

RECONHECIMENTO ESPECIAL DE MARCADORES PARA MANUSEIO DE COMPUTADORES

Fábio Luís Picelli Luchini

Universidade de Campinas (Unicamp) – Faculdade de Engenharia Elétrica e
Computação (FEEC) – Departamento de Computação e Automação (DCA)

email: luchini@dca.fee.unicamp.br

Projeto Financiado pela FAPESP (Processo: 98/16433-1)

Resumo

Neste trabalho descrevo a implementação de um sistema computacional, denominado REMMC (Reconhecimento Especial de Marcadores para Manuseio de Computadores), para realizar a tarefa de detecção e reconhecimento de um objeto inserido dentro uma imagem. Esta imagem é recebida por uma câmera de vídeo webcam, e o sistema procura um objeto específico na figura com a finalidade de encontrar a sua posição e seguir o seu deslocamento dentro da sequência de imagens recebidas pela câmera. A movimentação desse objeto, chamado marcador, é transmitido ao sistema operacional para simular o movimento do mouse. Com essa alternativa é possível controlar o computador sem colocar a mão no teclado ou no mouse. Esse objeto marcador pode ser um adesivo ou colante colocada em qualquer parte do corpo do usuário.

Esse sistema poderá ser utilizado por pessoas com deficiências motoras que tenham dificuldades ou impossibilidade de usar o mouse ou teclado do computador, detectando o melhor movimento que o indivíduo seja capaz de realizar para controlar o computador.

Desta forma o sistema proposto é uma alternativa eficiente às soluções ad hoc baseadas no desenvolvimento de periféricos que são desenhados especificamente para cada deficiente. Essas soluções além de serem mais caras, dependem sempre que um técnico se disponibilize a construí-las.

O sistema se baseia no conhecimento fisiológico do sistema visual humano, utilizando técnicas de linguagens formais nebulosas e de visão computacional. A eficiência do sistema foi testado com adultos e crianças normais e também crianças portadoras de deficiências motoras leves e graves. Os resultados comprovam a possibilidade de utilização do sistema para controle de softwares, inclusive educacional, embora o treinamento requerido para tanto dependa da capacidade motora inicial do indivíduo. O sistema poderá, portanto, ser utilizado tanto para lazer como para facilitar o aprendizado, quer em sistemas locais ou em processo de educação à distância (Internet).

Introdução

A adoção da postura bípede foi a solução encontrada pela natureza para a adaptação dos antecessores do *Homo Sapiens* às novas condições de vida, quando eles abandonaram as florestas em troca das savanas. Esta mudança de postura, enquanto solução para os problemas mais imediatos de sobrevivência, criou outras dificuldades para estes primeiros animais e todos seus descendentes (e.g. Aiello, 1997; Joseph, 1996; Orsntein, 1991) tais como adaptação termodinâmica, metabólica e reprodução. A postura bípede foi acompanhada por um estreitamento da bacia, requerendo uma antecipação da liberação do feto. Este fato exigiu um aumento nos laços de integração familiar e social como solução para atendimento das necessidades desse cérebro que nasce imaturo. A imaturidade cerebral, entretanto, passa a ser um fator positivo, pois aumenta a capacidade adaptativa do cérebro ao ambiente no qual deve se desenvolver o homem. As dificuldades de liberação do feto, porém, continuam como fatores etiológicos importantes para inúmeras disfunções cerebrais, entre as quais, a paralisia cerebral (e.g., Baron, Fennell and Voeller, 1995; Batchello and Dean, 1996; Spreen, Risser, Edgell, 1955;). Outras patologias congênitas, peri e post-natais de origem genética, infecciosa ou dependente da saúde materna, completam as causas mais frequentes da redução da capacidade cerebral e principalmente, dos distúrbios motores observados em crianças (e.g.; Baron-Cohen, 1995; Duane, 1995; DeFries and Gillis, 1995; Lubs et al, 1995).

Um fator complicador no processo de educação da criança deficiente é a variabilidade das estruturas cerebrais afetadas, o que requer um processo diagnóstico refinado e prático, que possibilite uma melhor compreensão das dificuldades da criança, fundamental para uma boa orientação do processo pedagógico. Por esses motivos, Rocha et al, 1998, 1999 propuseram o desenvolvimento do sistema ENSCER como um sistema informatizado para a análise quantitativa do EEG; processamento e padronização do processo de diagnose médica sobre o cérebro deficiente e para geração de um conjuntos de jogos educacionais para complementar o processo de ensino do cérebro deficiente. A utilização desse sistema com diferentes grupos de crianças tem mostrado que o computador pode ajudar muito o desenvolvimento cognitivo de crianças deficientes cerebrais, inclusive portadores de paralisia cerebral (PC).

As crianças com PC podem apresentar diferentes graus de dificuldades motoras, associadas ou não a uma deficiência mental (e.g. Aicardi, 1998). Quando a deficiência motora é leve, o manuseio do computador pode ser feito com equipamento normal (teclado e mouse), porém quando a habilidade motora se reduz acentuadamente, a solução tem sido o desenvolvimento de mouses e teclados especiais (Santarosa, 1994a) ou dos chamados ponteiros, que devem ser fixados na criança por meio de cintas para permitir a utilização de movimentos da cabeça para manuseio do teclado (Heidrich, 1999). Entretanto, essas soluções são caras, pois devem ser, em geral, projetadas individualmente.

Existem diversos equipamentos eletrônicos e softwares comerciais que ajudam o deficiente físico na operação do computador como exemplo o WinScan (Lacefield & Garthee, 1995), que mostra um quadro de operações e o usuário de alguma forma seleciona a opção desejada para simular a ação desejada, HeadMaster, da Prentke Romich, e o HeadMouse da Words+, que convertem movimentos da cabeça do usuário para movimentos proporcionais do cursor do mouse na tela através de um emissor de luz infravermelha e um sensor óptico. A operação de click do mouse deve ser feita por um dispositivo adicional, acionado por uma outra parte do corpo. Além desses existem outras soluções descritos por CAPOVILLA, F. C. et al. (1998, 2000).

A utilização de recursos tecnológicos na educação de sujeitos com necessidades especiais tem como meta, portanto, opor-se aos métodos mais tradicionais empregados na (re)educação e (re)habilitação destas populações (Valente, 1997; Mantoan e Valente, 1997; Valente, 1999). Neste sentido, não se trata de usar uma ferramenta tecnológica com o objetivo de “corrigir” uma “anormalidade intelectual” (física, sensorial, cognitiva). Mais do que oferecer assistência às necessidades do indivíduo para que ele possa desenvolver o seu potencial cognitivo, criativo e humano. Afasta-se, assim, das propostas educacionais que centram-se em métodos e técnicas na tentativa de corrigir ou minimizar tais desvios.

Da mesma forma o presente trabalho se propõem em dar mais uma alternativa de software capaz de controlar o mouse através de movimentos de um marcador ou algum objeto previamente identificado pelo sistema. Esse marcador pode ser uma etiqueta adesiva colocada em alguma parte do corpo do usuário no qual tenha o maior controle motor possível. Posicionamos uma câmera de vídeo focando o marcador em questão e o transmitimos o seu movimento ao software que controlará o ponteiro do mouse no sistema operacional.

A grande maioria das soluções disponíveis comercialmente são caras ou não atendem a todos de uma maneira geral. A nossa proposta foi o desenvolvimento de um software que pudesse dar mais uma alternativa ao usuário portador de deficiência física, de maneira que o atendesse de uma maneira mais geral e barata. O nosso software necessita apenas de uma câmera de vídeo, por exemplo WebCam, e um computador com o sistema operacional Windows, tendo esse itens o usuário já pode fazer uso do sistema.

Além disso, o acesso ao computador ampliará as possibilidades de sucesso escolar ou profissional do deficiente motor, pois com avanço tecnológico atual e as possibilidade que a Internet nos trás, criam enormes possibilidade de estudos e trabalhos via computador sem a necessidade de locomoção e as barreiras da comunicação do indivíduo são quebradas e suas idéias e opiniões podem ser levadas a qualquer parte do mundo.

1. Funcionamento do Software REMMC

O software REMMC (Reconhecimento Especial de Marcadores para Manuseio de Computadores) foi desenvolvido com a finalidade de ajudar o usuário a manipular o computador sem a ajuda do mouse ou teclado.

O software deve encontrar e reconhecer um marcador colocado em alguma parte do corpo que a pessoa possa movimentar, com o propósito de transformar o movimento do marcador em códigos de controle para a operação do computador.

Como qualquer sistema de controle, o REMMC precisa ser calibrado, isto é, o sistema precisa aprender a forma do marcador antes de inicializar o reconhecimento e além disso o software necessita ter algumas características do ambiente em que o usuário está operando. Com essas informações o sistema pode reconhecer e seguir o movimento do marcador e em seguida transformar esses movimentos em sinais de controle.

As principais etapas desse processo são:



Figura 1.1 - Etapas da Execução do REMMC

1.1. *Calibrando*

O ambiente em que o usuário está operando o software é de bastante interesse ao sistema. Se o sistema estiver calibrado para um ambiente claro e, em um dado momento, o usuário for para um outro local completamente diferente do anterior em relação à luminosidade, por mais que o sistema tente corrigir a imagem, não terá sucesso. Por esse motivo, o sistema tem alguns limites de tolerância para a variabilidade do ambiente em relação a luminosidade. Para isso é necessário informar ao software algumas características do marcador naquele local.

Por exemplo a cor do objeto marcador naquele local. Quando encontramos o marcador a sua cor é automaticamente transferida para opção “nível”, em seguida o usuário define o tamanho da janela em relação ao ambiente de uso.

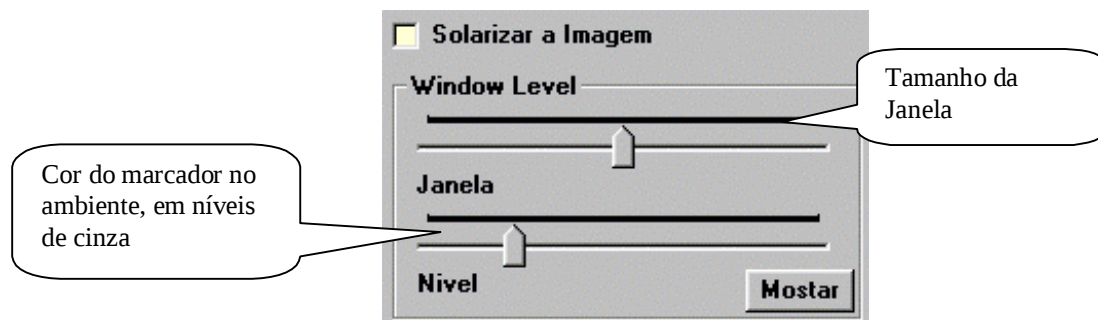


Fig. 1.1.1 - Ajustes do Ambiente de Uso

Ajustando algumas característica do marcador em relação ao ambiente de trabalho.

1.2. Adquirindo Imagem da WebCam

Existe uma grande variedade de câmeras de vídeo no mercado, algumas com custo elevado e outras com custo bem acessível à maioria dos usuários. O ponto inicial do projeto foi a resolução do problema utilizando equipamento de baixo custo, isso só foi possível por dois motivos: O barateamento dos computadores no geral podendo hoje em dia qualquer usuário ter em casa um micro Pentium 333 Mhz com 64 Mb de RAM; e o outro motivo foi a implementação dos filtros em C++ que dá uma grande performance de velocidade ao software.

A obtenção da imagem é feita através da CapCamera.DLL, desenvolvida em visual basic, que vai a posição de memória onde a imagem está sendo criada e a coloca em uma matriz. Este processo é bastante rápido porque só existe transferência de dados de memória para memória.

Exemplo:

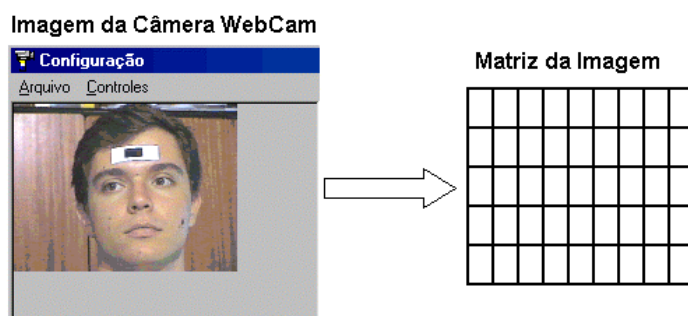


Fig. 1.2.1 - Aquisição da Imagem

1.3. Filtrando a Imagem

Todo o processamento do software se concentra na filtragem e separação de todos os objetos da imagem e a seleção do objeto marcador dos demais. Para ser realizado tal processamento é necessário que a imagem obtida passe por diversos filtros até chegar num ponto em que a imagem original se transforme em uma imagem binária mas mantendo todas as características dos objetos da imagem original.

Todos esses filtros estão agrupados na biblioteca Camera.dll, feita em Visual C++ 6.0, que será descrita mais adiante.

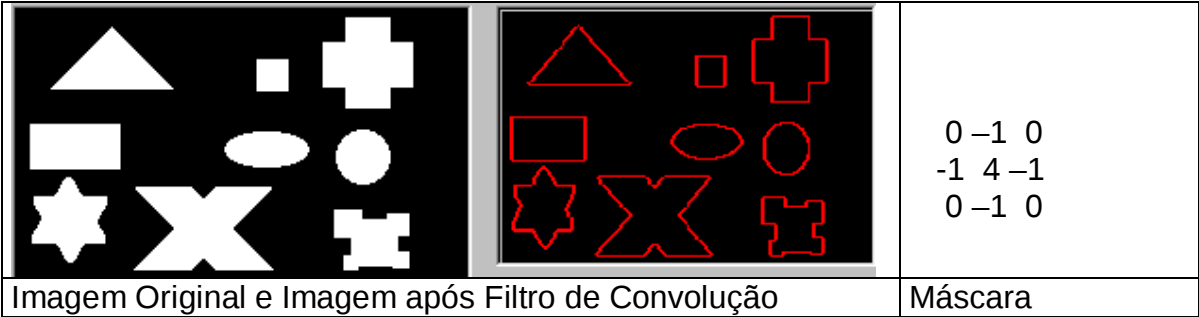


Fig. 1.3.1 - Aplicação de Filtros

1.4. Encontrando o Marcador

O marcador nada mais é que um objeto identificado previamente no sistema e pode ter qualquer forma ou cor.

Através de vários testes chegamos à conclusão de que o melhor marcador é aquele que tem a cor preta e impresso em uma etiqueta branca. Com o objeto no centro da etiqueta. Isso facilita bastante a separação dos objetos para a escolha do marcador. A escolha da cor preta se deve ao fato do nível de cinza do marcador preto estar perto de zero e, mesmo que haja alguma variação na luminosidade da imagem ele sempre estará em uma região mais escura que as demais.

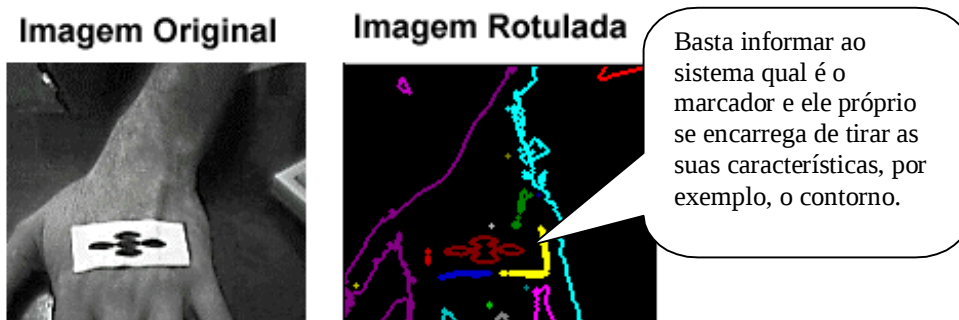


Fig. 1.4.1 - Definindo um Marcador

1.5. Transformando Marcador em Movimento

Se o sistema estiver apto a reconhecer o marcador dentro de uma imagem captada pela câmera basta o usuário movimentar o marcador para a direção desejada e o próprio software informará ao sistema operacional o novo posicionamento do ponteiro do mouse. Do mesmo modo existem movimentos padronizados para reproduzir o click ou duplo click do mouse.

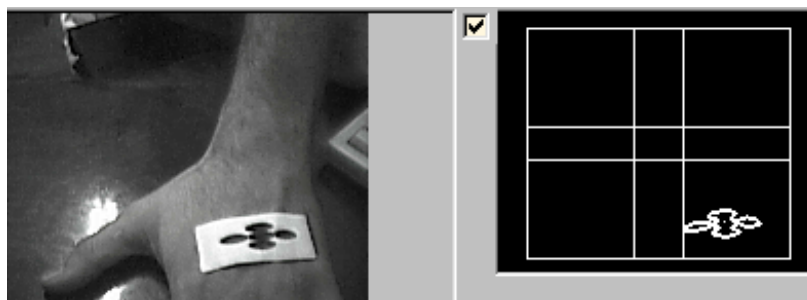


Fig. 1.5.1 - Identificando o Marcador

O objeto marcador foi encontrado no canto inferior direito do quadro. O programa tem que transferir esse movimento para o mouse, na direção diagonal direita inferior.

Neste caso o mouse irá se movimentar nesta direção enquanto o marcador estiver nesta posição. Para que o mouse pare, basta retornar o marcador ao centro do quadro. Para chegar a esse resultado o processo de transformação do marcador em movimento passa por duas fase :

1.5.1. Codificando o Movimento

A movimentação será detectada através da codificação da mudança relativa da posição do objeto marcador entre os frames i e $i+1$ de acordo com a metodologia introduzida por *Serapião et al, 1996* e ilustrada na Fig. 1.5.1.1. A posição no frame i será tomada como imagem origem de referência, e a posição do frame $i+1$ será referenciada à direção padrão mais próxima. Desta maneira podemos comparar a trajetória do marcador e saber a direção do seu movimento. No processo de codificação o sistema retira de cada objeto marcador seu ponto central, que será comparado com a demais imagens, desta forma o sistema ganha em simplicidade e rapidez.

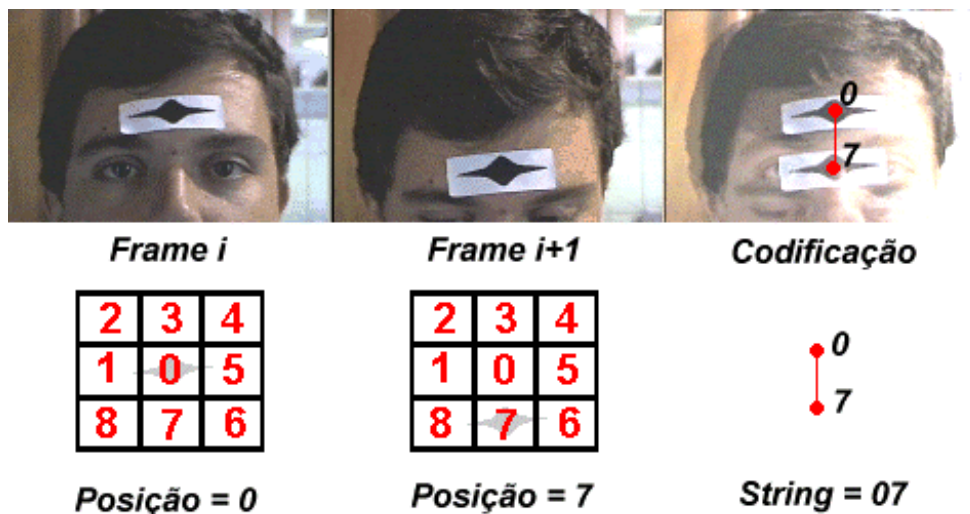


Fig. 1.5.1.1 - Codificando o Movimento do Marcador

Como mostra a figura acima a base de toda a codificação é o ponto central do marcador a partir dessa informação é possível rastrear toda a trajetória do objeto dentro da imagem criando uma string codificada como mostra os exemplos abaixo.

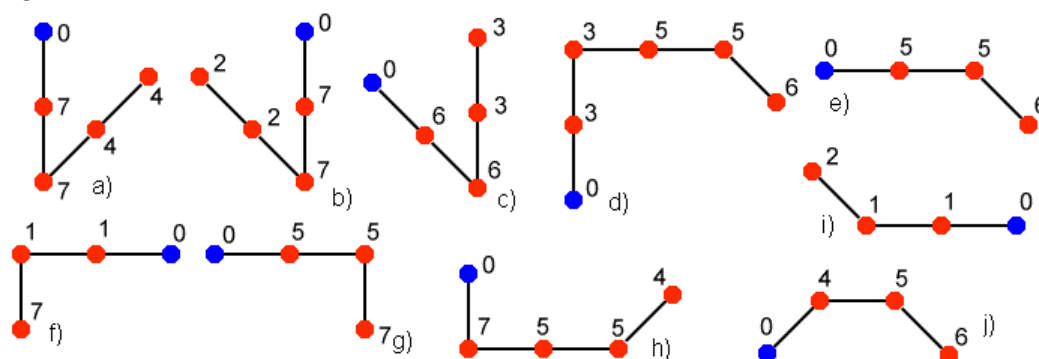


Fig. 1.5.1.2 - Exemplos de Movimentos Codificados

O bolinha azul indica que o marcador está parado no centro, esta informação é representada como 0, quando existir um movimento o sistema irá representar a sua trajetória conforme a tabela mostrada na Fig. 1.5.1.1.

Por exemplo o item f), tem uma string codificada na seguinte forma 0557.

- 0, Marcador no centro da imagem
- 5, Marcador foi para o lado direito
- 5, Marcador continua no lado direito
- 7, Marcador desceu

1.5.2. Interpretando a String Codificada do Movimento

A codificação do movimento gera uma cadeia de caracteres na sintaxe descrita no item anterior o próximo passo é a interpretação desta informação codificada de forma que possamos retirar operações de controle e filtrar possíveis ruídos.

No exemplo da Fig. 1.5.2.1, existem duas cadeias de caracteres que são representadas pela codificação do sistema REMMC. Nas cadeias abaixo o controle 0, significa que o mouse está parado, quando existir qualquer movimento diferente de 0, o sistema inicializa um contador que espera a repetição do mesmo movimento até que atinja um determinado limite a partir desse instante o mouse começa a se movimentar na direção desejada, quando o usuário quiser mudar de direção terá que esperar novamente este ciclo, esta repetição do mesmo movimento é muito importante no sistema porque ajudar a identificar ruídos ou operações padronizadas como mostra a Fig. 1.5.2.1.

a) 000555555555500007000077777777700
b) 003332333233333333333000077777700

- Mouse Parado
- Intenção de Movimento
- Movimento do Mouse
- Operações Padronizadas
- Ruído

Fig. 1.5.2.1 - Cadeia de Caracteres Codifica

A cadeia de caracteres pode ser redefinida em parte menores chamadas de *palavras* que são delimitadas pelo caractere 0, que representa o mouse parado no centro da imagem.

Pode-se definir três classes de palavras para controle do computador, de acordo com seu número de caracteres:

Curtas: 0 C₁...C_i 0

Médias: 0 C₁...C_j 0 onde j>i

Longas 0 C₁...C_n 0 onde n>j

Onde :
C: representa um movimento de 1 até 8
i: representa um limite de repetição curto
j: representa um limite de repetição médio
n: acima deste limite representa um movimento do mouse

Utiliza-se:

As palavras curtas para implementar códigos de controle:

030 (Click do Mouse)
070 (Duplo do Mouse)
010 (BackSpace)

As palavras médias para implementar o controle de teclado:

0760 (Letra A)
0780 (Letra B)
0320 (Tecla ENTER)
0340 (Tecla ESC)

As palavras longas para controle de direção do mouse:

05555555555550 (Mover Mouse para Direita)
07777777777770 (Mover Mouse para Baixo)

As operações padronizadas podem ser criadas a partir de movimentos que o usuário possa repetir de uma maneira simples e seqüencial, a fim que o mesmo possa usufruir de todos os recursos do computador. No exemplo acima, item *a*, existe um movimento representado por *070*, que significa marcador no centro, marcador para baixo e volta o marcador para o centro, o sistema REMMC irá interpretar essa informação como sendo um duplo click.

Quando o sistema percebe que o marcador não está mais no centro o sistema espera um ciclo de repetição para começar a mover o mouse, enquanto isso não ocorre, o sistema pode interpretar esse pedaço da informação chamado de *palavra*, como sendo um sinal de controle, como o limite da repetição pode ser configurável o usuário não precisa ser tão rápido para dar o duplo click, ele pode realizar a mesma operação realizando um movimento mais lento, por exemplo: *07770*. Esse limite de repetição define os conjuntos de palavras curtas, médias ou longas, como discutido acima.

No item *b* da Fig. 1.5.2.1, o mesmo acontece só que este movimento padronizado *030*, representa o click do mouse. Outra característica importante são os possíveis ruídos que podem ser gerados, nesse caso o sistema primeiro irá interpretá-lo como sendo um movimento padronizado, caso negativo irá ignorá-lo e iniciará o contador de intenção de movimento para 0, reiniciando a interpretação normal do movimento. Este mesmo processo ocorre também quando existe um movimento padronizado.

Tendo em vista estas opções, fica claro a possibilidade de ampliar os movimentos padronizados a fim de que possa cobrir todas as teclas do teclado mais os movimentos e controles do mouse como mostra a Fig. 1.5.2.2.

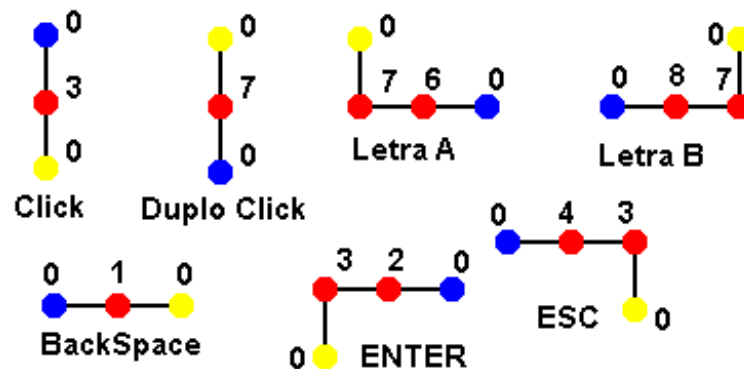


Fig. 1.5.2.2 - Palavras Codificadas no Sistema

No exemplo acima as bolinhas amarelas simbolizam o início do movimento padronizado e a bolinha azul indica o final da seqüência, e todos os caracteres entre este dois delimitadores representam a simulação da operação pretendida. Neste trabalho realizamos somente testes com a utilização do mouse.

Todas essas opções aqui descritas podem ser configuradas dentro do sistema, com a finalidade de atender a um grande número de indivíduos respeitando seus limites de velocidade e controle do sistema.

Essa é a seqüência básica de todo o sistema, primeiramente configuramos o ambiente de uso com todas as informações possíveis sobre o ambiente, depois a execução propriamente dita do software em background que fica repetindo todos os procedimentos anteriores a partir da aquisição da imagem até a localização, reconhecimento e movimento do objeto marcador. Lembrando que todos esses passos são processados praticamente em tempo real.

2. Referências

Aicardi, J. Diseases of the nervous system in childhood, Mac Keith Press, London, 1998

Aiello, L. C. - Brains and guts in human evolution: The expensive tissue hypothesis. *Braz. J. Genetics*, 1997/20:141-148

Alegre, L., A. F. Rocha and C. K. Morooka - Intelligent Approach of Rod Pumping Problems. *SPE*, 1993:26253/249-255

Baron, I. S., E. B. Fennell, K. K. S. Voeller - *Pediatric Neuropsychology in the Medical Setting*. Oxford University Press, Oxford and N. York, 1995

Baron-Cohen, S. - *Mindblindness*. Bradford Books, The MIT Press, Cambridge, 1995

Barreto Filho, M. A., M. Tygel, A.F. Rocha and C. K. Morooka Automatic downhole card generation and classification. *SPE* 1996, 311-318 (36605)

Batchelor, E. S. and R. S. Dean - *Pediatric Neuropsychology*. Allyn and Bacon, Boston, 1996

Damasio, A. R. (1990) Time-locked multiregional retroactivation: A systems-level proposal for the neural substrates of recall and recognition. In *Neurobiology of Cognition*, Eimas and Galaburda (editors), The MIT Press, 25-62

DeFries, J. C. and J. J. Gillis. Etiology of reading deficits in learning disabilities: Quantitative genetic analysis. In *Developmental Neuropsychology*, (Spreen, O., A. H. Risser and D. Edgell, eds.), Oxford University Press, Oxford and N. York, 1995

Duane, D. D. - Biological foundations of learning. In *Developmental Neuropsychology*, (Spreen, O., A. H.

Risser and D. Edgell, eds.), Oxford University Press, Oxford and N. York, 1995

GONZALEZ, Rafael & WOODS, Richard. "Digital Image Processing.", Addison Wesley, 1992

HEIDRICH, Regina de Oliveira "Criatividade na Educação de Deficientes Mentais com o uso de Softwares Educativos." Dissertação de Mestrado, UNESP, Bauru, 1999.

Joseph, R. Neuropsychiatry, Neuropsychology and Clinical Neuroscience. Williams and Wilkins, Baltimore Philadelphia, 1996

Lubs, H. A. M. Rabin, K. Carland-Saucire, X. L. Wen, K. Gross-Gelnn, R. Duara, B. Levin and M. L. Lubs. Genetic bases of developmental dyslexia: Molecular studies. In Developmental Neuropsychology, (Spreen, O., A. H. Risser and D. Edgell, eds.), Oxford University Press, Oxford and N. York, 1995

Ornstein, R. (1991) The evolution of consciousness (A Touchstone Book, Simon & Schuster, N. York)

PETZOLD, Charles & YAO, Paul. "Programando para Windows 95.", Editora Makron Books do Brasil, 1996

PETZOLD, Charles, "Programming Windows.", 5. ed. Microsoft Press, 1998

ROCHA, A. F. "O cérebro: Um breve relato de sua função." EINA, Jundiaí, 1999

ROCHA, A. F. "The brain as a symbol processing machine, Progress in Neurobiology", v. 53, no.2, pp. 121-198, 1997

Rocha, A. F. - Neural Nets: A theory for brains and machine. Lecture Notes in Artificial Intelligence, Springer-Verlag, 1992.

ROCHA, A. F. - The brain as a symbol processing machine. Progress in Neurobiology, 1997/53:121-198

Rocha, A. F., A. B. Serapião; A. Rondó, E. C. Rodella; F. Luchini; M. P. Rebello. Correlating Intelligence and Brain Activity. Proc. 6th Ann. Meeting Cognitive Neurosciences Society, 1999 Washington, D.C, USA

Rocha, A. F., et al, Ccomputer diagnosing and teaching of the brain impaired children. Proc. 5th Ann. Meeting Cognitive Neurosciences Society, 1998 San Fracnisco, Ca, USA.

Santarosa, L. M. C., Martins, A. R., Silveira, M. S., Franco, B. S. (1994a) Adaptação para o Português e Avaliação de um simulador de Teclado para portadores de Paralisia Cerebral. In: Anais do II Congresso Ibero-americano de Informática na Educação . Portugal: Vol. 1., p. (116-118).

Santarosa, L. M. C., Silveira, M. S., Virti, D. S. (1994b) Hipermeios na Construção da Leitura e Escrita: ambiente para Crianças e Portadores de Deficiências. In: Anais do II Congresso Ibero-americano de Informática na Educação . Portugal: Vol. 1., p. (119-121).

SERAPIÃO, A. B. S. , "Sensor: An Artificial Visual System", Progress Neurobiology, v. 53, no.2 pp.179-183, 1997.

SERAPIÃO, A. B. S., A. F. ROCHA, and M. P. F. REBELO. "Toward a Theory of Genetic Computing". In Genetic Algorithms and Soft Computing, F. Herrera and J. L. Verdegay, (Eds) (Physica Verlag, Heidelberg, 1996);68-94

Spreen, O., A. H. Risser and D. Edgell - Developmental Neuropsychology. Oxford University Press, Oxford and N. York, 1995