abordagens que podem ser utilizadas para produzir um teclado virtual assistivo com *layout* otimizado.

Esse trabalho está estruturado da seguinte forma: a Seção II descreve uma breve história sobre os *layouts* e faz uma classificação dos teclados existentes. O problema de otimização do teclado é apresentado na seção III. Na Seção II são descritas as pesquisas relacionadas a otimização dos *layouts*. A análise e os resultados dessa pesquisa são descritos na Seção V. Finalmente, a Seção VI descreve a conclusão desse trabalho e sugere trabalhos futuros.

II. TECLADOS E *Layouts*

O teclado QWERTY foi desenvolvido por Christopher Sholes em 1873. O principal objetivo desse *layout* era evitar um problema mecânico que ocorria durante a datilografia em máquinas de escrever [12], [13], [14]. Portanto, a estrutura do teclado QWERTY não foi elaborada com a finalidade de otimizar a performance de digitação, mas sim diminuir a velocidade com que as teclas eram pressionadas.

Após a construção da primeira máquina de escrever digital e posteriormente dos computadores, os problemas mecânicos se tornaram irrelevantes para a utilização dos teclados. Segundo [10], [14], a primeira tentativa de aprimoramento do teclado QWERTY foi realizada por Dvorak em 1986. O objetivo do novo *layout* era aumentar a performance de digitação. Para atingir essa finalidade Dvorak construiu um teclado considerando a frequência de ocorrência das letras

no idioma inglês. Caracteres que possuíam maior utilização foram redistribuídos entre as teclas mais fáceis de serem acessadas.

Apesar dos avanços conseguidos por Dvorak o teclado QWERTY permaneceu como o *layout* padrão para os teclados de computadores [15], [13], [16]. Esse fato pode ser parcialmente explicado pela inconveniência da alteração do padrão de teclado [16].

Porém, a evolução da tecnologia trouxe novos tipos de dispositivos como pdas, telefones celulares, tablets e relógios digitais. O *layout* QWERTY se mostrou ineficiente para as novas tecnologias, pois esse tipo de *layout* era grande demais para ser adaptado aos novos dispositivos. Os *layouts* podem ser classificados em quatro categorias, um único caractere (TUC), múltiplos caracteres (TMC), único dedo (TUD), multi dedos (TMD) [10].

Os teclados que possuem apenas uma letra por tecla são classificados como TUC. O QWERTY e o teclado proposto por Dvorak são exemplos desse tipo de categoria. Estes teclados são utilizados na entrada de dados em computadores e notebooks.

Os teclados do tipo TMCs são usados em aparelhos que possuem uma área de trabalho menor como telefones celulares e pdas. Por apresentar uma área de trabalho pequena os TMCs possuem a quantidade de teclas menor do que a quantidade de caracteres. Assim, em uma tecla pode existir mais de um caractere. Essa característica torna este tipo de teclado ambíguo, pois ao pressionar uma tecla o sistema não pode determinar o caractere desejado. A questão da ambiguidade das letras entre as teclas pode ser resolvida de duas maneiras, a nível de letra ou de palavra.

O método de desambiguidade no nível de letra consiste em escolher a letra desejada logo após a escolha da tecla. Portanto, após o usuário selecionar uma das teclas a letra relativa a tecla escolhida deve ser especificada. [17] apresentam alguns métodos de desambiguidade no nível de letra.

No nível de palavra esse processo é realizado assim que são escolhidas todas as teclas que formam a palavra. A partir da combinação dessas teclas o método de desambiguidade apresenta uma lista de palavras existentes a partir da sequência selecionada. Essa lista se baseia em um dicionário pré-definido. O algoritmo de desambiguidade mais popular é o T9 [18]. A princípio esse algoritmo foi construído para auxiliar pessoas com deficiência de digitação. Porém, ele se popularizou ao ser utilizado por diversas marcas de telefones celulares no mundo inteiro.

Com a popularização dos dispositivos móveis as pesquisas sobre teclados se voltaram para soluções que pudessem ser utilizadas com apenas uma mão [19]. Os teclados do tipo TUD surgiram a partir dessas pesquisas. Nesse tipo de abordagem o usuário usa uma das mãos para segurar o aparelho e a outra para selecionar as teclas. Para desenvolver teclados desse tipo é necessário considerar três características, são elas: o tempo de movimento entre as teclas desejadas, a

freqüência de transição entre os caracteres e o formato do teclado [10].

Finalmente, os TMD são teclados desenvolvidos para a digitação utilizando mais de um dedo. Geralmente, esses teclados são utilizados em dispositivos que possuem uma tela maior para disponibilizar o teclado como nos tablets. O objetivo desse tipo de teclado é reduzir o risco de problemas musculares e melhorar a postura durante os longos processos de digitação [10].

Devido ao surgimento dessas novas tecnologias foi necessário construir teclados otimizados para cada tipo de dispositivo. Além disso, com a possibilidade da construção dos teclados virtuais a criação de alternativas de teclados pode ser realizada com baixo custo. Porém, para otimizar os teclados virtuais é necessário entender qual é o problema e quais são os objetivos a serem atingidos.

III. PROBLEMA DE OTIMIZAÇÃO DO TECLADO

O problema de otimização do arranjo do teclado se resume em distribuir uma quantidade de caracteres M em um conjunto de N teclas e identificar a combinação entre as teclas e as letras que otimizam determinados objetivos. Os critérios de otimização são variados e dependem principalmente da finalidade para qual o teclado é construído. De acordo com [10] esses objetivos são classificados em duas categorias: critérios ergonômicos e critérios de eficiência relativos a desambiguidade e a predição de texto.

Os critérios ergonômicos estão relacionados a minimizar o esforço de digitação relacionado a movimentação das mãos e dos dedos. A finalidade do critério de eficiência dos métodos de desambiguidade e predição é minimizar o número de teclas pressionadas para realizar o processo de desambiguidade e/ou de predição de palavras.

Como descrito anteriormente a distribuição de caracteres em várias teclas pertence a uma classe de problemas computacionais que não podem ser resolvidos de maneira trivial. Isso acontece porque para encontrar a melhor solução é necessário verificar todas as combinações possíveis. O número de possibilidades pode ser calculado pela fórmula proposta por [20] 1.

$$A_{total} = \frac{M!}{\prod_{i=1}^N (k_i!) \times \prod_{n=1}^{K_{max}} (c_n!)} \quad (1)$$

Onde:

N - quantidade de teclas;

M - número de caracteres;

c_n - número de teclas com exatamente n caracteres; e

K_{max} - número máximo de caracteres que podem ser agrupados em uma tecla.

Para realizar o cálculo da quantidade de possibilidades possíveis Leshner [20] considera 26 letras distribuídas entre 9 teclas. Analisando o arranjo com três caracteres em 8 teclas e dois caracteres em uma tecla é possível realizar 3×10^{15} combinações. Naturalmente, pesquisar todo espaço

de soluções pertencente a esse problema é impossível, mesmo para os mais avançados computadores. Embora, os problemas do tipo np-completo sejam considerados sem solução, existem métodos que conseguem encontrar soluções parcialmente ótimas para esse tipo de problema [13].

IV. REVISÃO

Levine (1990) propôs a primeira tentativa de otimizar o *layout* do teclado utilizando técnicas computacionais. Ele usou algoritmos genéticos [15]. Nesse artigo os pesquisadores tentaram otimizar os teclados do tipo TUC e TMC. As funções objetivo para os teclados do tipo TUC consideram o tempo de transição das teclas e a sua frequência de ocorrência na língua inglesa. Na otimização dos teclados TMC a função mérito utilizada minimiza a quantidade de colisões entre as palavras e o número de teclas pressionadas. Para os teclados TMC o índice de eficiência na desambiguidade das letras foi de 86% utilizando 10 teclas e 62% com 4 teclas.

Em uma tentativa de aprimorar o trabalho de [15] Oommen (1990) propôs uma nova forma de minimizar a ambiguidade do teclado. [21] utilizou a técnica de aprendizagem estocástica por autômatos. O autômato proposto por ele distribui as letras entre as teclas buscando reduzir o número de colisões entre as palavras. O processo de construção do teclado é iniciado por um *layout* aleatório. As palavras do dicionário são analisadas uma a uma de acordo com o teclado atual. A cada palavra analisada é verificado o número de colisões produzidos por essa palavra e as letras são permutadas de acordo com esse número. [21] conseguiu um resultado melhor do que [15].

Em 1993 Light apresentou um trabalho utilizando o algoritmo de Reconhecimento Simulado para otimizar os teclados TUC [13]. Nessa abordagem é usada uma única função objetivo. Essa função considera a frequência de utilização das letras no idioma inglês e o tempo de transição entre as teclas. Os teclados construídos com essa abordagem apresentam graus de eficiência de 8 e 3 por cento melhores do que as abordagens QWERTY e Dvorak. Os trabalhos de Levine [15] e Oommen [21] não foram citados por Light.

[20] apresentou um trabalho de otimização utilizando o método n-opt. Para calcular a eficiência dos teclados construídos pelo algoritmo de otimização, Leshner elaborou uma matriz de confusão. Essa matriz é construída a partir do corpus da linguagem e apresenta uma estrutura de dados composta bidimensional. Os valores da matriz representam o número total de vezes que o caractere β foi sugerido antes do caractere α quando α era o caractere desejado.

O algoritmo de otimização utilizado por Leshner usa a matriz de confusão para identificar os melhores arranjos do teclado. Para teclados com 9 teclas esse método foi eficiente chegando a atingir 90% de eficiência. Leshner comparou seu trabalho com o de Levine e obteve resultados ligeiramente melhores do que o seu correlato [15].

Em 2001 Zhai e seus pesquisadores avançaram com as pesquisas de teclados TUC utilizando o algoritmo Metrópolis para construir um novo *layout* [22]. O objetivo de Zhai era distribuir as letras de modo a diminuir a movimentação das mãos e dos dedos de acordo com uma linguagem qualquer. Zhai fez uma nova análise da performance de digitação de alguns teclados como QWERTY, CHUBON, FITALY e OPTI. O teclado proposto em [22] superou a performance dos demais teclados em até 10% e do teclado QWERTY em 43%.

[16] utilizou a metaheurística da colônia de formigas para otimizar o teclado levando em consideração apenas os aspectos ergonômicos. Este trabalho foi desenvolvido no ano de 2003 e obteve um teclado diferenciado agrupando as consoantes de um lado do teclado e o resto dos caracteres como vogais, pontuação e caracteres especiais do outro lado. Apesar de ter encontrado um *layout* distinto de todas as outras abordagens, [16] não fez nenhum tipo de teste para mostrar a performance de digitação da sua proposta.

Em 2005 os algoritmos genéticos voltaram a ser utilizados para a otimização dos teclados virtuais [23], [14]. [23] utilizou essa técnica para aprimorar os teclados TUC. A otimização proposta por Raynal usou a mesma função objetivo de [22]. Assim, a principal finalidade era reduzir a movimentação das mãos e dos dedos na frequência dos dígrafos da linguagem. Os resultados obtidos por [23] foram ligeiramente maiores do que os resultados de [22].

[14] descreveu um método de otimização de teclados TUC baseado nos princípios da eficiência propostos por [24]. Cada princípio apresentado por Norman foi traduzido em uma função objetivo. Essas funções são minimizadas utilizando um algoritmo genético. Goettl conseguiu construir teclados de 40 a 20 por cento mais eficientes do que os teclados QWERTY e Dvorak, respectivamente.

No ano de 2007 Sorensen desenvolveu um trabalho com o objetivo aprimorar a digitação das mensagens pelo celular [12]. Para otimizar o TMC de nove teclas foi utilizado um método de busca randômica. Sorensen tentou minimizar o custo total de digitação de cada palavra e o número de colisões entre elas. A análise dos resultados desta otimização mostrou que as vogais não devem ser agrupadas na mesma tecla, alguns caracteres agrupados formam uma boa combinação para diminuir a ambiguidade e não existe tendências quantitativas na distribuição das letras por tecla.

[25] utilizou o algoritmo de subida da colina para redistribuir as letras de maneira ótima. Esse trabalho utilizou a abordagem TUC e o método de varredura linha e coluna. Além disso, ele utilizou duas funções objetivo. A primeira considerou o número de passos necessários para selecionar um determinado caractere. A segunda função penaliza determinadas estruturas de teclado, como por exemplo, estruturas que não mantêm a sequência de grupos como a de números, caracteres de pontuação e letras. O trabalho de Francis não relatou a eficiência da abordagem adotada.

O objetivo do trabalho proposto por Yin [10] foi aprimorar o arranjo dos teclados TMC utilizando o método de nuvem de partículas. Esse trabalho se destacou dos outros, pois considerou mais de uma função objetivo para a otimização. As propriedades otimizadas por Yin foram acessibilidade da tecla, conforto da postura de digitação, número de teclas pressionadas e colisões entre as palavras. Yin conseguiu bons resultados e comparou o seu trabalho com outras abordagens como teclado alfabético, a abordagem de Levine e a distribuição de frequência.

Finalmente, em 2012 Brouillette tentou aumentar a performance de digitação dos teclados do tipo TUC utilizando algoritmos genéticos. O trabalho de Brouillette também considerou o tempo de aprendizado para os novos usuários. Segundo [26] o teclado proposto possuía o dobro da performance de digitação do teclado QWERTY.

V. ANÁLISE E RESULTADOS

A tabela I classifica os trabalhos encontrados nesta revisão. Nessa tabela são apresentados o nome do método utilizado para minimizar a função objetivo, assim como os objetivos da minimização de cada trabalho. As duas últimas colunas indicam o tipo de teclado otimizado e se esse teclado foi idealizado como um software assistivo.

As características de otimização foram divididas em 5 tipos: acessibilidade das teclas, conforto da postura, número de teclas pressionadas, colisões entre as palavras e a capacidade de aprendizado.

Acessibilidade da tecla está relacionada a localização das teclas no *layout* do teclado. As teclas que possuem maior taxa de utilização devem ser mais fáceis de serem acessadas. Segundo [10], a sequência de digitação dos dedos pode causar lesões musculares e prejudicar a performance de digitação. O critério de postura analisa a alternância de utilização dos dedos entre as duas mãos.

O número de teclas pressionadas representa a quantidade de opções que devem ser selecionadas para inserir uma texto. Essa medida impacta diretamente na performance de digitação. Pois quanto maior é o número de interações entre o teclado e o usuário menor é a performance de digitação.

A colisão entre os termos ocorre quando uma mesma sequência de teclas origina mais de uma palavra. As colisões são comuns em teclados multi-caracteres ou ambíguos. Quanto menor é o número de colisões mais rápido o usuário seleciona a palavra desejada e maior é a sua performance de entrada de dados.

Finalmente, a capacidade de aprendizado mede a facilidade de aprender e utilizar o novo *layout* proposto. Quanto mais alto é esse valor mais fácil o usuário se adequa ao novo teclado e mais rápido ele consegue aumentar sua performance de digitação.

Nota-se na tabela I que apenas um trabalho considerou a capacidade de aprendizado do usuário [26]. Apesar desta

Trabalho	Algoritmo	Acessibilidade	Conforto	Teclas Presionadas	Colisões	Aprendizado	Tipo	Assistivo
[15]	Algoritmos Genéticos	X		X	X		TMC	Sim
[21]	Automato de Aprendizagem Estocástica				X		TMC	Sim
[13]	Recozimento Simulado	X					TUC	Não
[20]	N-opt			X			TMC	Sim
[22]	Metropolis	X					TUC	Não
[16]	Colônia de Formigas	X	X				TUC	Não
[14]	Algoritmos Genéticos	X	X				TUC	Não
[23]	Algoritmos Genéticos	X					TUC	Não
[12]	Busca Aleatória			X	X		TMC	Não
[25]	Subida a Colina			X			TUC	Sim
[10]	Nuvem de Partículas	X	X	X	X		TMC	Sim
[26]	Algoritmos Genéticos		X			X	TUC	Não

Tabela I
TABELA DE REVISÃO COM TODOS OS ARTIGOS PESQUISADOS

característica ser essencial para pessoas que estão iniciando a utilização de uma nova tecnologia.

Outro ponto que pode ser observado nos trabalhos futuros é o tipo de acesso as opções do teclado. Os usuários de teclados assistivos muitas vezes possuem restrições motoras graves como os portadores de SE. Nessas situações, deve ser aplicado ao teclado o método de varredura. Apenas o trabalho de [25] considerou o tipo de acesso indireto. Segundo [27], é essencial considerar no método de otimização a fase de seleção da tecla para a construção de teclados que usam métodos de varredura.

Apesar de otimizar um teclado para portadores de deficiência [25] escolheu uma abordagem TUC para modelar o teclado. De acordo com [28], [29], a abordagem TMC é melhor do que os teclados TUC. [30] afirmou que um dos principais objetivos do teclado ambíguo é reduzir o esforço de digitação. Além disso, teclados ambíguos minimizam a interação entre o usuário e o software [31].

O método mais utilizado para otimização do teclado foi o algoritmo genético. Bons resultados foram obtidos utilizando esta técnica. Além disso, esse método permitiu estabelecer multi-objetivos, ou seja, mesmo que as funções sejam antagonicas é possível otimizar as funções mérito. Uma boa abordagem é utilizar esse método em conjunto com o método n-opt. O algoritmo genético amplia o espaço de busca possibilitando encontrar os mínimos globais. As soluções encontradas por este método podem ser otimizadas utilizando o método n-opt.

O método n-opt foi escolhido por otimizar razoavelmente bem o problema do arranjo de teclas em teclado ambíguo. Além disso, o processo do n-opt permite evitar as combinações desnecessárias, como alteração entre as letras em uma mesma tecla. Isso permite que o algoritmo funcione com um número menor de possibilidades.

Outro ponto importante a ser destacado no processo de otimização do teclado assistivo para pacientes de SE é definir as funções objetivo que serão utilizadas. A princípio pretende-se usar uma função que calcule o esforço de digitação considerando o método de varredura linear. A

segunda função determinará o índice de colisões das palavras mais utilizadas pelo usuário. Os critérios ergométricos não serão considerados, pois quando o teclado utiliza um método de varredura esses critérios são irrelevantes.

VI. CONCLUSÃO

Nessa revisão foram analisados os artigos relacionados a otimização dos *layouts* dos teclados. Foram apresentados os tipos de teclados encontrados na literatura. Logo, foi apresentada uma tabela com a categorização de cada trabalho pesquisado e finalmente a metodologia de otimização que pode ser adotada nos novos trabalhos.

O principal objetivo dessa pesquisa era identificar os tipos de métodos utilizados para otimizar o *layout* dos teclados virtuais. Esse objetivo foi alcançado parcialmente, pois para ter exatidão nesse resultado uma revisão sistemática deve ser elaborada e executada. Esse processo de revisão está sendo elaborado e será realizado futuramente.

Após a finalização dessa revisão pretende-se aprimorar o método de otimização apresentado de acordo com os novos resultados e implementá-lo. Os teclados assistivos provenientes desse método deverão ser testados com portadores de deficiência motora grave.

AGRADECIMENTOS

Agradecemos ao programa ProApp regido pelo Instituto Federal de Goiás pelos recursos cedidos.

REFERÊNCIAS

- [1] K. M. Wilkinson and S. Hennig, "The state of research and practice in augmentative and alternative communication for children with developmental/intellectual disabilities," *Mental Retardation and Developmental Disabilities Research Reviews*, vol. 13, no. 1, pp. 58–69, 2007.
- [2] S.-W. Park, Y.-I. Yim, S.-h. Yi, H.-y. Kim, and S.-m. Jung, "Augmentative and alternative communication training using eye blink switch for locked-in syndrome patient," *Annals of Rehabilitation Medicine*, vol. 36, no. 2, pp. 268–272, 2012.

- [3] Y.-F. Fu and C.-S. Ho, "A fast text-based communication system for handicapped aphasiacs," in *Intelligent Information Hiding and Multimedia Signal Processing, 2009. IHH-MSP '09. Fifth International Conference on*, 2009, pp. 583–594.
- [4] P. Biswas and D. Samanta, "Friend: A communication aid for persons with disabilities," *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, vol. 16, no. 2, pp. 205–209, 2008.
- [5] S. Ghosh, "Designing an Efficient Virtual Keyboard for Text Composition in Bengali," no. c, pp. 84–87, 2011.
- [6] J. H. Kim, L. Aulck, M. C. Bartha, C. a. Harper, and P. W. Johnson, "Differences in typing forces, muscle activity, comfort, and typing performance among virtual, notebook, and desktop keyboards." *Applied ergonomics*, vol. 45, no. 6, pp. 1406–13, Nov. 2014.
- [7] S. Kwon, D. Lee, and M. K. Chung, "Effect of key size and activation area on the performance of a regional error correction method in a touch-screen QWERTY keyboard," *International Journal of Industrial Ergonomics*, vol. 39, no. 5, pp. 888–893, Sep. 2009.
- [8] P. Emiliani *et al.*, "A study of two-inputs scanning methods to enhance the communication rate," *Assistive Technology from Adapted Equipment to Inclusive Environments: AAATE 2009*, vol. 25, p. 132, 2009.
- [9] J. Miró-borrás, P. Bernabeu-soler, J. Miró-borrás, and P. Bernabeu-soler. "Text Entry in the E-Commerce Age : Two Proposals for the Severely Handicapped," vol. 4, no. 1, pp. 101–112, 2009.
- [10] P.-Y. Yin and E.-P. Su, "Cyber swarm optimization for general keyboard arrangement problem," *International Journal of Industrial Ergonomics*, vol. 41, no. 1, pp. 43–52, 2011.
- [11] S. P. Ladany, "A model for optimal design of keyboards," *Computers Operations Research*, vol. 2, no. 1, pp. 55 – 59, 1975.
- [12] K. Sörensen, "Multi-objective optimization of mobile phone keymaps for typing messages using a word list," *European Journal of Operational Research*, vol. 179, no. 3, pp. 838–846, 2007.
- [13] L. Light and P. Anderson, "Designing better keyboards via simulated annealing," 1993.
- [14] J. S. Goettl, A. W. Brugh, and B. A. Julstrom, "Call me e-mail: arranging the keyboard with a permutation-coded genetic algorithm," in *Proceedings of the 2005 ACM symposium on Applied computing*. ACM, 2005, pp. 947–951.
- [15] S. H. Levine and C. Goodenough-Trepagnier, "Customised text entry devices for motor-impaired users," *Applied ergonomics*, vol. 21, no. 1, pp. 55–62, 1990.
- [16] J. Eggers, D. Feillet, S. Kehl, M. O. Wagner, and B. Yannou, "Optimization of the keyboard arrangement problem using an ant colony algorithm," *European Journal of Operational Research*, vol. 148, no. 3, pp. 672–686, 2003.
- [17] M. Silfverberg, I. S. MacKenzie, and P. Korhonen, "Predicting text entry speed on mobile phones," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '00. New York, NY, USA: ACM, 2000, pp. 9–16.
- [18] D. Grover, M. King, and C. Kushler, "Reduced keyboard disambiguating computer," Oct. 6 1998, uS Patent 5,818,437.
- [19] E. Matias, I. S. MacKenzie, and W. Buxton, "One-handed touch typing on a qwerty keyboard," *Human-Computer Interaction*, vol. 11, no. 1, pp. 1–27, 1996.
- [20] G. W. Lesh, B. J. Moulton, and D. J. Higginbotham, "Optimal character arrangements for ambiguous keyboards," *Rehabilitation Engineering, IEEE Transactions on*, vol. 6, no. 4, pp. 415–423, 1998.
- [21] B. J. Oommen, R. S. Valiveti, and J. Zgierski, "A fast learning automaton solution to the keyboard optimization problem," in *Proceedings of the 3rd International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems - Volume 2*, ser. IEA/AIE '90. New York, NY, USA: ACM, 1990, pp. 981–990.
- [22] S. Zhai and B. A. Smith, "Alphabetically biased virtual keyboards are easier to use: layout does matter," in *CHI'01 extended abstracts on Human factors in computing systems*. ACM, 2001, pp. 321–322.
- [23] M. Raynal and N. Vigouroux, "Genetic algorithm to generate optimized soft keyboard," in *CHI'05 extended abstracts on Human factors in computing systems*. ACM, 2005, pp. 1729–1732.
- [24] D. Norman and D. Rumelhart, "Studies of typing from the Inr research group," in *Cognitive Aspects of Skilled Typewriting*, W. Cooper, Ed. Springer New York, 1983, pp. 45–65.
- [25] G. Francis and E. Johnson, "Speed accuracy tradeoffs in specialized keyboards," *Journal of Human Computer Studies*, vol. 69, no. 7-8, pp. 526–538, 2011.
- [26] A. Brouillette, U. Sharma, and J. Kalita, "Multi-objective optimization for efficient brahmic keyboards," in *24th International Conference on Computational Linguistics*, 2012.
- [27] J. Miró-Borrás, P. Bernabeu-Soler, R. Llinares, and J. Igual, "Ambiguous keyboards and scanning: The relevance of the cell selection phase," in *Human-Computer Interaction-INTERACT 2009*. Springer, 2009, pp. 1–4.
- [28] C. Topal and B. Benligiray, "On The Efficiency Issues of Virtual Keyboard Design," no. 111, 2012.
- [29] A. Molina, O. Rivera, I. Gómez, and G. Sánchez, "Evaluation of unambiguous virtual keyboards with character prediction," *Assistive Technology from Adapted Equipment to Inclusive Environments: AAATE 2009*, vol. 25, p. 144, 2009.
- [30] S. Bhattacharya and S. Laha, "Bengali text input interface design for mobile devices," *Universal Access in the Information Society*, vol. 12, no. 4, pp. 441–451, Aug. 2012.
- [31] Y. Guerrier, M. Baas, C. Kolski, and F. Poirier, "Comparative Study between AZERTY-type and K- Hermes virtual Keyboards dedicated to Users with Cerebral Palsy," 2011.