

Interfaces prestativas baseadas em visão computacional e informação de contexto

Thiago Schumacher Barcelos

Texto do exame de qualificação apresentado como requisito parcial à obtenção do grau de Mestre em Ciência da Computação ao Instituto de Matemática e Estatística da Universidade de São Paulo.

Orientador: Prof. Dr. Carlos Hitoshi Morimoto

SÃO PAULO
2004

Resumo

O surgimento de dispositivos computacionais em formatos diversos e que não se assemelham aos computadores de mesa atuais exige o desenvolvimento de novas técnicas de interação que se adaptem às características desses novos dispositivos. Na busca por novas modalidades de comunicação a serem incorporadas a essas novas técnicas, uma possibilidade em estudo há alguns anos é o uso de técnicas de rastreamento de olhar. Neste trabalho descrevemos o estado-da-arte na aplicação do rastreamento de olhar a sistemas interativos, e propomos uma extensão baseada em informação de contexto para o MAGIC Pointing, uma técnica de interação que associa o controle motor com o rastreamento de olhar para a execução de tarefas de apontamento em interfaces WIMP usuais.

Abstract

New computer devices in alternative formats that are not similar to regular desktop systems call for the development of new interaction techniques that are better suited to the characteristics of these new devices. In the search for communication modalities applicable to new interaction techniques, the use of eye tracking devices has been considered for some years now. In this paper we describe the state-of-art in the application of eye tracking techniques to the development of interactive systems; we also propose an extension based on context information to MAGIC Pointing, an interaction technique that combines regular motor control of a pointing device with data gathered from an eye tracking device to help a user perform pointing tasks in regular WIMP interfaces.

Sumário

1	Introdução	1
1.1	Organização deste trabalho	3
2	Visão, rastreamento e aplicações	5
2.1	Estrutura do olho humano	5
2.2	Técnicas de rastreamento de olhar	7
2.2.1	Rastreamento de olhar intrusivo	8
2.2.2	Rastreamento de olhar não-intrusivo	9
3	Técnicas de Interação	13
3.1	Conceitos de projeto	14
3.1.1	O olho	14
3.1.2	Identificação de fixações	14
3.1.3	Seleção e o problema do “Toque de Midas”	15
3.1.4	Modelagem do contexto e interfaces prestativas	16
3.2	Técnicas de interação implementadas	18
3.2.1	<i>Mundo das janelas</i>	19
3.2.2	<i>Pequeno príncipe</i>	19

3.2.3	Seleção por tempo de latência	20
3.2.4	MAGIC Pointing	21
3.2.5	Renderização de simulações em tempo real	23
3.2.6	Modelagem do contexto	23
3.3	Análise comparativa	26
4	Proposta deste trabalho	29
4.1	Implementação do MAGIC Pointing	30
4.2	Extensões do MAGIC Pointing	34
4.3	Teste com usuários	35
4.3.1	Tarefa	36
4.3.2	Procedimento	38
4.3.3	Avaliação	40
5	Rumos futuros	49
5.1	Plano de trabalho	50
A	Questionário do teste	51

Lista de Figuras

2.1	Estrutura do olho humano	6
2.2	Pupila iluminada; as linhas indicam o centro da pupila e do brilho ocular	9
2.3	Imagens de Purkinje	10
3.1	Seleção usando tempo de latência (conforme exibido em [SJ00])	21
3.2	Digitação de ideogramas do idioma chinês através do <i>pinyin</i> [WZS01]	22
4.1	Câmera e conjuntos de <i>leds</i> infravermelhos	31
4.2	Pupila iluminada (a), pupila escurecida (b), e diferença entre as imagens (c) . .	31
4.3	Aplicação de testes	36
4.4	Avaliação dos usuários: rapidez da modalidade	42
4.5	Avaliação dos usuários: conforto da modalidade	43
4.6	Ilustração de dois testes propostos por Fitts	44
4.7	Lei de Fitts: retas $T = a + b.ID$ para cada modalidade	47
A.1	Questionário aplicado aos participantes do teste	52

Lista de Tabelas

3.1	Exemplo de gramática completa: passos cognitivos para digitação de palavras	25
4.1	Modalidade preferida: de 1 (para a melhor) a 5 (para a pior)	43
4.2	Coefficientes de correlação linear - parâmetros da Lei de Fitts	45
4.3	Coefficientes de regressão linear para $T = a + b.ID$, onde $ID = \log_2 A/W + 1$.	47
5.1	Plano de trabalho: agosto/2004 a fevereiro/2005	50

Capítulo 1

Introdução

Nos últimos vinte anos, nos computadores de uso pessoal se consolidou o uso de uma arquitetura de periféricos de entrada e saída padronizada: entrada de dados através de teclados alfanuméricos e do *mouse*; saída de dados visual através de um monitor, e eventualmente complemento sonoro através de uma placa sintetizadora de sons. Essa arquitetura se popularizou juntamente com o paradigma de interação WIMP (Windows, Menus, Icons and Pointing), usado em sistemas como o MacOS, da Apple e o Windows, da Microsoft.

Recentemente, o avanço de tecnologias de miniaturização de componentes permitiu o surgimento de novos dispositivos computacionais em formatos diversos, como os *PDA's* (Personal Digital Assistants) e *tablet PCs*, e a incorporação de recursos computacionais em equipamentos diversos e até no ambiente (no fenômeno conhecido como computação ubíqua). Essa nova realidade exige o desenvolvimento de novas formas de interação homem-máquina cujas características se adaptem melhor às particularidades desses novos ambientes.

Uma proposta que vem sendo discutida é a da multimodalidade, ou seja, incorporar ao diálogo com dispositivos computacionais mais informações sobre o usuário como formas alternativas de entrada, tais como a voz, o olhar, ou movimentos corporais, de forma a criar

diferentes formas de diálogo para usuários de diferentes faixas etárias, níveis de instrução e limitações físicas e motoras [OC00], além de possibilitar a criação de formas de interação que aproveitem as habilidades de comunicação naturalmente adquiridas pelos usuários no convívio social [Tur98]. Através de uma maior variedade de modalidades de comunicação, abrem-se novas possibilidades para que os computadores não fiquem mais o tempo todo passivamente esperando por uma ação do usuário e possam, a partir de um maior conhecimento do estado do usuário, tomar a iniciativa de auxiliá-lo na execução de suas tarefas – convencionou-se chamar as interfaces humano-computador criadas segundo esse princípio de *interfaces prestativas* [Tur98, Ver02].

Em particular, a incorporação de técnicas de rastreamento de olhar em interfaces tem sido considerada por vários anos. Através de diferentes abordagens, o objetivo de tais técnicas é a determinação do ponto observado por um indivíduo em uma superfície 2D, ou a estimação da linha de visão do indivíduo, quando não é considerada a observação de uma superfície em particular. Como nós usamos nossos olhos para explorar constantemente o ambiente à nossa volta, potencialmente essa informação pode colaborar na construção de um canal de comunicação natural e rápido em comparação com os métodos de entrada tradicionais, tais como o teclado e o mouse.

Porém, alguns fatores devem ser levados em consideração no desenvolvimento de técnicas de interação que incorporem o rastreamento de olhar. Restrições inerentes à tecnologia atual introduzem uma quantidade considerável de ruído nos dados oriundos do processo de rastreamento [GEN95]. Além disso, pela sua própria estrutura física, o olho não é uma “ferramenta” precisa que usamos para manipular o ambiente à nossa volta como as mãos, por exemplo.

Assim, interpretar corretamente os dados fornecidos pelo processo de rastreamento de olhar é um desafio no projeto de tais técnicas de interação.

Nosso trabalho se desenvolve a partir de uma técnica de interação desenvolvida por Zhai,

Morimoto e Ihde [ZMI99], denominada MAGIC (Manual and Gaze Input Cascaded) Pointing, que usa a informação da posição da tela observada pelo usuário, associada ao controle motor de um dispositivo de apontamento tradicional para a execução de tarefas de apontamento em interfaces WIMP. Essa técnica permite, a partir da informação sobre o olhar, que o cursor usado para o apontamento se mova de forma mais “prestativa”, ou seja, apareça na vizinhança da posição que recebe a atenção visual do usuário.

A nossa proposta é associar ao MAGIC Pointing informações sobre o contexto atual da interação, aproximando-o mais do conceito de uma interface prestativa, a fim de investigar como essa combinação pode produzir uma interface melhor tanto em aspectos objetivos, como a velocidade ou precisão na realização de tarefas, quanto em aspectos subjetivos como a satisfação dos usuários. Para tanto, desenvolvemos uma versão para Linux do sistema de rastreamento de olhar proposto por Morimoto, Koons, Amir e Flickner [CKAF00], que, juntamente com informações obtidas a partir do contexto da interação, é usado pelo MAGIC Pointing e por suas extensões a serem propostas por nós. Além disso estamos projetando e conduzindo testes com usuários operando o sistema para a obtenção de medidas que permitam avaliar as vantagens e desvantagens dessas extensões.

1.1 Organização deste trabalho

No Capítulo 2 descreveremos brevemente algumas características da estrutura e dos movimentos realizados pelo globo ocular humano relevantes para compreender as dificuldades encontradas no desenvolvimento de uma técnica de rastreamento de olhar; depois, exibiremos e compararemos diferentes técnicas descritas na literatura, discutindo as vantagens e desvantagens de cada uma.

No Capítulo 3 daremos uma definição de *técnica de interação*, e descreveremos como o ras-

tratamento de olhar foi empregado até o momento na área de Interação Humano-Computador através da comparação entre algumas técnicas de interação já implementadas; veremos também alguns exemplos da associação do rastreamento de olhar ao uso de informação de contexto que tem sido propostos.

O Capítulo 4 descreve a proposta do nosso trabalho, a partir da discussão do estado-da-arte feita nos capítulos anteriores. Também será exibida a definição e resultados de um teste com usuários que mostram os resultados preliminares de nosso trabalho. Por fim, no Capítulo 5 discutiremos os rumos futuros do desenvolvimento do trabalho a partir da nossa avaliação dos resultados obtidos até o momento e das dificuldades encontradas.

Capítulo 2

Visão, rastreamento e aplicações

2.1 Estrutura do olho humano

A visão é um dos sentidos mais importantes na percepção do ambiente à nossa volta; a todo momento, a usamos para obter informações, aprender e guiar a maior parte dos nossos movimentos, dentre outras aplicações.

A parte externamente visível do olho é composta pela esclerótica (a parte branca do olho), pela íris (a parte colorida do olho) e pela pupila. A íris é protegida e coberta pela córnea, uma membrana transparente. A pupila é a abertura circular no centro da íris que controla a entrada de luz através do aumento ou diminuição do seu tamanho. Atrás da pupila localiza-se o cristalino, uma lente transparente que capta a luminosidade, focando a imagem observada no fundo do olho, onde encontra-se a retina (Figura 2.1).

A retina é a região que contém células fotossensíveis, as principais responsáveis pelo processo de detecção da luz. Dessas células, a proporção de 94% é dos chamados *bastonetes*, estruturas com alta sensibilidade à luz, permitindo a visão mesmo em ambientes com pouca luminosidade, mas pouca sensibilidade à cor; o restante compõe-se dos *cones*, pouco sensíveis

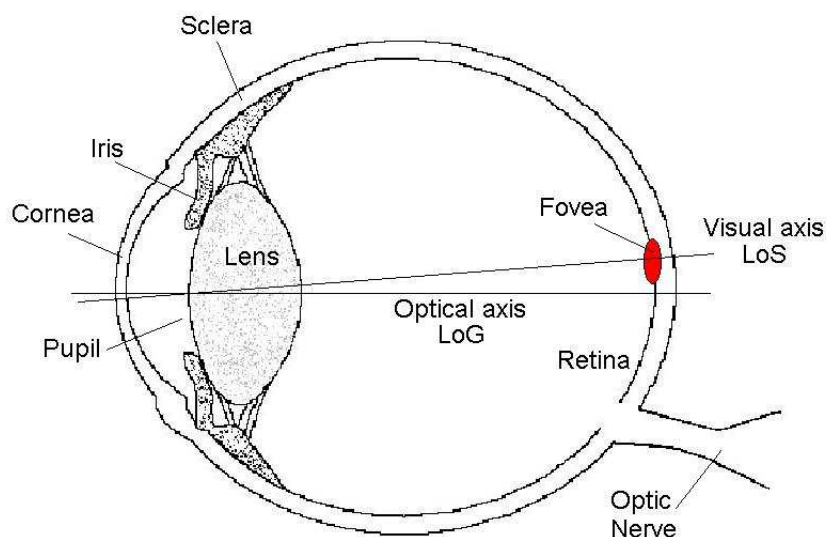


Figura 2.1: Estrutura do olho humano

à luz mas muito sensíveis às cores [GEN95].

Os cones estão concentrados no centro da retina – de outra forma, a distância entre eles seria muito grande, prejudicando a acuidade visual em condições normais de iluminação do dia. Nessa área na retina há uma depressão, chamada *fóvea*, constituída completamente por cones, e responsável pela visão detalhada. A fóvea não se localiza exatamente no eixo óptico do olho, definido como a semi-reta que passa pelo centro do globo ocular e o centro da pupila; o eixo que realmente define para onde está direcionada a atenção visual do indivíduo é definido pela semi-reta que passa pelo centro da fóvea e o centro da pupila. A fóvea cobre um ângulo visual de aproximadamente 1 grau.

A cada momento, o objeto que recebe atenção visual deve ser focado na fóvea; este evento é chamado de *fixação*, e dura entre 200 e 600 milissegundos [Jac90]; o olho muda de posição entre uma fixação e outra através de um *movimento sacádico*. Este é um movimento voluntário, que dura entre 30 e 120 milissegundos; porém sua natureza é dita *balística*, ou seja, uma vez

iniciado não é possível mudar sua direção ou a posição final [GEN95]. Mesmo durante uma fixação, o globo ocular não permanece estático o tempo todo. Ele sofre pequenas mudanças de direção, corrigidas por movimentos de ajuste, chamados *movimentos microssacádicos*, para que o objeto de interesse permaneça focado na fóvea [Jac95]. Além disso, o globo ocular pode realizar outros movimentos, que acontecem para o ajuste do foco visual a objetos em movimento, ou a largos movimentos da cabeça [BG90, YS75].

2.2 Técnicas de rastreamento de olhar

Um conjunto de diferentes técnicas desenvolvidas ao longo do último século podem ser classificadas sob a denominação de técnicas de rastreamento de olhar [YS75]. Inicialmente, tais técnicas foram empregadas nos campos da psicologia experimental, da oftalmologia e da neurologia e áreas correlatas, como ferramenta para compreender como a busca visual participa de processos cognitivos, e também no estudo e detecção de anormalidades nas características motoras do olho. O contínuo aumento da capacidade de processamento e diminuição de custo dos microprocessadores, aliado à ampla disponibilidade de dispositivos que permitem a captura de dados para o processo de rastreamento de forma não intrusiva (por exemplo, câmeras de vídeo), permitiu o desenvolvimento de técnicas de rastreamento de olhar razoavelmente robustas e que operam em tempo real.

As técnicas de rastreamento de olhar podem ser agrupadas em duas categorias:

intrusivas: requerem o contato físico de algum tipo de instrumento de medição com o indivíduo.

não-intrusivas: se utilizam da obtenção de imagens de vídeo do olho, da cabeça ou de ambos e posterior aplicação de algoritmos de visão computacional para estimar a posição observada pelo usuário.

Nas seções seguintes descreveremos e comentaremos implementações de rastreadores de olhar que pertencem a essas duas categorias.

2.2.1 Rastreamento de olhar intrusivo

Young e Sheena [YS75] descrevem algumas técnicas de rastreamento de olhar que exigem o contato de instrumentos de medição com o indivíduo. Um exemplo é a técnica conhecida como *eletrooculografia*, que se baseia em pequenas diferenças de potencial elétrico existentes entre a córnea e a retina. Essa técnica exige a colocação de eletrodos ao redor dos olhos para a medição, e é sensível à variações na luminosidade ambiente: um ambiente escuro pode induzir a diminuição das diferenças de potencial, impossibilitando a medição. Em boas condições, a precisão do método é de 1,5 a 2 graus do ângulo visual. Os cuidados necessários com a aparelhagem de medição e a necessidade do uso dos eletrodos tornam problemática o uso dessa técnica fora do campo médico [GEN95].

Robinson [Rob63] descreve uma técnica que se utiliza de uma lente de contato especial que reage à campos eletromagnéticos de alta frequência gerados ao redor do indivíduo. A lente é fixada através de uma pequena sucção no olho do indivíduo, permitindo assim determinar a linha de visão com uma boa precisão, cerca de 0,08 graus do ângulo visual. Porém, essa é uma das técnicas mais invasivas, além da desvantagem do desconforto gerado pelo uso de uma lente especial, e de questões de saúde a respeito da geração de campos de alta frequência ao redor do indivíduo [GEN95].

Apesar de poderem trazer uma medição mais precisa, as técnicas intrusivas de rastreamento de olhar tem em comum o desconforto e a pouca praticidade causados pelo contato com instrumentos de medição. Esses fatores tornam sua aplicação ao controle de interfaces praticamente impeditivo [Jac90, GEN95], restringindo sua aplicação à execução de medições experimentais.

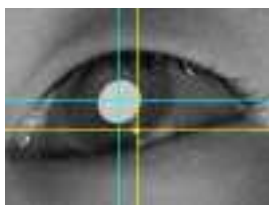


Figura 2.2: Pupila iluminada; as linhas indicam o centro da pupila e do brilho ocular

2.2.2 Rastreamento de olhar não-intrusivo

A partir da obtenção de imagens em vídeo do olho, as técnicas não intrusivas estimam a posição observada pelo indivíduo através do reconhecimento de alguma característica do olho. Uma delas é a fronteira entre a esclerótica e a íris; como a esclerótica é, em condições normais de saúde ocular, branca, e a íris é de uma cor mais escura, essa fronteira pode ser detectada e rastreada facilmente. Mas na maioria dos indivíduos a íris é parcialmente coberta pelas pálpebras, limitando bastante a precisão vertical do rastreamento [GEN95, YS75].

Várias técnicas (por exemplo [OMY02, CKAF00, MSS91]) são baseadas na identificação da pupila, que é menos sujeita a variabilidades individuais do que a fronteira entre a esclerótica e a íris. O contraste entre a pupila e a íris é, por outro lado, muito menor. Uma estratégia para minimizar esse problema é usar uma fonte de luz alinhada com o eixo óptico da câmera, de forma a gerar uma imagem iluminada da pupila, como o efeito de “olho vermelho” que se nota em fotografias amadoras (Figura 2.2). Esse efeito é possível graças à reflexão da luz pelo fundo do olho. Com uma câmera sensível adequada, é possível usar uma fonte de luz infravermelha não visível que não distraia o usuário.

Uma vez detectada a posição da pupila, é desejável que seus movimentos possam ser estimados em relação à outra característica do olho. Uma possibilidade usual é a detecção de uma *imagem de Purkinje* (Figura 2.3), que é uma das reflexões de luz que aparecem em diferentes camadas do olho. A primeira imagem de Purkinje é a reflexão da luz pela córnea, e

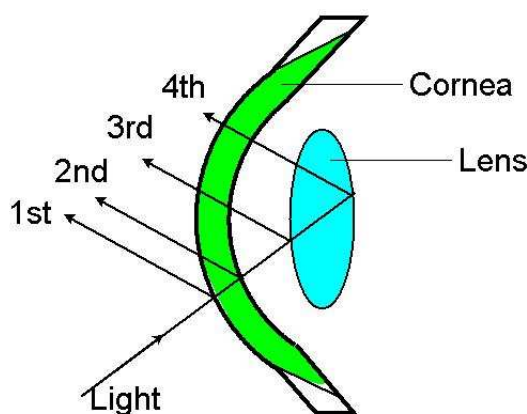


Figura 2.3: Imagens de Purkinje

é a mais clara e simples de ser identificada, seguida pela quarta imagem, que é a reflexão da luz pela parte de trás do cristalino [YS75].

Baluja e Pomerlau [BP94] desenvolveram uma técnica alternativa que não tenta identificar nenhuma característica do olho em particular, mas usa uma rede neural para identificar a posição observada. São obtidas imagens de um ângulo mais aberto, captando a cabeça do usuário, a partir das quais segmenta-se a imagem do olho, que é usada como entrada na rede neural. A técnica permite algum movimento da cabeça, porém sua precisão é pior do que a obtida por outras técnicas, ficando em torno de 2 graus do ângulo visual.

Independentemente da característica do olho que é reconhecida e rastreada, há a necessidade de se mapear a posição de tal característica na imagem de vídeo em relação à posição da linha de visão do indivíduo. Para tal, as técnicas de rastreamento não-intrusivas exigem a execução de algum procedimento de calibragem. O procedimento usual de calibragem é solicitar que o usuário olhe para alguns pontos na tela, enquanto a correspondência com a posição da característica do olho na imagem é medida.

Então, os dados gerados posteriormente no processo de rastreamento serão então ajustados

a partir dos parâmetros calculados no processo de calibragem. Morimoto *et al.* [CKAF00] descrevem um cálculo simples da posição observada, baseado em uma transformação de segunda ordem. Ohno *et al.* [OMY02] utilizam modelos geométricos do globo ocular associados a propriedades de refração da luz para calcular a posição observada. Mesmo na técnica baseada em redes neurais [BP94], há a necessidade de um procedimento similar ao de calibragem para o treinamento da rede neural; o usuário deve acompanhar um cursor em movimento por cerca de três minutos para a obtenção de imagens para o treinamento.

Como a posição do olho em relação à posição observada varia com a movimentação da cabeça, outra limitação exigida é que a cabeça do indivíduo permaneça estática durante e após o processo de calibragem, em geral com o uso de um apoio para o queixo. Algumas adaptações propostas permitem porém alguma movimentação da cabeça; por exemplo Morimoto *et al.* [CKAF00] implementam um servo-mecanismo na câmera que reorienta a captura de imagens de acordo com pequenas mudanças de posição da cabeça.

As técnicas de rastreamento de olhar não-intrusivas tem em comum algumas limitações: os usuários tem que passar por um muitas vezes tedioso procedimento de calibragem cada vez que usam o sistema, além de não poderem mexer livremente a cabeça. Porém mesmo com essas limitações o desconforto ainda é menor do que o causado pelas técnicas intrusivas, e por isso rastreadores de olhar baseados na captura de imagens em vídeo tem sido usados com freqüência no desenvolvimento de sistemas interativos baseados em rastreamento de olhar. As técnicas de interação expostas na literatura que utilizam-se de rastreamento de olhar têm que contornar as limitações inerentes à tecnologia atual de rastreamento, como veremos no próximo capítulo.

Capítulo 3

Técnicas de Interação

Uma *técnica de interação* é uma forma de usar um dispositivo físico de entrada para executar uma tarefa em um diálogo entre um ser humano e um dispositivo computacional [Jac90]. Com o desenvolvimento de técnicas de rastreamento de olhar razoavelmente robustas, não-intrusivas e capazes de fornecer dados em tempo real, surgiu o desafio: como o rastreamento de olhar pode participar da construção de uma técnica de interação?

Na seção 3.1 vamos introduzir os problemas mais comuns que surgem a partir da aplicação de técnicas de rastreamento de olhar a interfaces, e propostas para superá-los descritas na literatura; além disso vamos expor os conceitos de interfaces prestativas e multimodalidade, amplamente discutidos na atualidade da pesquisa em IHC, e onde se insere o rastreamento de olhar nesse contexto. Na seção 3.2 vamos descrever e comentar a implementação das propostas de algumas técnicas de interação que incorporam o rastreamento de olhar, e na seção 3.3 discutiremos as vantagens e desvantagens das técnicas descritas na seção anterior.

3.1 Conceitos de projeto

3.1.1 O olho

O processo de rastreamento de olhar encontra na estrutura do olho humano um fator fundamental de limitação de sua precisão. Conforme citado na seção 2.1, a fóvea é a área de maior acuidade visual da retina, e tem um área correspondente a cerca de 1 grau do ângulo visual. Não é possível saber qual o ponto exato focado pelo olho dentro desse intervalo de 1 grau, que corresponde aproximadamente ao tamanho do polegar de um indivíduo visto por ele com seu braço estendido. Além disso, os movimentos microssacádicos realizados inconscientemente durante as fixações são um fator de introdução de ruído nos dados gerados pelo rastreador.

3.1.2 Identificação de fixações

Um rastreador de olhar que opere em tempo real fornece estimativas de pontos observados pelo usuário a intervalos de tempo pequenos. Suponha um sistema de rastreamento de olhar não-intrusivo que tenha capacidade de processar 30 quadros de vídeo por segundo; esse sistema forneceria a estimativa de um ponto observado pelo usuário a cada 33 milissegundos, aproximadamente. Esses dados deve passar por um processo de filtragem para obter informações semanticamente relevantes sobre o processo visual. Em particular, os dados provenientes do rastreador devem ser agrupados em unidades que identifiquem as fixações e os movimentos sacádicos. Este é o primeiro passo para obter informação sobre o ponto de interesse do usuário.

A estratégia mais simples para implementar essa filtragem é o uso de um limiar espacial e outro temporal – somente ocorrências de pontos cuja distância entre si seja menor que um dado valor, e dentro de um intervalo mínimo de tempo são agrupados e considerados como componentes de uma mesma fixação [Jac90, SJ00, ZMI99]. Estes valores são definidos por uma combinação do prévio conhecimento da duração temporal média dos eventos do

processo visual com testes empíricos. Uma estratégia mais sofisticada é apresentada por Salvucci e Goldberg em [Sal99a]; essa estratégia é baseada no uso de uma cadeia de Markov oculta (HMM) com dois estados: *fixação* e *movimento sacádico*. Essa modelagem permite a reestimação dos parâmetros correspondentes aos limiares temporal e espacial. Em tese, essa reestimação permite o ajuste fino dos parâmetros, de forma a obter uma melhor detecção das fixações. Porém esse processo é computacionalmente mais complexo, e atualmente só é usado para análise de dados *a posteriori* [SG00].

3.1.3 Seleção e o problema do “Toque de Midas”

A estratégia mais divulgada de aplicação do rastreamento de olhar às interfaces já existentes é usar o olhar para executar tarefas de apontamento e seleção. Essa idéia está baseada em algumas evidências. Glenstrup e Engell-Nielsen [GEN95] argumentam sobre numerosas evidências que há relação entre o interesse dos indivíduos e o que eles estão olhando. Outra fonte de evidência é o modelo construído por Card, Moran e Newell [CMN80], que prevê o tempo T gasto para a execução de tarefas de apontamento com o uso de dispositivos tradicionais, como o *mouse*. A equação seguinte mostra uma versão simplificada do modelo:

$$T = T_M + T_V + T_P + T_R \quad (3.1)$$

Nele, a subtarefa motora (isto é, efetivamente mover o dispositivo para a posição correta), que leva tempo T_P , é precedida por uma tarefa cognitiva que leva tempo T_M e por uma tarefa visual (isto é, a busca visual do alvo em questão) que leva tempo T_V . T_R é o tempo de resposta do sistema. Hoje, apesar das evidências de que esse modelo não é preciso [HK99], ele ainda traz uma estimativa razoável e justifica os esforços no sentido de aplicar o rastreamento de olhar para a execução de tarefas de apontamento.

Porém, o mapeamento direto do olhar (mais especificamente, das fixações) a um comando de seleção do sistema cria o problema identificado por Jacob [Jac90] e chamado de “*Toque de Midas*” – uma seleção pode ser ativada em qualquer posição da tela observada pelo usuário, tenha ele a intenção de fazê-la ou não. Após a filtragem dos dados do rastreador de olhar, um desafio no projeto de uma técnica de interação é evitar o problema do Toque de Midas, implementando mecanismos para que o usuário indique quando deseja executar um comando de seleção. A primeira abordagem a esse problema foi a implementação de um tempo de latência – a seleção de um objeto é efetuada somente após um intervalo de tempo. Alternativamente, a seleção pode ser ativada manualmente pelo pressionamento de um botão pelo usuário. O MAGIC Pointing [ZMI99] traz uma abordagem alternativa que incorpora o rastreamento de olhar ao apontamento tradicional e será descrito em maiores detalhes mais adiante.

3.1.4 Modelagem do contexto e interfaces prestativas

O estudo de registros gerados por rastreadores de olhar tem sido importante para o campo da psicologia cognitiva, pois eles fornecem evidência sobre a forma com que indivíduos buscam informação visual para resolver problemas, quanto tempo gastam para analisar essa informação e quando precisam voltar à informação previamente acessada [Sal99b].

Ao incorporar o olhar e outras informações sobre o estado do usuário no processo de interação com sistemas computacionais cria-se um potencial para que as interfaces, dispostas de mais informação sobre o estado da atenção do usuário, não fiquem mais o tempo todo passivas à espera de um comando, mas se tornem *prestativas*¹, ou seja, possam se adaptar dinamicamente às necessidades do usuário a cada momento.

Vertegaal [Ver02], em sua discussão sobre interfaces prestativas, identifica que para atingir esse objetivo tais interfaces constroem um modelo do estado passado, presente e futuro da

¹Nossa tradução do inglês *attentive interfaces*

atenção do usuário, levando-se também em consideração a disponibilidade de recursos do sistema – ou seja, um modelo de informações que fazem parte do *contexto* da interação. O conceito de interfaces prestativas vem ampliar a proposta de interfaces perceptivas de Turk [Tur98] – além da perceber o estado do usuário, a interface também modela as possíveis ações que podem ser tomadas e dinamicamente sugere a tomada da ação mais apropriada.

Recentemente, sistemas computacionais de rastreamento de olhar de forma não-intrusiva tem sido empregados no desenvolvimento de algoritmos de *interpretação*² de movimentos do olhar. Um algoritmo de interpretação toma com entrada uma seqüência de ações do usuário definidas em um certo protocolo³ e devolve uma estimativa, baseada em um modelo previamente construído, de qual é a intenção do usuário ao tomar essas ações.

O projeto de técnicas de interação baseadas em rastreamento de olhar pode se beneficiar do uso de algoritmos de interpretação ([Sal99a], [SA00], [Edw98]). Uma vez que seja possível definir um modelo cognitivo da interface, as estimativas devolvidas por um algoritmo de interpretação servem como parâmetros de ajuste para as operações a serem executadas. No caso da aplicação do rastreamento de olhar a tarefas de apontamento, um algoritmo de interpretação pode ajudar a definir qual objeto o usuário tem a real intenção de selecionar, compensando as imprecisões inerentes ao olho humano (veja seção 3.1.1) e ruído gerado pelo processo de rastreamento.

²Nossa tradução do inglês *tracing algorithms*, usada aqui para evitar a confusão com a expressão *rastreamento* de olhar (*eye tracking*), usada para se referir ao processo de obter o ponto observado pelo usuário ou sua linha de visão

³A idéia de protocolo implica a filtragem semântica dos dados do dispositivo de entrada rastreador das ações; no caso do rastreamento de olhar aplicado a interfaces teríamos como componentes desse protocolo as fixações e os movimentos sacádicos

3.2 Técnicas de interação implementadas

Vamos agora exibir algumas técnicas de interação descritas na literatura. O objetivo dessa seção não é enumerar exaustivamente as aplicações do rastreamento de olhar a interfaces, mas ilustrar através de alguns dos trabalhos mais relevantes na área os conceitos de projeto expostos na seção anterior.

A partir da forma como o rastreamento de olhar é empregado podemos tentar organizar as suas aplicações em categorias. Duchowski [Duc02] as classifica em dois grandes grupos: *de diagnóstico* e *interativas*. Aplicações de diagnóstico são aquelas onde os dados provenientes do processo de rastreamento são armazenados e processados posteriormente para o estudo dos padrões de atenção do usuário dado um determinado estímulo. Tais aplicações tipicamente não mudam seu estado (em particular, o estímulo apresentado) de acordo com a mudança da atenção visual do usuário.

O segundo grupo, das chamadas aplicações *interativas*, é o foco de interesse dos exemplos desta seção. Tais aplicações reagem à posição observada pelo usuário, e a forma como reagem as divide em duas subcategorias:

seletivas: são as aplicações que usam a informação do ponto observado pelo usuário para selecionar alvos da interface, de forma análoga a um dispositivo de apontamento como o *mouse*. Exemplos de sistemas desta categoria são o protótipo descrito por Jacob (seção 3.2.3) e o MAGIC Pointing (seção 3.2.4).

de contingência do olhar: são as aplicações que usam a informação do ponto observado pelo usuário para redistribuir a informação exibida de forma a otimizar a renderização de *displays* complexos ou o uso de um canal de comunicação com largura de banda limitada. Podem ser classificadas nesta categoria o Mundo das Janelas de Bolt (seção 3.2.1), o ambiente interativo do Pequeno Príncipe de Starker e Bolt (seção 3.2.2) e o

sistema de renderização de simulações de O’Sullivan, Dingliana e Howlett (seção 3.2.5).

3.2.1 *Mundo das janelas*

Uma das aplicações pioneiras do rastreamento de olhar é o *mundo das janelas* descrito por Bolt [Bol81]. O mundo das janelas constitui-se de um *videowall* que exhibe diferentes seqüências de vídeo e, de acordo com a atenção visual recebida, as janelas correspondentes às seqüências aumentam ou diminuem; o som da seqüência que recebe maior atenção visual também tem seu volume aumentado. É uma das primeiras tentativas de inferir a intenção do usuário através do olhar em um sistema interativo que se tem notícia.

3.2.2 *Pequeno príncipe*

Glenstrup e Engell-Nielsen [GEN95] descrevem em seu trabalho um sistema desenvolvido por Starker e Bolt (1990) que conta, em uma versão animada, a história “O Pequeno Príncipe”, de Antoine de Saint Exupery. A história é contada em um ambiente (o pequeno planeta habitado pelo príncipe) que contém vários objetos, tais como vulcões, escadas e flores.

O espectador é monitorado por um rastreador de olhar baseado na reflexão de luz pela córnea; quando é detectada a atenção do espectador sobre algum objeto do ambiente, o sistema exhibe maiores informações sobre o objeto. Essa detecção é feita através da contagem do número de fixações feitas sobre o objeto associada com o tempo de duração das fixações. É feita também uma tentativa rudimentar de inferência do contexto: os objetos são agrupados em categorias, e quando se detecta que a atenção está distribuída de forma aproximadamente igual por vários objetos de uma mesma categoria, o sistema exhibe informações sobre a categoria de objetos.

3.2.3 Seleção por tempo de latência

Jacob [Jac90] descreve os protótipos de algumas técnicas de interação usando rastreadores de olhar. Seu trabalho levanta a questão do “Toque de Midas” e implementa duas alternativas para a seleção: o usuário pode pressionar um botão indicando a intenção de selecionar o objeto observado ou pode aguardar um tempo de latência razoavelmente curto (definido entre 150 e 250 milissegundos). Ambas as alternativas são oferecidas em paralelo, e é reportado que em experiências preliminares o tempo de latência se mostrou mais conveniente.

O protótipo baseia-se em uma tela onde são apresentados vários navios do lado direito, e uma área para exibição de texto do lado esquerdo (Figura 3.2.3). Quando um navio é selecionado da maneira descrita acima, informações sobre o mesmo são exibidas. Foi implementada também a operação de “arrastar-e-soltar” navios: uma vez selecionado um navio, o usuário pressiona um botão e olha para o lugar onde deseja posicionar o navio, soltando então o botão. A seleção usando o olhar também foi empregada no protótipo para a seleção de menus do tipo “pull-down”; nesse caso a estratégia do tempo de latência curto não se mostrou eficiente – o autor especula que isso acontece porque o tempo necessário para a leitura de itens do menu que não são familiares ao usuário é falsamente interpretado como o tempo de latência para a seleção. O aumento do tempo de latência também parece não ser razoável, pois gera a necessidade de manter a atenção visual sobre um item de menu durante um intervalo de tempo pouco natural.

Esse protótipo foi avaliado com maior rigor por Sibert e Jacob [SJ00] através de testes com usuários; mostrou-se que tarefas de apontamento e seleção simples foram realizadas através do rastreamento de olhar com tempo de latência de 150 milissegundos em menor tempo em relação ao apontamento e seleção usando-se somente o *mouse*.

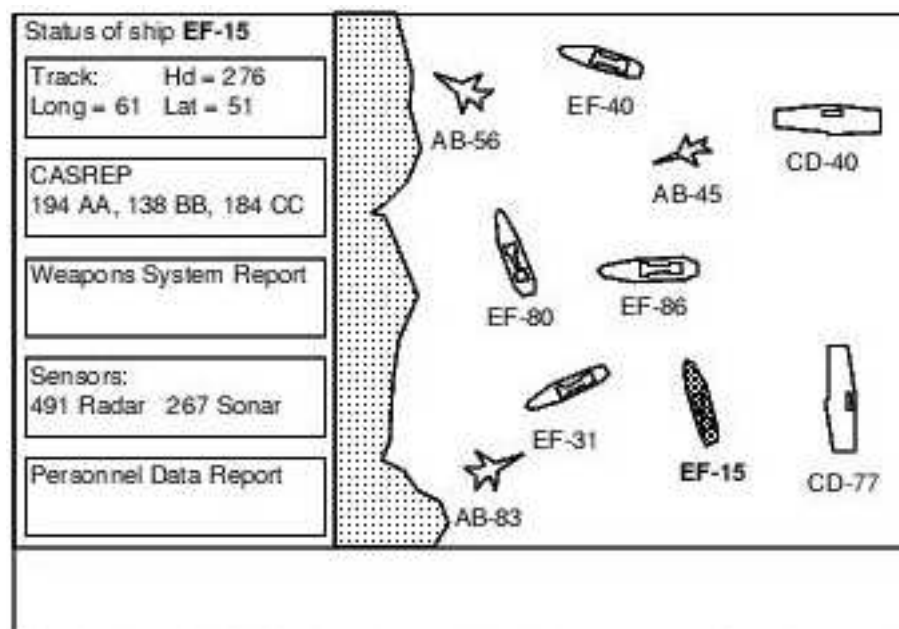


Figura 3.1: Seleção usando tempo de latência (conforme exibido em [SJ00])

3.2.4 MAGIC Pointing

O MAGIC Pointing [ZMI99, Zha03] de Zhai, Morimoto e Ihde trouxe uma nova abordagem à forma de executar tarefas de apontamento auxiliadas por rastreamento de olhar. Levando em consideração as limitações na precisão causadas pelo processo de rastreamento e a calibragem do sistema, propõem uma técnica de interação que combina o rastreamento de olhar com um dispositivo de apontamento convencional, como um *mouse* ou um *trackball*. A posição da tela indicada pelo rastreamento de olhar é considerada uma estimativa da posição observada pelo usuário, sujeita a erro em algum grau. Nesse momento o usuário ajusta a posição do cursor através do controle motor do dispositivo tradicional, obtendo maior precisão.

O cursor acompanha o olhar quando a atenção visual se volta para um alvo distante da posição atual do cursor. Duas estratégias diferentes de reposicionamento são implementa-

das: na primeira (chamada estratégia *conservadora*) o reposicionamento é ativado a partir de um movimento motor iniciado pelo usuário no dispositivo tradicional; na segunda (chamada estratégia *liberal*) o movimento motor não é necessário.

Posteriormente, Wang, Zhai e Su [WZS01] desenvolveram uma aplicação do MAGIC Pointing a um domínio específico: a digitação de ideogramas do idioma chinês através do *pinyin*, uma representação fonética usando caracteres do alfabeto ocidental. O problema surge com a grande quantidade de ideogramas que correspondem a uma mesma representação fonética, exigindo a eliminação de ambiguidades de alguma forma. Nesse sistema isso é feito através de botões que exibem os caracteres correspondentes à representação fonética que acabou de ser digitada pelo usuário. A seleção dos botões é feita usando o MAGIC Pointing, e é empregada uma estratégia para atenuar o efeito da precisão limitada do rastreamento de olhar: os botões não são dispostos em filas ou colunas, mas em um formato de 'W' (Figura 3.2.4), de forma a maximizar a distância entre os alvos.

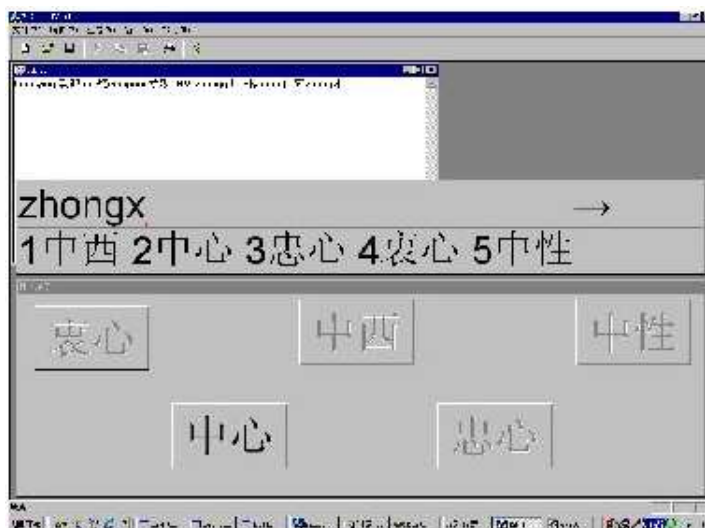


Figura 3.2: Digitação de ideogramas do idioma chinês através do *pinyin* [WZS01]

3.2.5 Renderização de simulações em tempo real

O’Sullivan, Dingliana e Howlett [ODH02] descrevem um sistema onde o rastreamento de olhar é usado para otimizar o uso de recursos computacionais aplicados à renderização de colisões entre objetos complexos.

Renderizar objetos em tempo real para aplicações como jogos e sistemas de realidade virtual (VR) é uma tarefa computacionalmente complexa, e muitas vezes é necessário renderizar alguns objetos com um nível de detalhamento (LOD - *level of detail*) menor para manter uma taxa de redesenho aceitável para o display. Outra estratégia para atingir esse objetivo é manter modelos em vários níveis de detalhamento para o tratamento e renderização de colisões entre objetos. Em [ODH02] os autores utilizam o rastreamento de olhar para determinar a região do display onde está focada a atenção visual do usuário, e a renderização de colisões nessa região terá alocada uma parcela maior de tempo dos recursos computacionais.

Um teste foi conduzido, onde usuários classificaram seqüências de animações de colisões entre dois objetos de acordo com a qualidade da renderização, como “boas” ou “más”, de acordo com critérios aprendidos em uma sessão de treinamento onde foram expostos a animações com maior ou menor nível de detalhamento. Os resultados mostraram que, quando as seqüências onde o nível de detalhamento das colisões foi modificado dinamicamente de acordo com as informações do rastreamento de olhar, a qualidade da renderização percebida pelos usuários foi maior.

3.2.6 Modelagem do contexto

Como comentado na seção 3.1.4, um método de compensar a imprecisão do processo de rastreamento que tem sido empregado recentemente é incorporar mais informação sobre o *contexto* do diálogo em execução entre o usuário e o computador. Salvucci descreve em [SA00]

uma interface, denominada IGO (*Intelligent Gaze-added Operated Interface*) que simula um desktop comum de um ambiente WIMP (Windows, Menus, Icons and Pointing) que pode ser operado através do olhar. A seleção é efetuada através do pressionamento de um botão no teclado, de maneira semelhante ao descrito em [Jac90]; porém aqui a funcionalidade do apontamento tradicional é mantida, de forma que o apontamento pelo olhar seja uma forma adicional de interagir com o sistema. As fixações identificadas pelo sistema de rastreamento são mapeadas nos objetos através de um modelo probabilístico: dada uma fixação g , o item da interface i_{best} no qual a fixação será mapeada é aquele que maximiza a probabilidade $P(i|g)$, onde i representa o evento “selecionar o objeto i ”. Daí temos:

$$\begin{aligned} i_{best} &= \operatorname{argmax}_{i \in I} (P(i|g)) \\ &= \operatorname{argmax}_{i \in I} \left(\frac{P(g|i)P(i)}{P(g)} \right) \\ &= \operatorname{argmax}_{i \in I} (P(g|i)P(i)) \end{aligned}$$

$P(g|i)$, a probabilidade de uma fixação (x, y) dada a intenção de se selecionar o item i é modelada através de uma distribuição gaussiana bidimensional centrada no item. No cálculo de $P(i)$ é onde especificamente é incorporada a informação de contexto da tarefa em execução atualmente. Cada um dos itens da interface recebe uma pontuação (posteriormente normalizada na escala de probabilidades) de acordo com a última tarefa executada, ou de acordo com algum estado da interface. Por exemplo, o botão “X” de fechamento de uma janela tem sua pontuação aumentada após a seleção de algum item na janela, considerando que o usuário tipicamente não fecha uma janela sem efetuar pelo menos uma seleção nela. O menu que contém o comando *Esvaziar lixeira* tem sua pontuação aumentada quando há itens na lixeira, e assim por diante.

Em [Sal99a] é desenvolvida uma modelagem mais complexa do contexto de uma aplicação

que implementa a digitação de palavras olhando para um teclado exibido no vídeo. Para um dado conjunto de palavras pré-definido, é construída uma gramática (Tabela 3.2.6) que em suas regras de produção modela os passos cognitivos que podem ser tomados a partir de um dado estado. No exemplo é exibida uma gramática completa que modela os passos cognitivos para a digitação das palavras *rat*, *trap* e *part*.

Há três possibilidades para a construção da gramática, enumeradas em ordem crescente do grau de informação seqüencial que possuem:

- **gramática de primeira ordem:** representa transições de letra para letra;
- **gramática de segunda ordem:** representa transições de pares de letras para letras;
- **gramática completa:** representa cada palavra separadamente.

start	→	word rat
start	→	word trap
start	→	word part
rat	→	R A T end
trap	→	T R A P end
part	→	P A R T end
end	→	out

Tabela 3.1: Exemplo de gramática completa: passos cognitivos para digitação de palavras

O usuário inicia a tarefa selecionando o botão **word**, que mostra a palavra a ser digitada, e encerra a digitação selecionando o botão **out**, representados por seus nomes na gramática. A partir da gramática é construída uma cadeia de Markov oculta (HMM) que segue a estrutura da gramática dos passos cognitivos e tem como subunidades fundamentais de construção cadeias de Markov ocultas para a identificação das fixações em cada alvo (no caso, em cada letra do teclado virtual). Uma vez montado esse modelo, um protocolo de fixações e movimentos

sacádicos obtido a partir do rastreador de olhar é mapeado na seqüência mais provável de estados usando o Algoritmo de Viterbi [Rab89].

Temos aí duas abordagens razoavelmente diferentes ao uso do contexto em interfaces que incorporam o rastreamento de olhar. A primeira tem uma estrutura e uma implementação mais simples, além de possibilitar de forma imediata a modelagem da mudança de estados da interface; ela exige, por outro lado, a codificação manual prévia das regras de pontuação que geram as probabilidades de seleção de itens. Note também que, para efeito de discussão, a interface exibida em [SA00] é uma simulação muito simplificada de uma interface WIMP real. A segunda apresenta um modelo mais preciso dos passos cognitivos, porém o desempenho da interpretação em tempo real do modelo decresce à medida que se incorpora mais informação seqüencial na gramática e o número de palavras aumenta, o que parece limitar a incorporação de modelagens de contexto muito detalhadas em sistemas interativos. Essa informação carece de uma avaliação mais precisa, já que [Sal99a] não traz referências quanto ao *hardware* usado na implementação do sistema.

3.3 Análise comparativa

Várias técnicas de interação baseadas em rastreamento de olhar foram desenvolvidas considerando o olhar como uma estimativa direta do objeto de interesse. Em outras palavras, tais técnicas (pertencentes à categoria das aplicações interativas seletivas de Duchowski [Duc02]) foram projetadas considerando que, se o usuário está olhando para um objeto, essa informação é suficiente para considerar que ele deseja tomar alguma ação sobre aquele objeto. É o que acontece no sistema interativo do Pequeno Príncipe (seção 3.2.2 e posteriormente no protótipo de Jacob (seção 3.2.3). Mesmo identificando e reconhecendo a existência do problema do “Toque de Midas” e propondo a solução do tempo de latência, O trabalho de Jacob, como uma

das primeiras abordagens à implementação de tarefas de seleção usando o olhar, identifica que a estratégia do tempo de latência não é bem sucedida em todos os casos de uso que surgem em um protótipo mais próximo do funcionamento de interfaces reais. Por exemplo, em tarefas que exigem leitura ou exploração visual mais detalhada, o tempo de latência deve ser mantido alto para evitar seleções não-intencionais. A deficiência dessas técnicas está em não considerar com a devida importância a própria natureza exploratória do processo da visão, que está em ação a todo momento, mas é mais evidente quando há a necessidade de executar tarefas complexas.

Uma técnica de interação mais recente que ainda usa o olhar diretamente para tarefas de seleção é o IGO de Salvucci [SA00]. Nesse caso a intenção de selecionar é indicada através do pressionamento de uma tecla. Porém, a interface de testes é toda projetada para minimizar a imprecisão do processo de rastreamento – os objetos selecionáveis são bem maiores do que o usual em interfaces WIMP, e a informação de um contexto restrito é empregada para eliminar eventuais ambiguidades. De qualquer forma, uma técnica de interação que associa a seleção de objetos diretamente ao interesse visual acaba por sobrecarregar a visão com uma tarefa não natural, que é usá-la para atuar sobre o ambiente à nossa volta. Apesar dos problemas, essa estratégia parece justificável em casos particulares em que o usuário não poderia interagir com o sistema de outra forma, como o de pessoas acometidas por tetraplegia ou doenças degenerativas [HJM⁺89, Edw98].

Em outra linha de desenvolvimento estão as interfaces que empregam o rastreamento de olhar como um canal de entrada passivo, usado para redistribuir dinamicamente a informação exibida (ou aplicações interativas de contingência de olhar [Duc02]). Nessa linha podemos classificar o Mundo das Janelas de Bolt (seção 3.2.1) e mais recentemente as técnicas de renderização de simulações (seção 3.2.5). Essas aplicações do rastreamento de olhar tem menos dificuldades a contornar quanto à precisão do rastreamento ou à tomada de decisão sobre quando o usuário deseja ou não tomar uma ação no objeto que recebe sua atenção

visual – uma estimativa menos precisa da região observada já é suficiente para redimensionar a informação exibida [ODH02].

Conhecidos os problemas enfrentados e as características de ambas as linhas de desenvolvimento poderíamos concluir que apenas as aplicações interativas de contingência de olhar tem condição de se adaptar melhor à tecnologia de rastreamento atual e às características do processo de visão. O MAGIC Pointing (seção 3.2.4), entretanto, teve sucesso em implementar uma técnica de interação para a execução de tarefas de apontamento que associa o melhor das duas linhas de desenvolvimento descritas acima: ele inova no sentido de tratar o olhar como um canal de entrada passivo, trazendo uma estimativa da região de interesse do usuário. A atuação continua tendo o componente motor, e diminui-se o esforço cognitivo de explicitamente atuar sobre o ambiente com o olhar.

Capítulo 4

Proposta deste trabalho

Frente às vantagens potenciais e às deficiências atuais da tecnologia de rastreamento de olhar, nossa proposta consiste em incorporar a informação proveniente do rastreamento de olhar ao processo de interação com interfaces do tipo WIMP para a realização de tarefas de apontamento. A nossa implementação e testes tem como objetivo investigar as implicações da aplicação do rastreamento de olhar a esse paradigma de interação; buscamos identificar as situações em que o rastreamento de olhar pode tornar o processo de interação *melhor* para o usuário – melhor tanto em aspectos objetivos, como a velocidade ou precisão na realização das tarefas, quanto em aspectos subjetivos como a sua satisfação.

Em particular, queremos avaliar a necessidade e a importância de incorporar ao processo de interação, juntamente com o rastreamento de olhar, informação sobre o contexto atual da interação. A combinação de duas fontes de informação adicional, uma sobre o estado do usuário e outra sobre a tarefa em curso, aproxima as técnicas de interação que desenvolvemos do paradigma de interfaces prestativas.

A técnica de interação escolhida como ponto de partida para nosso trabalho é o MAGIC Pointing, descrito na seção 3.2.4. Sua estratégia inova ao tratar o olhar como um canal de

entrada mais passivo do que outras técnicas que tentaram atingir o mesmo objetivo, e assim começa a aproximar-se do conceito de uma interface prestativa [Zha03]. Sendo o MAGIC Pointing a técnica encontrada que melhor se adapta ao objetivo de nossas investigações, a partir dele propomos nossas extensões. A escolha do paradigma WIMP como domínio no qual basear o projeto de nossa interface justifica-se pela sua ampla disponibilidade nos sistemas computacionais hoje em dia, assim nossas extensões podem ser propostas sobre um padrão já bem estudado e conhecido, que permite avaliar com um grau maior de confiabilidade a eficiência dessas extensões.

Na seção 4.1 a implementação do MAGIC Pointing é descrita com maiores detalhes; na seção 4.2 descrevemos a primeira proposta para incorporação de informação de contexto no sistema, e na seção 4.3 vamos descrever e comentar a aplicação e os resultados de um teste com usuários.

4.1 Implementação do MAGIC Pointing

O MAGIC Pointing utiliza os dados fornecidos pela nossa implementação do rastreador de olhar desenvolvido por Morimoto *et al.* [CKAF00]. O rastreador baseia-se no reconhecimento da pupila e da primeira imagem de Purkinje (brilho ocular) através da iluminação do olho por dois conjuntos de *leds* infra-vermelhos (Figura 4.1). O primeiro conjunto é posicionado próximo ao eixo óptico da câmera, de forma a obter uma imagem iluminada da pupila através da reflexão da luz pelo fundo do olho, uma técnica que mencionamos na seção 2.2.2. O segundo conjunto é afastado do eixo óptico e produz o brilho ocular. Os conjuntos de luzes brilham alternadamente, de forma a obter em seqüência imagens do olho com a pupila clara e escura. A diferença entre essas duas imagens produz uma terceira imagem (Figura 4.2) onde pode facilmente ser obtida a segmentação da pupila.



Figura 4.1: Câmera e conjuntos de *leds* infravermelhos

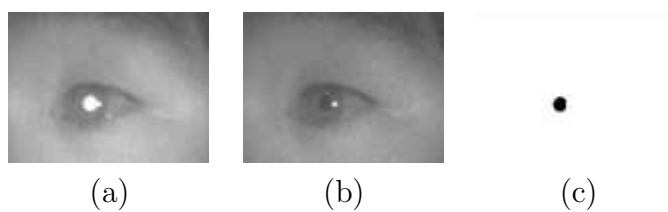


Figura 4.2: Pupila iluminada (a), pupila escurecida (b), e diferença entre as imagens (c)

As posições do centro da pupila e do centro do brilho ocular são rastreadas para o processo de calibragem: nele, o usuário é solicitado a olhar para nove pontos sobre a tela. Para cada ponto da tela, que denominaremos (x_t, y_t) , é obtida na imagem de vídeo o vetor entre o centro do brilho ocular e o centro da pupila, que denominaremos (x_v, y_v) . Define-se então uma transformação de segunda ordem entre (x_t, y_t) e (x_v, y_v) da seguinte forma:

$$\begin{aligned}x_t &= a_0 + a_1x_v + a_2y_v + a_3x_vy_v + a_4x_v^2 + a_5y_v^2 \\y_t &= a_6 + a_7x_v + a_8y_v + a_9x_vy_v + a_{10}x_v^2 + a_{11}y_v^2\end{aligned}$$

O par de equações acima é gerado para cada um dos 9 pontos, produzindo finalmente um sistema sobredeterminado de 18 equações e 12 incógnitas a_0, a_1, \dots, a_{11} , que são então determinadas através da minimização do erro quadrático. Durante o uso do rastreador, a cada quadro de vídeo capturado é obtido um vetor (x'_v, y'_v) ao qual é aplicada a transformação de segunda ordem com os parâmetros calculados durante a calibragem para obter um ponto (x'_t, y'_t) na tela.

O rastreador é capaz de processar cerca de 30 quadros de vídeo por segundo, e nessa frequência fornece estimativas de pontos observados pelo usuário sobre a tela. A primeira filtragem dos dados pelo MAGIC Pointing tem como objetivo identificar fixações e movimentos sacádicos: quando o rastreador fornece dois pontos consecutivos suficientemente próximos na tela¹, é considerado que houve uma fixação.

Quando é verificada a ocorrência de uma fixação, o cursor pode então potencialmente ter sua posição alterada pelo MAGIC. Na estratégia *conservadora*, o cursor só é reposicionado para a vizinhança da posição observada quando o usuário explicitamente faz um movimento

¹atualmente, essa distância é definida empiricamente como 70 pixels

com o dispositivo de apontamento convencional. A direção desse movimento inicial feito pelo usuário é utilizada para posicionar o cursor como se ele estivesse se movendo “em direção” à posição observada; por exemplo, se o movimento inicial com o *mouse* é feito para cima, o cursor aparecerá ligeiramente abaixo da posição observada. Na estratégia *liberal*, a diferença é que o movimento inicial com o dispositivo de apontamento não é necessário.

Em ambas as estratégias, para que o cursor seja reposicionado, algumas condições devem ser satisfeitas:

- o botão esquerdo do dispositivo de apontamento não deve estar sendo pressionado. Nas interfaces WIMP, o pressionamento do botão esquerdo juntamente com a movimentação indica a execução de uma operação de arrastar-e-soltar, que exige um ajuste mais preciso da posição do cursor.
- a posição observada pelo usuário deve estar suficientemente afastada² da posição atual do cursor. Uma posição observada próxima da posição atual do cursor sugere que o usuário está executando alguma tarefa que exige manipulação mais precisa, como desenho, onde novamente o ajuste fino manual da posição do cursor é mais adequado.
- o último reposicionamento do cursor feito manualmente pelo usuário ou automaticamente pelo MAGIC deve ter ocorrido há pelo menos 200 milissegundos. Essa condição busca minimizar a ocorrência do problema do “Toque de Midas” quando o usuário desloca por alguns instantes a atenção para uma posição da tela distante do cursor.

Alguns parâmetros citados, como a distância mínima entre pontos consecutivos que define uma fixação, são dinamicamente ajustáveis na implementação, e atualmente foram definidos através de ajustes empíricos.

²atualmente definido como pelo menos 10% da largura do monitor em pixels

4.2 Extensões do MAGIC Pointing

Para realizar as tarefas possíveis dentro das interfaces WIMP atuais, os usuários se habituaram a lidar com os chamados *widgets*, ou controles. Quando se decompõe uma tarefa usual nessas interfaces em subtarefas, é altamente provável que em algum ponto da decomposição várias subtarefas se caracterizem por “selecionar o widget X” ou “apontar o widget Y”.

Se grande parte do tempo que os usuários passam interagindo com interfaces WIMP é gasto apontando e selecionando *widgets* com um dispositivo de apontamento convencional, parece razoável incorporar ao MAGIC Pointing informação sobre o posicionamento e tamanho dos *widgets* como parte do contexto da interação. Assim, essa é a nossa primeira proposta: tendo essa informação disponível, e levando em consideração que o olho dificilmente é atraído por espaços vazios, o MAGIC reposiciona o cursor do *mouse* sobre o *widget* mais próximo da posição da tela observada.

A opção de apontar para o widget mais próximo foi implementada como uma opção adicional a cada uma das estratégias de posicionamento do cursor (liberal ou conservadora). O MAGIC Pointing mantém uma lista dos *widgets* disponíveis a cada momento e seu respectivo posicionamento e tamanho. Essa lista deve ser atualizada sempre que uma janela é redimensionada, movida, minimizada ou maximizada. Atualmente, apenas os widgets da janela ativa são armazenados nessa lista, para simplificar a implementação; usamos a hipótese que a seleção de *widgets* fora da janela ativa é um evento raro, e que usualmente é necessário trazer a janela em questão para o primeiro plano.

Nossa implementação provê duas alternativas para a construção da lista de *widgets*. Na primeira, o MAGIC Pointing detecta automaticamente quando uma nova janela é trazida para o primeiro plano ou a janela atual muda suas dimensões e, a partir do ID dessa janela, fornecido pelo sistema gráfico X Window [Inc04], percorre uma estrutura em árvore das chamadas *subjanelas* vinculadas a esse ID, obtendo a partir do ID dessas *subjanelas* o posicionamento e

tamanho dos *widgets*. Em nossos testes, verificamos que essa alternativa tem sucesso em obter os *widgets* de programas construídos com os *toolkits* Motif e Qt [AS04], mas não nos programas construídos com o GTK+ [Pro04]. Esse *toolkit* não faz uso dos IDs fornecidos pelo X Window para identificar os *widgets* que cria. Para suprir essa deficiência e prover maior flexibilidade, a segunda alternativa permite que uma aplicação construída ou adaptada para ser operada com o MAGIC Pointing informe a ele a posição dos seus widgets através de um mecanismo simples de chamada remota de procedimentos (RPC) denominado DCOP (Desktop COmmunications Protocol [BE04]). O DCOP é disponibilizado pelo ambiente KDE para intercomunicação entre suas aplicações, e foi desenvolvido para ser extremamente simples e portanto mais leve que outros mecanismos de aplicação mais geral como o CORBA (Common Object Request Broker Architecture), por exemplo.

4.3 Teste com usuários

Para explorar inicialmente o efeito da incorporação da informação de contexto, definimos e conduzimos um teste com usuários para avaliar a extensão definida na seção anterior em relação ao MAGIC Pointing original. O teste foi projetado com o objetivo de obter uma medida do desempenho dos usuários na execução de uma tarefa de apontamento e seleção em uma aplicação de testes em comparação com a mesma tarefa executada sem o auxílio do MAGIC Pointing. Para realizar essa comparação, nos baseamos no modelo proposto pela Lei de Fitts. Além disso, procuramos obter uma avaliação da satisfação subjetiva dos usuários com o novo método. As medições e registros de impressões dos usuários realimentarão um novo ciclo de refinamento das técnicas de interação.

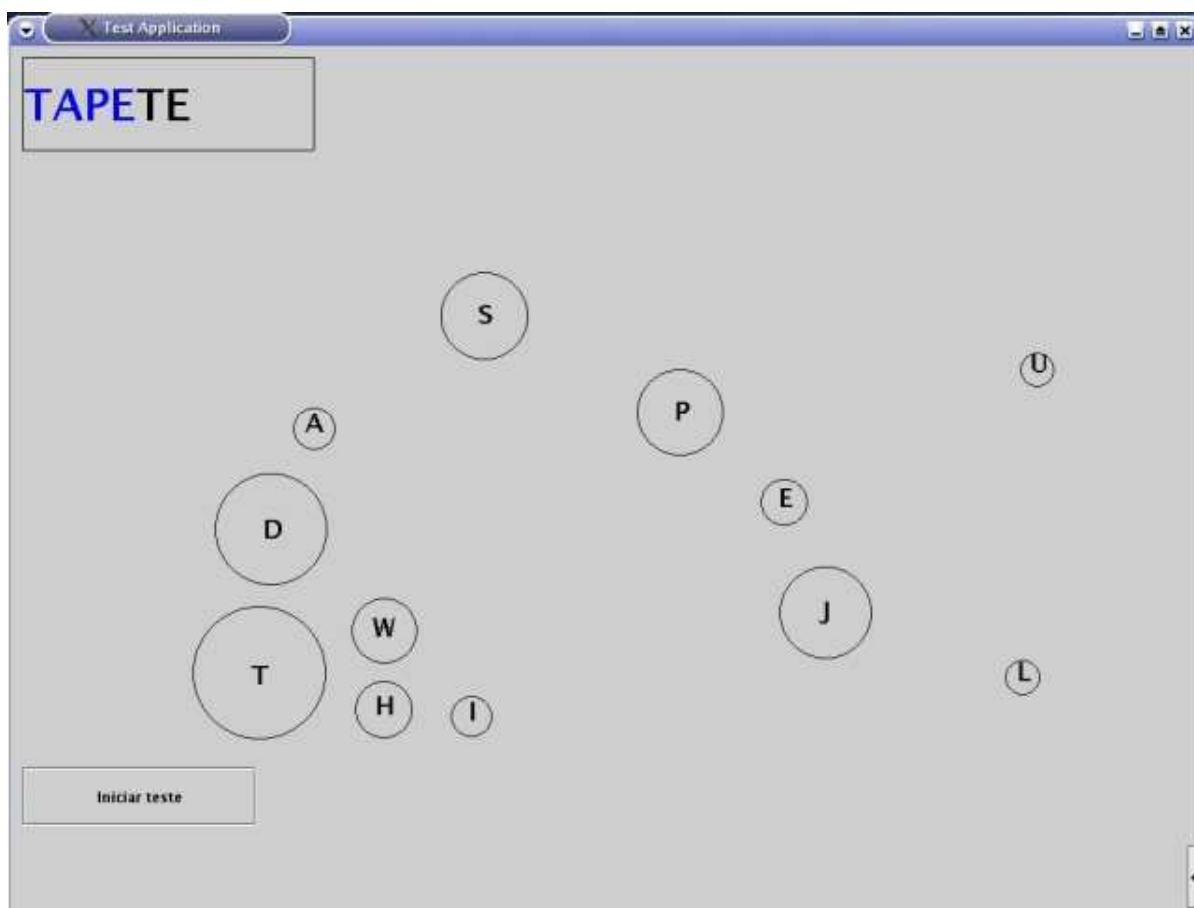


Figura 4.3: Aplicação de testes

4.3.1 Tarefa

Para a execução do teste, foi construída uma aplicação de testes na linguagem JAVATM. Em uma janela maximizada, são exibidos 12 botões circulares, cada um deles contendo uma letra do alfabeto. A tarefa do teste consiste em ativar botões em uma seqüência correspondente a uma palavra de 6 letras exibida no canto superior esquerdo da tela. No canto inferior esquerdo é disponibilizado também um botão para iniciar o teste, para efeito de início de medição do tempo de execução da tarefa (Figura 4.3).

À medida que o usuário ativa os botões corretos de acordo com a seqüência de letras da palavra, as letras já “digitadas” são exibidas em azul. No caso da ativação do botão errado, um *bip* sonoro é tocado e a letra correspondente ao botão que deveria ser ativado naquele momento é exibida em vermelho.

Preparação prévia das instâncias da tarefa

Para nosso teste, é desejável que tenhamos alvos de tamanhos suficientemente variáveis, cujas distâncias entre si sejam também variáveis, e um espalhamento não ordenado das letras atribuídas a cada botão. Isso acontece por dois motivos: primeiramente, queremos assegurar que o processo cognitivo necessário para cumprir a tarefa inclua, na máxima extensão possível, a busca visual do alvo a cada passo, simulando a execução de uma tarefa em um ambiente real. Além disso, uma maior variabilidade de tamanhos de alvos e distâncias entre alvos significa uma maior variabilidade dos parâmetros da Lei de Fitts. Buscamos obter, então, dados de melhor qualidade para os futuros testes de aderência a partir da definição do teste.

Assim, para cada palavra escolhida para ser uma instância do teste, são efetuados os seguintes passos de aleatorização:

1. se n_P é o número de letras distintas da palavra, $n_P \leq 6$, sorteiam-se aleatoriamente e com igual probabilidade outras $12 - n_P$ letras;
2. para cada botão b_i , $i = 1, \dots, 12$, vamos denotar por $d(b_i)$ o diâmetro de b_i . $d(b_i)$ é sorteada no intervalo $[0,87; 4,36]$ centímetros, correspondente ao intervalo de 1 a 5 graus de ângulo visual à uma distância de 50 cm do monitor. O intervalo foi escolhido de forma a simular a variabilidade de tamanhos de alvos em uma interface real.
3. uma localização (x, y) é escolhida aleatoriamente para cada botão dentro de uma área retangular de 30×15 centímetros localizada no centro da janela do teste.

Uma vez definidas as letras atribuídas a cada botão e as posições dos botões, estes parâmetros são aplicados toda vez que essa instância da tarefa for aplicada aos usuários.

4.3.2 Procedimento

O usuário participante do teste é recepcionado e informado sobre o que é teste e qual é seu objetivo. Ele é brevemente orientado sobre a aplicação de testes e sobre a tarefa que deverá ser realizada.

Então, é feito o processo de calibragem do rastreador de olhar. Para minimizar o erro instrumental, os usuários usam um apoio para queixo durante toda a duração do teste. É ressaltado para o usuário que não é ele que está sendo testado, mas o *design* de uma nova técnica de interação; o usuário é também orientado a executar a tarefa o mais rápido que puder.

Neste momento o usuário pode usar livremente o sistema por cerca de 5 minutos para ter um contato inicial com cada modalidade de controle do cursor do *mouse* auxiliado pelo rastreamento de olhar. Após esse período de prática livre, são realizadas várias sessões da tarefa, uma vez para cada modalidade de uso do MAGIC Pointing, dentre as combinações possíveis dos seguintes parâmetros:

- **Estratégia de ativação do posicionamento do cursor:** liberal ou conservadora;
- **Política de posicionamento do cursor:** apontamento simples, diretamente para a posição observada, ou apontamento para o círculo mais próximo (seção 4.2).

É também realizada uma sessão de controle, executada sem o uso do MAGIC Pointing (ou seja, com o uso do *mouse* apenas); o usuário continua usando o apoio para queixo durante a sessão de controle.

Em cada sessão o usuário executa a tarefa para 10 palavras. As mesmas palavras são utilizadas em todas as sessões, e sua ordem de apresentação é sorteada aleatoriamente. A ordem de execução das sessões também é sorteada; a única restrição feita é que as duas sessões com o uso da mesma estratégia de ativação do cursor (liberal ou conservadora) são executadas em seqüência.

Em cada execução da tarefa, são obtidas três medidas:

- tempo total para cumprimento da tarefa: obtido automaticamente pela aplicação, e computado desde a ativação do primeiro botão até a ativação do sexto e último botão;
- tempo entre a ativação de um botão e a ativação do próximo botão, em função da distância entre os alvos e tamanho do alvo de destino: para verificação da conformidade com a Lei de Fitts (vide seção 4.3.3);
- número de erros no cumprimento da tarefa: é computado um erro quando o usuário clica em um botão não correspondente à seqüência, ou quando ele clica fora de qualquer botão.

Após o término de todas as sessões, cada usuário responde um questionário (exibido no Apêndice A), graduando cada uma das afirmações seguintes em uma escala de 0 (discordo totalmente) a 5 (concordo totalmente):

- o MAGIC Pointing me auxiliou a executar a tarefa mais rapidamente
- usar o MAGIC Pointing é mais confortável do que usar apenas o *mouse*

O usuário também é solicitado a ordenar, de acordo com a sua preferência, cada uma das cinco modalidades que experimentou durante o teste.

4.3.3 Avaliação

O teste foi conduzido com 10 usuários, entre alunos e funcionários do Instituto de Matemática e Estatística da USP. O tempo médio de experiência dos usuários com o uso do *mouse* é de 9,4 anos.

Na primeira análise comparativa, agrupamos em uma média por usuário o tempo total de cumprimento das tarefas executadas para cada modalidade. Tarefas cumpridas com a ocorrência de erros foram descartadas do cálculo. A modalidade de uso foi significativamente influente na média do tempo total por usuário em uma análise de variância (ANOVA) para amostras pareadas ($F_{4,32} = 10,33$ com $p < 0,05$).

O tempo médio de cumprimento de uma tarefa foi de 6,264 segundos para o *mouse*, 9,767 segundos para a modalidade de apontamento conservadora ajustando para o círculo mais próximo e 10,115 segundos para a mesma modalidade sem o ajuste. A modalidade de apontamento liberal ajustando para o círculo mais próximo teve uma média de 10,491 segundos, enquanto que a mesma modalidade sem o ajuste teve uma média de 11,243 segundos.

Na amostra de usuários utilizada houve em média uma pequena vantagem para as modalidades que implementam o ajuste da posição do cursor do *mouse* para o círculo mais próximo em relação às mesmas modalidades sem o ajuste; porém essa vantagem não se mostrou estatisticamente relevante em testes t de Student pareados, $p < 0,05$. Uma análise das médias individuais sugere uma adaptação heterogênea dos usuários a essas modalidades: 55% dos usuários foram mais rápidos com a modalidade conservadora associada ao ajuste para o círculo mais próximo, enquanto que 54% dos usuários tiveram melhor desempenho na modalidade liberal.

O desempenho inferior dos usuários com o MAGIC Pointing em comparação ao uso apenas do *mouse* conflita com resultados de testes anteriores [ZMI99], especialmente em relação à modalidade liberal de apontamento. Analisamos três fatores que podem ter contribuído para esse resultado. Em primeiro lugar, o tempo de exposição inicial à técnica de interação pode

ter sido insuficiente para a aquisição da habilidade motora necessária para a nova técnica, especialmente a coordenação do movimento do olhar com o controle manual do *mouse*. Além disso, como a tarefa envolvia a busca visual dos alvos a serem ativados, na modalidade liberal o cursor do *mouse* freqüentemente acompanhava essa busca, e não raro os usuários sentiam ter “perdido” a localização do cursor. Identificamos também freqüentemente a ocorrência de um problema citado por Salvucci e Anderson em [SA00]: muitos erros cometidos acontecem quando o usuário mantém a atenção visual em um botão no qual deseja clicar, mas antes de completar o clique desvia a atenção para a busca do próximo botão. Se nesse pequeno intervalo de tempo o cursor é desviado para o novo foco de atenção visual do usuário, a ativação do botão não é completada. Isso aconteceu especialmente na modalidade liberal, o que pode explicar o número de erros significativamente maior nessa modalidade: 47 na modalidade com o ajuste para o círculo mais próximo e 48 na modalidade sem ajuste, contra 22 na modalidade conservadora com ajuste e 13 na sem ajuste.

Estes resultados sugerem um refinamento mais detalhado nas condições usadas pelo MAGIC Pointing para o reposicionamento do cursor de forma a atenuar as dificuldades no controle do cursor encontradas pelos usuários em uma tarefa mais complexa do que a descrita em [ZMI99]; a tarefa e o procedimento do teste também serão revistos para investigar os efeitos da falta de aprendizagem da técnica de interação.

Dentre as modalidades do MAGIC Pointing testadas, as modalidades que ofereciam o ajuste para o círculo mais próximo foram inicialmente mais rápidas do que as modalidades correspondentes que não ofereciam o ajuste. Esse fato motiva a realização de investigações futuras para indicar se essa vantagem, embora pequena, justifica a incorporação de informações mais complexas sobre o contexto da interação na interface.

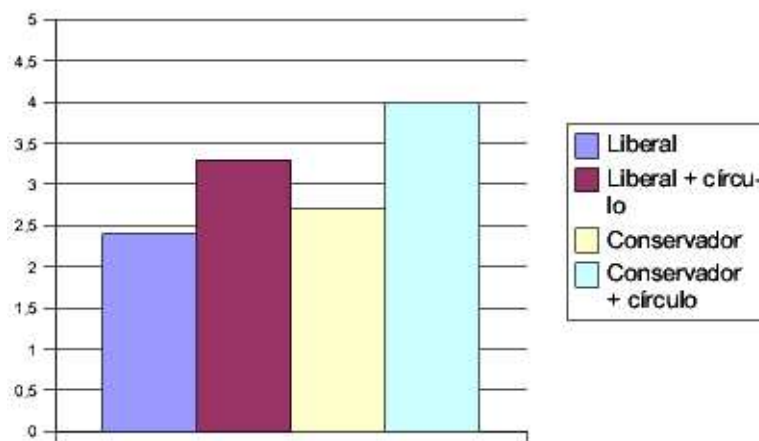


Figura 4.4: Avaliação dos usuários: rapidez da modalidade

Avaliação subjetiva

Os usuários participantes do teste atribuíram uma boa avaliação à estratégia de posicionamento conservadora; na escala de 0 a 5, sendo 5 o maior grau de concordância com a afirmação, a estratégia conservadora com ajuste para o círculo mais próximo recebeu uma média de 4 para a afirmação “Esta modalidade me ajudou a executar a tarefa mais rapidamente” e uma média de 3,4 para a afirmação “Esta modalidade é mais confortável do que usar apenas o *mouse*” (figuras 4.4 e 4.5)

Na classificação da modalidade preferida, a modalidade conservadora obteve um posicionamento médio igual ao do *mouse*: 2,4. Com base nas impressões deixadas pelos usuários após o teste, analisamos que um dos motivos da avaliação inferior dada pelos usuários à modalidade liberal foi a sensação de “perda de controle” do cursor citada anteriormente.

Modelagem pela Lei de Fitts

Uma medida usual do desempenho de dispositivos de apontamento é dada pela chamada *Lei de Fitts*. Proposta por P.M. Fitts na década de 50 [Fit54], ela fornece uma estimativa do

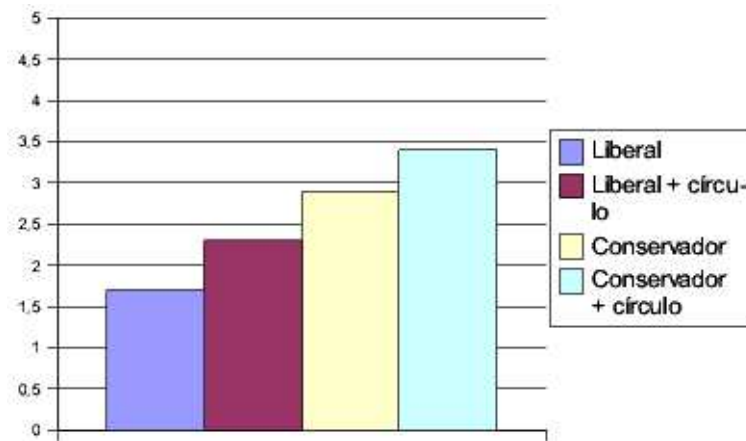


Figura 4.5: Avaliação dos usuários: conforto da modalidade

Modalidade	Classificação média
<i>Mouse</i>	2,4
Conservador com círculo mais próximo	2,4
Conservador	3,2
Liberal com círculo mais próximo	3,3
Liberal	3,7

Tabela 4.1: Modalidade preferida: de 1 (para a melhor) a 5 (para a pior)

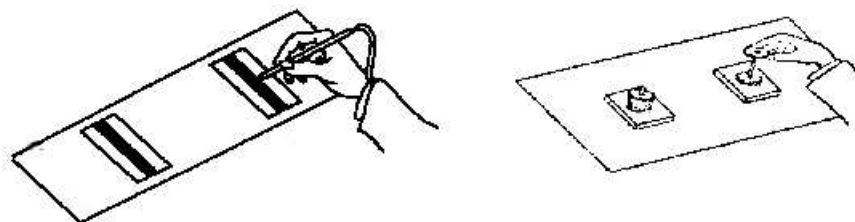


Figura 4.6: Ilustração de dois testes propostos por Fitts

tempo necessário para uma pessoa executar um movimento manual em direção a um alvo em função da distância do alvo e o tamanho do mesmo. Pode ser expressa através da equação

$$T = a + b \log_2(2A/W) \quad (4.1)$$

onde T é o tempo de execução do movimento, A é a amplitude do movimento realizado e W é o tamanho do alvo. Os coeficientes a e b são constantes usualmente determinadas empiricamente através de regressão linear, podendo variar para tarefas em diferentes contextos. O termo do logaritmo $\log_2(2A/W)$ é por vezes chamado de *índice de dificuldade (ID)* da tarefa.

Um dos testes propostos originalmente por Fitts envolvia o toque de duas chapas metálicas dispostas paralelamente com uma caneta e obtenção do tempo necessário para tocar a segunda chapa após tocar a primeira; a amplitude A é tomada como a distância entre as chapas e W é tomado como a largura da chapa. Outros testes envolviam a transferência de discos entre pinos dispostos também paralelamente (Figura 4.6).

Um modelo validado em um contexto de aplicação à primeira vista tão diferente do contexto dos sistemas computacionais interativos que surgiriam nas próximas décadas foi aplicado com sucesso na previsão do tempo necessário para a execução de tarefas de apontamento usando-se variados dispositivos. MacKenzie, Sellen e Buxton [ISB91] propuseram a seguinte variação da formulação da Lei de Fitts, que permitiu em análise dos autores melhor aderência a dados

Modalidade	ID e T	A e T	W e T
<i>Mouse</i>	0,183	0,129	-0,131
Conservador + círculo mais próximo	-0,007	-0,004	0,009
Conservador	0,082	-0,018	-0,015
Liberal + círculo mais próximo	0,061	0,057	-0,047
Liberal	0,153	0,079	-0,145

Tabela 4.2: Coeficientes de correlação linear - parâmetros da Lei de Fitts

empíricos:

$$T = a + b \log_2(A/W + 1) \quad (4.2)$$

Nessa variação o índice de dificuldade $ID = \log_2(A/W + 1)$ é maior que zero por construção, e o inverso do coeficiente b é denominado *índice de performance (IP)*, medido em bits por segundo (ID é medido em bits devido ao logaritmo na base 2). O índice de performance pode ser usado para a comparação do desempenho de diferentes dispositivos de apontamento.

Em nosso experimento, desejamos verificar como as modalidades do MAGIC Pointing se comportam em relação ao *mouse* quanto à Lei de Fitts. Assim, para cada tarefa de cada modalidade, agrupamos o tempo de clique entre pares de botões em triplas (A, W, T) , representando respectivamente a distância entre os botões (ou seja, a amplitude do movimento), o diâmetro do segundo botão clicado e o tempo de clique. Primeiramente, para cada modalidade, calculamos o coeficiente de correlação linear entre $ID = \log_2(A/W + 1)$ e T , entre A e T e entre W e T , expostos na Tabela 4.3.3.

Verificamos, conforme esperado, valores que indicam correlação linear entre ID e T nos testes com o *mouse*. A modalidade liberal também mostrou uma tendência à correlação linear entre ID e T nos mesmos padrões do *mouse*, de forma mais acentuada na modalidade sem ajuste para o círculo mais próximo. Se por um lado esses resultados mostram (até em contradição com a natureza do projeto da modalidade) indícios que o tempo para cliques entre

círculos segue um padrão semelhante à modalidade puramente motora (*mouse*), por outro sabemos que ajustes imprecisos nos critérios de reposicionamento do cursor para a tarefa do teste nesta modalidade podem ter causado uma maior necessidade do ajuste motor da posição realmente desejada pelo usuário para o cursor, adicionando um componente motor à tarefa em maior grau do que o previsto.

As demais três modalidades (conservadora com e sem ajuste, e liberal com ajuste) apresentaram valores mais próximos de zero da correlação linear para ID , A e W do que o *mouse*, sendo a modalidade com os valores mais baixos a conservadora com ajuste para o círculo mais próximo. Esse é um indício que algumas modalidades do MAGIC Pointing podem não apresentar dependência linear entre o tempo de movimentação e a distância ou o tamanho do alvo – ou seja, não seguem o modelo tradicional para aquisição motora de alvos.

A partir do conjunto dos dados para cada modalidade e do cálculo do índice de dificuldade ID , obtivemos os coeficientes de regressão linear por mínimos quadrados da equação $T = a + b.ID$ para cada uma das modalidades. A modalidade conservadora com ajuste para o círculo mais próximo foi a única a apresentar um coeficiente negativo ($b = -29,75$) mas com valor absoluto pequeno em comparação com as outras modalidades. Com o ajuste dos dados aos coeficientes da Lei de Fitts notamos novamente uma relação não usual nessa modalidade entre o índice de dificuldade ID e o tempo T : um alvo com maior índice de dificuldade não implica necessariamente em um aumento no tempo de execução do movimento, podendo inclusive implicar em uma ligeira redução.

A Tabela 4.3.3 exibe os coeficientes para as demais modalidades, além do índice de performance ($IP = 1/b \times 1000$), exibido em bits por segundo. A Figura 4.7 exibe as retas $T = a + b.ID$ obtidas por regressão linear para as modalidades.

Pela métrica do índice de performance, as modalidades liberal com ajuste para o círculo mais próximo e conservadora mostraram desempenho superior ao do *mouse*, que apresentou

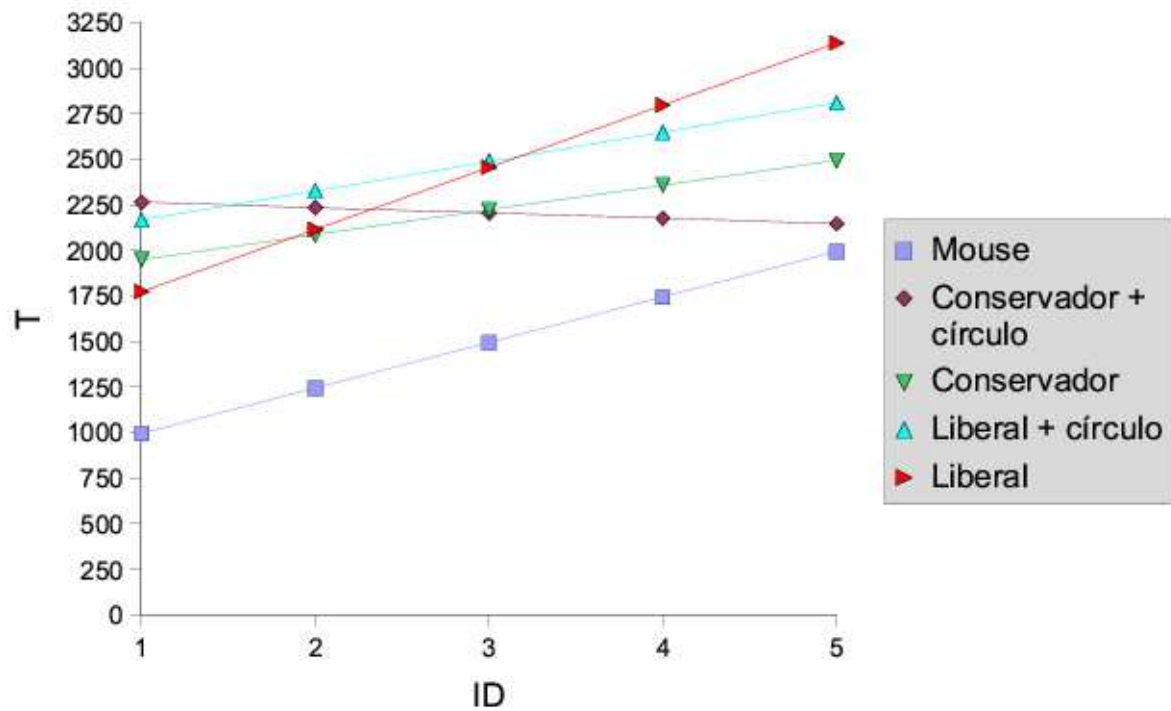


Figura 4.7: Lei de Fitts: retas $T = a + b.ID$ para cada modalidade

Modalidade	a (ms)	b (ms/bit)	IP (bits/s)
<i>Mouse</i>	745,63	249,98	4,00
Conservadora	1815,17	135,99	7,35
Conservadora + círculo mais próximo	2296,43	-29,75	-33,62
Liberal + círculo mais próximo	2006,67	160,76	6,22
Liberal	1433,85	341,33	2,93

Tabela 4.3: Coeficientes de regressão linear para $T = a + b.ID$, onde $ID = \log_2 A/W + 1$

desempenho superior ao da modalidade liberal. Para efeito comparativo, o índice de performance para o *mouse* é comparável ao encontrado em outros testes com tarefas de apontamento, como [ISB91] e [ZMI99].

Capítulo 5

Rumos futuros

A partir dos resultados obtidos até o momento, pretendemos desenvolver uma modelagem mais completa do contexto da interação. Para tal estamos trabalhando na construção de definições para as *ações* fundamentais do processo de interação, ou seja, que podem ser tomadas tanto do usuário sobre a interface quanto da interface sobre o usuário. Uma vez determinado um conjunto básico de ações que, combinadas, possam formar uma variedade razoável de tarefas possíveis no sistema, pretendemos construir uma arquitetura de inferência probabilística baseada em aprendizagem que permita uma estimativa mais precisa da intenção do usuário a partir do histórico de ações tomadas no passado juntamente com a informação da atenção visual.

Paralelamente, vemos a necessidade de refinar nossos procedimentos de teste de forma a investigar com maior segurança algumas das tendências verificadas, como o desempenho de algumas modalidades do MAGIC Pointing na modelagem pela Lei de Fitts e a independência entre distância e tamanho dos alvos e tempo de aquisição. Serão realizados ainda ajustes no sistema de rastreamento de olhar e no próprio MAGIC Pointing visando aprimorar a precisão do sistema e evitar os problemas detectados em nossos testes.

Item	ago	set	out	nov	dez	jan	fev
(a)	x	x					
(b)			x	x			
(c)				x	x		
(d)					x	x	x

Tabela 5.1: Plano de trabalho: agosto/2004 a fevereiro/2005

5.1 Plano de trabalho

Preveremos, então, a execução das seguintes fases nos próximos meses, de acordo com o cronograma da Tabela 5.1:

- (a) Refinamento do software e do procedimento de testes;
- (b) Condução dos novos testes;
- (c) Análise dos resultados dos testes;
- (d) Elaboração da dissertação.

Apêndice A

Questionário do teste

Apresentamos neste apêndice o questionário que foi aplicado aos usuários participantes do teste descrito no [Capítulo 4](#).

Seqüência:

Idade: Sexo: () M () F

Experiência com uso do mouse (em anos):

Para cada modalidade de uso do MAGIC Pointing, e para cada afirmação, marque de 0 (discordo totalmente) a 5 (concordo totalmente):

	Liberal	Liberal círculo mais próximo	Conservador	Conservador círculo mais próximo
Esta modalidade me ajudou a executar a tarefa mais rapidamente				
Esta modalidade é mais confortável do que usar apenas o mouse				

Numere as modalidades de 1 (para a que você mais gostou) a 5 (para a que você menos gostou), de acordo com a experiência que você teve no teste:

- () Somente o mouse
 () Liberal
 () Liberal com círculo mais próximo
 () Conservador
 () Conservador com círculo mais próximo

Figura A.1: Questionário aplicado aos participantes do teste

Referências Bibliográficas

- [AS04] Trolltech AS. Qt toolkit. Web site, July 2004. <http://www.trolltech.com/products/qt/index.html>. 35
- [BE04] Preston Brown and Matthias Ettrich. DCOP: Desktop COmmunications Protocol. Web site, July 2004. <http://developer.kde.org/documentation/library/kdeqt/dcop.html>. 35
- [BG90] V. Bruce and P. R. Green. *Visual perception: physiology, psychology and ecology (2nd edition)*. Lawrence Erlbaum Associates Ltd., 1990. 7
- [Bol81] Richard A. Bolt. Gaze-orchestrated dynamic windows. *Computer Graphics*, 15(3):109–119, august 1981. 19
- [BP94] S. Baluja and D. Pomerlau. Non-intrusive gaze tracking using artificial neural networks. Technical Report CMU-CS-94-102, School of Computer Science, Carnegie Mellon University, Pittsburg PA, USA, 1994. 10, 11
- [CKAF00] C.H.Morimoto, D. Koons, A. Amir, and M. Flickner. Pupil detection and tracking using multiple light sources. *Image and Vision Computing*, v.18(no.4):pp. 331–336, 2000. 3, 9, 11, 30

- [CMN80] S. K. Card, T. P. Moran, and A. Newell. The keystroke-level model for user performance time with interactive systems. *Communications of the ACM*, 23(7):396–410, 1980. 15
- [Duc02] Andrew T. Duchowski. A breadth-first survey of eye-tracking applications. *Behavioral Research Methods, Instruments and Computers*, pages 1–16, 2002. 18, 26, 27
- [Edw98] Gregory Edwards. A tool for creating eye-aware applications that adapt to changes in user behavior. In *Proceedings of ACM ASSETS 98*, Marina del Rey, California, 1998. 17, 27
- [Fit54] P.M. Fitts. The information capacity of the human motor systems in controlling the amplitude of movement. *Journal of Experimental Psychology*, 1954. 42
- [GEN95] A. Glenstrup and T. Engell-Nielsen. Eye controlled media: Present and future state. Master’s thesis, Institute of Computer Science, University of Copenhagen, 1995. 2, 6, 7, 8, 9, 15, 19
- [HJM⁺89] Thomas E. Hutchinson, K. Preston White Jr., Worthy N. Martin, Kelly C. Reichert, and Lisa A. Frey. Human-computer interaction using eye-gaze input. In *IEEE Transactions on Systems, Man and Cybernetics*, volume 19, pages 1527–1534, 1989. 27
- [HK99] Anthony J. Hornof and David E. Kieras. Cognitive modeling demonstrates how people use anticipated location knowledge of menu items. In *Human Factors in Computing Systems: CHI 99 Conference Proceedings*, pages 410–417, 1999. 15
- [Inc04] The XFree86 Project, Inc. X Window System. Web site, July 2004. <http://www.xfree86.org>. 34

- [ISB91] MacKenzie I.S., A. Sellen, and W. Buxton. A comparison of input devices in elemental pointing and dragging tasks. In *Proceedings of the ACM Conference on Human Factors in Computing Systems - CHI '91*, pages 161–166. ACM, 1991. 44, 48
- [Jac90] Robert J. K. Jacob. What you look at is what you get: eye movement-based interaction techniques. In *Proceedings of the CHI'90*, pages 11–18, 1990. 6, 8, 13, 14, 16, 20, 24
- [Jac95] Robert J. K. Jacob. Eye tracking in advanced interface design. In W. Barøeld and T. Furness, editors, *Advanced Interface Design and Virtual Environments*, pages 258–288, Oxford, England, 1995. Oxford Press. 7
- [MSS91] G. A. Myers, K. R. Sherman, and L. Stark. Eye monitor: Microcomputer-based instrument uses an internal mode to track the eye. In *Computer*, volume 24, pages 14–21. IEEE Computer Society Press, Los Alamitos, CA, USA, 1991. 9
- [OC00] Sharon Oviatt and Philip Cohen. Multimodal interfaces that process what comes naturally. *Communications of the ACM*, 43(3):45–53, March 2000. 2
- [ODH02] Carol O'Sullivan, John Dingliana, and Sarah Howlett. Eye-movements and interactive graphics. In J. Hayöna, R. Radach, and H. Deubel, editors, *The Mind's Eyes: Cognitive and Applied Aspects of Eye Movement Research*, Oxford, England, 2002. Elsevier Science. 23, 28
- [OMY02] Takehiko Ohno, Naoki Mukawa, and Atsushi Yoshikawa. FreeGaze: a gaze tracking system for everyday gaze interaction. In *Proceedings of the Eye Tracking Research & Application Symposium 2002 (ETRA-02)*, pages 125–132, 2002. 9, 11

- [Pro04] The GNOME Project. GTK+ toolkit. Web site, July 2004. <http://developer.gnome.org/arch/gtk/>. 35
- [Rab89] Lawrence R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. In *Proceedings of the IEEE*, volume 77, 1989. 26
- [Rob63] D. A. Robinson. A method of measuring eye movements using a scleral search coil in a magnetic field. In *IEEE Transactions on Biomedical Engineering* 10, pages 137–145, 1963. 8
- [SA00] Dario D. Salvucci and John R. Anderson. Intelligent gaze-added interfaces. In *Human Factors in Computing Systems: CHI 2000 Conference Proceedings*, 2000. 17, 23, 26, 27, 41
- [Sal99a] Dario D. Salvucci. Inferring intent in eye-based interfaces. In ACM Press, editor, *Human Factors in Computing Systems: CHI 99 Conference Proceedings*, pages 254–261, 1999. 15, 17, 24, 26
- [Sal99b] Dario D. Salvucci. *Mapping Eye Movements to Cognitive Processes*. PhD thesis, Carnegie Mellon University, 1999. 16
- [SG00] Dario D. Salvucci and Joseph H. Goldberg. Identifying fixations and saccades in eye-tracking protocols. In *Proceedings of the Eye Tracking Research and Applications Symposium*, pages 71–78, 2000. 15
- [SJ00] Linda E. Sibert and Robert J. K. Jacob. Evaluation of eye gaze interaction. In *Proceedings of the CHI'00*, pages 281–288, 2000. iii, 14, 20, 21
- [Tur98] Matthew Turk. Moving from GUIs to PUIs. In *Proceedings of the Symposium on Intelligent Information Media*, 1998. 2, 17

- [Ver02] Roel Vertegaal. Designing attentive interfaces. In *Proceedings of ACM ETRA Symposium on Eye Tracking Research and Applications 2002*. ACM Press, 2002. [2](#), [16](#)
- [WZS01] Jingtao Wang, Shumin Zhai, and Hui Su. Chinese input with keyboard and eye-tracking – an anatomical study. In *Proceedings of the CHI'01*, pages 349–356, 2001. [iii](#), [22](#)
- [YS75] L. Young and D. Sheena. Methods & designs: Survey of eye movement recording methods. *Behavioral Research Methods and Instrumentation*, 7(15):397–429, 1975. [7](#), [8](#), [9](#), [10](#)
- [Zha03] Shumin Zhai. What's in the eyes for attentive input. *Communications of the ACM*, 46(3):34–39, March 2003. [21](#), [30](#)
- [ZMI99] S. Zhai, C. Morimoto, and S. Ihde. Manual and gaze input cascaded (MAGIC) pointing. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI'99)*, pages 246–253, 1999. [3](#), [14](#), [16](#), [21](#), [40](#), [41](#), [48](#)