UNIVERSIDADE FEDERAL DA BAHIA

INSTITUTO DE MATEMÁTICA

DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO

Paulo César de Alencar Gonçalves Filho

Um Motor para Interface Homem-Máquina Baseada em Interação com Avatar

Salvador

2011

Paulo César de Alencar Gonçalves Filho

Um Motor para Interface Homem-Máquina Baseada em Interação com Avatar

Monografia apresentada ao Curso de graduação em Ciência da Computação, Departamento de Ciência da Computação, Instituto de Matemática, Universidade Federal da Bahia, como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação.

Orientador: Prof. Luciano Oliveira Co-orientadora: Profª . Vaninha

Salvador

2011

RESUMO

Uma interface homem-máquina (IHM) é um sistema composto de hardware e software que realiza a comunicação entre o humano e o computador. Uma vez que a utilização de máquinas tem se tornado cada vez mais pervasivo em nossos dias, a construção de IHMs desempenha um papel de fundamental importância. Nesse contexto e adicionalmente à diminuição do custo do hardware, a interação homem máquina tem se tornado cada vez mais natural. A naturalidade de tal interação pode ser medida pelo grau de compreensão da IHM de realizar uma interação mais similar a interação entre dois humanos. Até então, o que se percebe é o homem necessitando de adequação aos hardwares convencionais, tais como teclado, mouse, tela sensível ao toque, entre outros. Isto tem como consequência uma comunicação homem-máquina burocrática e elitista.

Recentemente, vê-se um crescente de novos tipos de interface baseadas em reconhecimento de gestos, onde o ser humano interage com um avatar (ser virtual) a partir de reconhecimentos de alto nível (gestos, voz, etc) a fim de fazer com que a máquina responda a alguns comandos. IHMs baseadas na interação com avatar tentam fazer com que um personagem virtual utilize uma linguagem natural humana para se comunicar com o usuário. Desta forma, a utilização de uma interface como esta torna mais fácil a interação humana-computador. Nessa direção, este trabalho visa contribuir com um motor genérico para a criação de IHMs baseadas em avatar.

Com este objetivo, um conjunto de módulos presentes no motor proposto foi definido com base em outros trabalhos. As estruturas principais do motor proposto são: os módulos de reconhecimento, lógico e avatar. O módulo de reconhecimento transforma dados de entrada alto nível (imagens de gestos ou sintetização de voz) em uma linguagem entendida pela máquina. O módulo lógico é responsável pela inteligencia artificial do personagem e pela ligação da interface com as funcionalidades do aplicativo. Por fim, o módulo do avatar é responsável por atributos do personagem como sintetização de voz e gesticulação labial.

Além destas estruturas, é possível a agregação de novos módulos dependendo do tipo do projeto. Como são estruturas de domínios distintos, possuem regras de comunicação distintas, porém precisam de uma forma de integração. A informação de um gesto de alto nível identificado pelo módulo reconhecedor precisa ser repassada para o módulo lógico executar a tarefa correta. Essa função de integração é exercida pelo gerenciador de sinais. A comunicação através de sinais e os módulos isolados definem uma visão geral do motor, mas não servem como uma forma de avaliar a integração de todo o sistema. Para tanto, foi elaborada uma aplicação exemplo com base no motor proposto. Tal prova de conceito utiliza um personagem e o reconhecimento de gestos da mão. O avatar é modelado tridimensionalmente a fim de expressar um comportamento e uma comunicação verbal do avatar. A comunicação verbal se da à partir de uma proposta de generalização de movimentos da boca com base em padrões comuns de posicionamento dos lábios. O reconhecimento de gestos da mão é realizado com a ajuda de uma câmera time-of-flight.

Palavras-chave: interface homem-máquina, motor, sincronia labial, reconhecimento de gestos, avatar, câmera Time-of-Flight.

LISTA DE FIGURAS

LISTA DE ABREVIATURAS E SIGLAS

cog_z	Centro de gravidade no eixo z,	p. 41
HMM	Hidden Markov Model,	p. 16
IHM	Interface Homem-Máquina,	p. 10
SV	Support Vector,	p. 15
SVM	Support Vector Machine,	p. 15
TOF	Time-of-flight,	p. 17

SUMÁRIO

1 INTRODUÇÃO

1.1 MOTIVAÇÃO

Uma Interface homem-máquina (IHM) é um sistema com recursos de hardware e software que promove a comunicação entre o computador e o homem. As IHMs fazem traduções entre os domínios da linguagem humana e da linguagem do computador. Com essa tradução de domínios, tornou-se possível a utilização do processamento de máquina em tarefas realizadas pelo homem. Atividades repetitivas que eram realizadas manualmente passaram a ser automatizadas no computador. Além disso, a alta velocidade em cálculos da máquina passou a otimizar o tempo de execução de uma tarefa.

Para ocorrer a tradução de linguagens na IHM é necessário que o homem utilize dispositivos de comunicação. Estes dispositivos tradicionalmente contém o domínio do alfabeto da linguagem entendida por um usuário do computador, porém não usam a linguagem natural humana (gestos e fala). O teclado, por exemplo, é um dispositivo de entrada de informações no computador. Ele possui um conjunto de caracteres que fazem parte do alfabeto de um país, porém os comandos e atalhos utilizados para comunicação com a máquina não são intuitivos. Ou seja, exigem certos conhecimentos para a correta manipulação. Neste caso, o dispositivo e o homem possuem o mesmo domínio utilizado pela linguagem (letras e números), mas suas aplicações não utilizam a mesma linguagem (FIGURA 1a). Em contrapartida, se uma IHM utiliza a linguagem natural humana (gestos, voz, etc), as interações e formas de utilização ficam mais intuitivas. Tarefas que antes eram executadas por comandos de um teclado podem ser realizadas através de ordens ou pedidos, como se interlocutor estivesse conversando com outra pessoa (FIGURA 1b).Portanto, a etapa de aprendizagem do funcionamento dos comandos de interface pode ser ignorada se a IHM puder se comunicar na linguagem natural humana.

Existem propostas de interfaces homem-máquina que tentam interagir através da linguagem natural para se comunicar com um usuário. Elas se utilizam de personagens virtuais que podem fazer gestos e reproduzir diálogos semelhantes ao de um homem. Os comandos apresentados

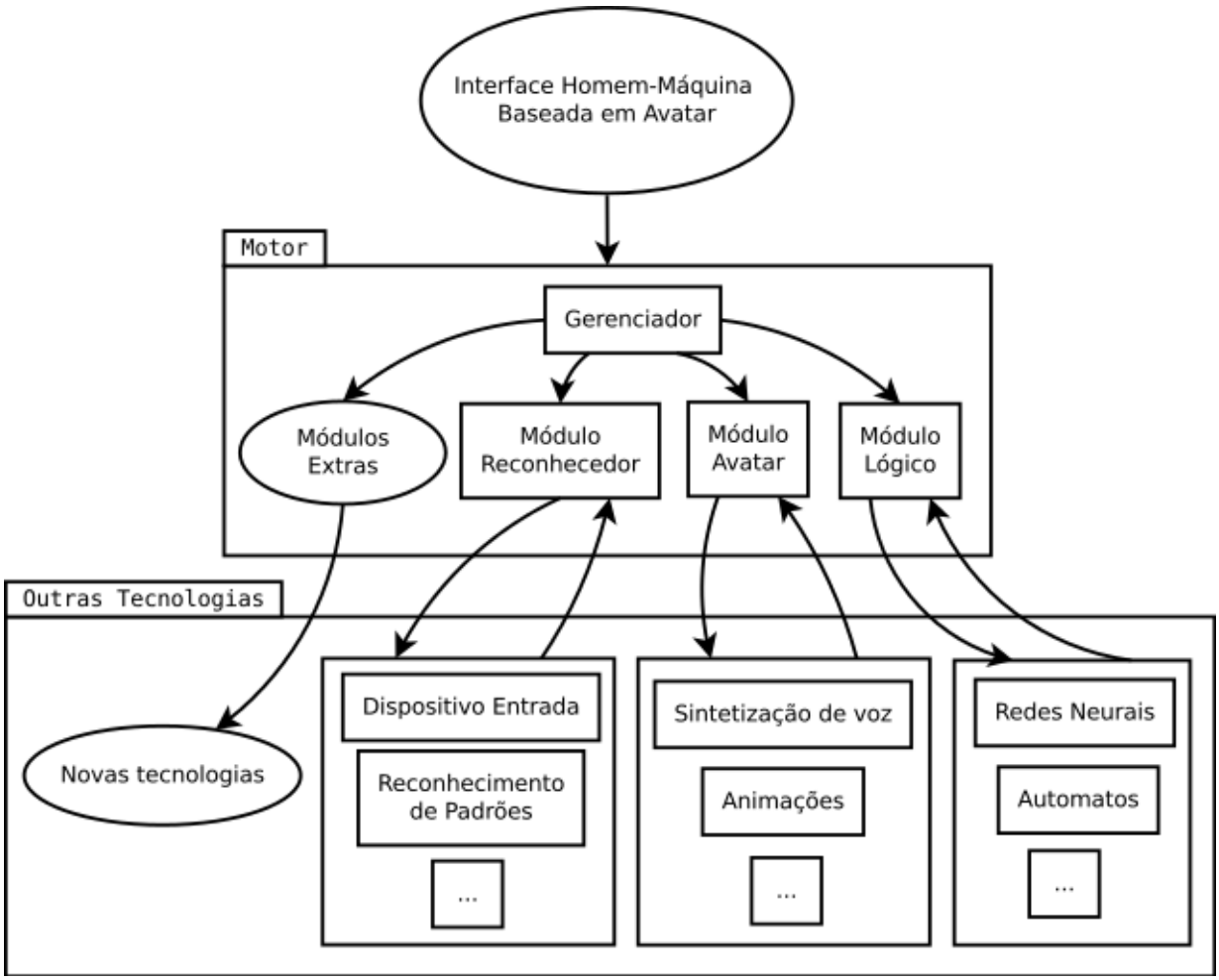

Figura 1.1: Diagrama de utilidade do motor proposto. [FALAR DOS MODULOS]

pelo usuário também podem ser mostrados em forma de voz ou gestos. Essas interfaces podem ser denominadas como IHM baseada em interações com avatar. O objetivo é fazer com que a interação com o software pareça uma conversa entre pessoas.

Para a utilização da interface se assemelhar ao diálogo entre pessoas, é necessário que a IHM com avatar apresente um conjunto de recursos que substituam o cognitivo humano durante a comunicação. A máquina precisa entender como interpretar e reproduzir os gestos e voz de um indivíduo. Este conhecimento abrange varias áreas de conhecimento, tais como: ambientes virtuais, reconhecimento de padrões, engenharia de software, entre outros. Entretanto é necessário um conjunto grande de funcionalidades que podem tornar o desenvolvimento individual deste tipo de interface genérica inviável na prática. Espera-se é que exista um sistema que agregue os principais recursos da IHM para o desenvolvedor se concentre puramente na elaboração dos serviços. Este conjunto de recursos dentro do sistema agregador da IHM

Existem algumas propostas de arquitetura de motores para interfaces homem-máquina com avatar, porém possuem limitações funcionais [REFERENCIA]. Grande parte é voltada para

um domínio de conhecimento específico, ou seja, apenas para atender a necessidade do projeto [REFERENCIAS]; outros de alguma área da IHM baseada em avatar [REFERENCIA]. A motivação do trabalho aqui proposto foi, portanto, o desenvolvimento de um motor genérico de que pudesse atender genericamente as necessidades de projetos de IHM baseados em interações com avatar. A Fig. 1.1 ilustra a idéia da proposta.

1.2 OBJETIVOS

O objetivo geral do trabalho foi desenvolver um motor genérico de interface homem-máquina com interação baseada em avatar. Particularmente, os seguintes objetivos específicos podem ser listados:

- Reconhecedor de gestos da mão como forma de comunicação entre o usuário e o computador;
- Desenvolvimento de um sistema de animação automática para a gesticulação do avatar;
- Construção de um motor visando a atualização e inclusão de módulos;
- Implementação de uma prova de conceito que integrasse as diversas funcionalidades do motor.

1.3 CONTRIBUIÇÃO

As contribuições realizadas por esta monografia são:

- Os motores para desenvolvimento de interfaces homem-máquina baseados em interações com avatar atualmente são limitados a um domínio de assunto e/ou línguas específicas. São motores criados para satisfazer apenas às necessidades que o projeto de uma IHM se propõe. Neste sentido, este trabalho apresenta uma solução de motor genérico que sirva tanto para pesquisas e quanto para projetos de IHMs baseadas em interações com avatar;
- Para que ocorra uma comunicação natural entre o homem e o avatar, é necessário que o avatar consiga se expressar tão próximo dos gestos do ser humano quanto possível. Esta proximidade se dá a partir movimentos corporais ou comunicação verbal. Para a ilustração de um caso de uso, muniu-se o motor da capacidade de uma gesticulação natural por parte do avatar. A sincronia labial do avatar precisa, portanto, estar condizente com sua voz. Os lábios do personagem devem estar posicionados de forma semelhante à humana quando pronunciar uma determinada letra ou conjunto de letras. Deve haver uma gesticulação correta para o conjunto de palavras que forem pronunciadas. Este recurso faz com que a comunicação do avatar seja mais realista, a fim de passar a sensação ao usuário de que há um personagem inteligente se comunicando com ele. Em geral, os motores com soluções de gesticulação fornecem [REOLHAR PESQUISA DE ARTIGOS] são feitos para línguas específicas, o que limita a utilização das animações labiais a uma linguagem.

Elas não tentam generalizar a gesticulação como a proposta presente no motor neste trabalho. O motor apresenta uma solução de sincronização genérica com base nos padrões labiais mais expressivos presentes nas línguas inglesa, chinesa e espanhola. Foram encontrados quatro padrões que são iguais ou semelhantes a outros posicionamentos labiais. A explicação da criação e funcionamento da sincronia é discorrida na Seção 4.5.

- MODIFICAÇÃO RECONHECIMENTO

1.4 MAPA DA MONOGRAFIA

A monografia está organizada da seguinte forma:

- **Capítulo 2 (Estado-da-arte):** apresenta um conjunto de conceitos das tecnologias usadas no trabalho, descrição de soluções existentes e análise comparativa entre elas e a implementação proposta pela monografia;
- **Capítulo 3 (Visão geral do motor proposto e da prova de conceito):** descreve de forma geral como a interface proposta irá funcionar. Faz um resumo dos Capítulos 4, 5.5 e 5;
- **Capítulo 4 (Interface homem-máquina com interação baseada em avatar):** mostra como está organizada toda a arquitetura do sistema proposto e discute características relevantes de módulos específicos;
- **Capítulo 5 (Prova de conceito):** apresenta a especificação do modelo de interface implementado. Neste capítulo é mostrado as imagens, textos, automato de execução do avatar e análise de resultados do reconhecimento de gestos;
- **Capítulo 6 (Discussão e conclusão):** discorre sobre os resultados obtidos, conclusões tiradas deles e possíveis trabalhos futuros.

2 ESTADO-DA-ARTE

Para haver um embasamento teórico na elaboração do trabalho, este capítulo faz uma revisão bibliográfica de trabalhos na área de interfaces homem-máquina baseadas em avatar e de reconhecimento de gestos da mão. Depois é feita uma análise comparativa dos conceitos revisados com o trabalho proposto.

2.1 INTERFACE HOMEM-MÁQUINA COM AVATARES

IHMs baseadas em interações com avatar são sistemas providos de hardware e software que utilizam um personagem virtual para realizar a comunicação entre o homem e o computador. A máquina tenta criar uma forma de interação próxima à humana para trocar informação com o usuário. Está forma de comunicação busca chegar próxima a linguagem natural humana (gestos e voz) para facilitar a manipulação do sistema, pois dessa forma, não há uma necessidade do usuário aprender novas regras de comunicação como comandos específicos, simbologias diferentes ou novas palavras. Basta a pessoa utilizar comandos e pedidos como se estivesse conversando com outra pessoa.

Primeiramente, estes tipos de interfaces baseadas em avatar devem levar em conta o personagem virtual (Figura 2.1). O modelo deve ter uma aparência visual atrativa e deve passar a sensação de percepção do usuário (XIAO; STASKO; CATRAMBONE, 2002). Sistemas com modelos mais realísticos passam a sensação de que são mais inteligentes e a comunicação humana ocorre quando há a percepção mútua dos interlocutores. Como forma de atingir estes paradigmas, a maioria das propostas utilizam avatares humanoides que interagem de frente para a tela , como se estivessem olhando o usuário (BALDASSARRI; CEREZO; SERON, 2008) (UCHINO et al., 2007). Outros sistemas acrescentam mais expressividade à face e postura do avatar para melhorar a imersão na utilização da interface (CAROLIS; ROSIS; CAROFIGLIO, 1999) (HERRERA et al., 2010). Quanto maior a semelhança com comportamentos humanos, melhor é a assimilação e identificação do usuário com a interface.

Figura 2.1: Esboço do funcionamento de uma interface homem-máquina baseada em avatar. O personagem virtual no monitor se comunica em linguagem natural com uma pessoa do mundo real. Ambos utilizam gestos corporais e voz para interagir.

Estas características do personagem virtual, levam em conta apenas sua forma de representação para que o usuário tenha a sensação de presença de uma entidade inteligente. Mas, como se trata de uma interface, é necessário que haja maneiras de se realizar a comunicação.

Como forma de expressão, uma IHM baseada em avatar deve poder utilizar gestos ou voz (linguagem natural) para fazer a comunicação com uma pessoa. Este tipo de interação facilita a troca de informação para o usuário, por se tratar de um protocolo de comunicação de domínio humano. Porém, a máquina precisa expressar-se com uma linguagem desconhecida para seu sistema. É necessário que haja uma conversão da linguagem de máquina para linguagem do homem. Como forma do personagem utilizar a voz para se comunicar, existem soluções com sintetizadores que recebem texto como entrada e retornam áudio como saída (UCHINO et al., 2007). O computador transforma as palavras escritas em palavras faladas que podem ser reproduzidas em caixa de som. Além dessa proposta, existem soluções que utilizam a expressão corporal para se comunicar com o usuário ou para complementar a comunicação verbal do personagem . Como exemplo, há a utilização desses conceitos em avatares que tentam interagir com pessoas surdas através de linguagens de sinal (LOMBARDO et al., 2011) ou tentam expressar mais informações sobre emoções através de posturas corporais ou faciais (HERRERA et al., 2010).

Além da necessidade de expressão do avatar para que o usuário entenda a informação de

uma maneira mais fácil. É necessário que a máquina consiga entender a linguagem natural humana para realizar a comunicação da forma esperada. O computador precisa reconhecer gestos e voz para entender de forma efetiva a informação do usuário. Para suprir esta necessidade na interação homem-máquina com linguagem natural, existem algumas soluções que tentam utilizar câmeras para reconhecer gestos da mão, face e postura para que o avatar possa tirar informações extras do usuário (BALDASSARRI; CEREZO; SERON, 2008) (UCHINO et al., 2007). Também há outra proposta que faz o reconhecimento de voz e tenta interpretar a ideia do usuário para manter um diálogo sobre determinado domínio de assunto (DEMARA et al., 2008).

Com o ciclo de comunicação de reconhecimento e sintetização da linguagem natural do computador e com o personagem virtual, é possível se estabelecer uma IHM baseada em interações com avatar. As funcionalidades básicas da interface são atendidas com estes recursos, mas não é só isso o necessário. O personagem, como foi citado anteriormente, precisa ter certo grau de realismo para ocorrer uma maior imersão na interface. Neste contexto, existe a sincronia labial, que tenta unir a sintetização de voz com animações labiais do personagem virtual. A medida que o áudio da fala for emitido, o avatar deve apresentar uma gesticulação condizente com o som de um fonema. Ou seja, o modelo virtual do personagem deve ter o visema correto para cada letra pronunciada no sintetizador de voz. Como forma de tentar solucionar este problemas, foram achadas soluções de gesticulação para línguas específicas (HUANG; YIN; ZENG, 2005) (WATERS; LEVERGOOD, 1994). Com estas propostas é possível fazer animações para as determinadas línguas. Existem outras propostas que fazem uma compilação dos visemas de várias línguas, fazem uma interpolação destes conjuntos e mapeiam o resultado para as línguas utilizadas (OH et al., 2010) (WANG; XU, 2010). Dessa forma apenas um único sistema consegue dar suporte a mais de uma língua.

Como pode ser visto, a elaboração de uma interface homem-máquina baseada em avatar exige conhecimento de várias áreas da computação, entre elas, reconhecimento de padrões de gestos ou voz, sintetização de voz, ambientes virtuais, sincronia labial e outras. Normalmente estas áreas são bastantes distintas e podem exigir muito esforço para elaborar um sistema integrado onde utilize todos estes recursos. Desse modo, surge um problema de desenvolvimento, ou o projeto é estendido para elaborar a interface com avatar ou é simplificado com o uso de uma interface tradicional (teclado, mouse, etc). Para solucionar este problema, existem os motores. Eles são sistemas que fornecem um conjunto de funcionalidades e bibliotecas para a elaboração de aplicações de determinada área.

Na ambiente de IHM baseada em interações com avatar, existem motores que disponibi-

lizam vários recursos para o desenvolvimento da interface. Há uma proposta de motor onde o usuário ou desenvolvedor pode construir toda IHM e suas funcionalidades através de scritps (BALDASSARRI; CEREZO; SERON, 2008). Outra proposta é um motor com arquitetura modular, onde cada módulo manipula uma área de conhecimento e é aplicado em computadores diferentes (UCHINO et al., 2007). Também há uma arquitetura de motor para avatares de comunicação baseada em diálogo (DEMARA et al., 2008). O personagem interpreta a voz humana, processa as informações e gera uma resposta condizente com a conversa.

2.2 RECONHECIMENTO DE GESTOS

Na linguagem natural humana, os gestos são recursos usados para realizar a comunicação entre duas pessoas. Através de sinais manuais, expressões faciais e posicionamento corporal, a comunicação verbal é complementada. Para que uma IHM baseada em interações com avatar que usa a linguagem natural consiga se comunicar com uma pessoa, é necessário que o computador saiba reconhecer esses gestos presentes na linguagem humana. A máquina precisa distinguir uma expressão gestual da outra e saber o significado delas para que seja bem sucedida na comunicação. Como tentativa de solução desse problema, a área de reconhecimento de padrões estuda formas de reproduzir estratégias da visão e percepção cerebral humana através de algoritmos. Ela tenta enxergar e categorizar informações usando o processamento do computador (NIXON; AGUADO, 2008). Dessa forma, a máquina pode buscar por padrões da forma de interação humana para compreender a mensagem.

Um padrão é definido com uma descrição quantitativa ou estrutural de um objeto ou uma entidade de interesse (BOW, 2002). Vários objetos ou entidades com um o mais padrões em comum podem ser separados em classes de padrão de acordo com as características semelhantes. Por exemplo, homem, mulher e gorila estão na mesma classe de animais bípedes enquanto cavalo e cachorro estão na classe dos quadrúpedes. Porém homem e mulher estão na mesma classe de espécie (Homo sapiens) e o gorila pertence a outra classe. Um objeto pode ser categorizado em mais de uma classe de padrão. Portanto para reconhecer os membros de um grupo, basta identificar qual o padrão comum entre eles.

O reconhecimento de padrões é o processo de categorização de qualquer medida ou dados em classes ou categorias. Este ato de reconhecer define se determinado padrão pertence ou não a um conjunto. No caso de gestos manuais, pode-se identificar se uma determinada imagem é uma mão ou não e depois buscar com qual gesto ela se assemelha mais, por exemplo, categorizar a mão em gesto de um, dois ou três dedos.

O ato de reconhecer o objeto muitas vezes necessita de um tratamento prévio da imagem, pois é comum a ocorrência de fatores como ruídos, baixa luminosidade e outros objetos, que podem atrapalhar na identificação. A imagem precisa estar de um jeito próximo ao ideal para que o computador consiga categorizá-la corretamente. Portanto o reconhecimento de padrões pode ser dividido em pré-processamento de imagem (extração de features) e classificação (SNYDER; QI, 2010) (MONI; ALI, 2009). Estes dois itens são definidos como:

- Pré-processamento de imagem: tratamento da imagem antes de ser classificada na tentativa de facilitar e aumentar o desempenho do reconhecimento. No mundo real as imagens não são perfeitas, elas possuem pertubações que tornam o padrão difícil de ser identificado. O pré-processamento da imagem pode ser feito para reduzir uma frequência da imagem (ruído), por exemplo, e facilitar o reconhecimento. O resultado dessa transformação da imagem é chamado de feature;
- Classificação: é a tentativa propriamente dita de tentar reconhecer um padrão objeto ou entidade a partir de um recurso de entrada. Nessa área é comum a aplicação de conhecimentos sobre inteligência artificial para escolher a classe do objeto.

O pré-processamento e a classificação podem ser aplicados em várias áreas de reconhecimento de padrões. Estas técnicas não se limitam apenas a imagens. Podem ser usados em reconhecimento de sons, sinais e outros. Mas por delimitação do escopo da proposta, o foco deste trabalho é apenas em reconhecimento de gestos da mão. A seguir são listados alguns exemplos de estratégias utilizadas para identificar gestos manuais:

- Reconhecimento de gestos utilizando o método Viola-Jones e Support Vectors Machine (SVM) (Liu Yun, 2009): o autor propõe módulos separados entre reconhecimento de gesto, extração feature de momentos invariantes e classificação do gesto. Na primeira etapa é aplicado o método Viola-Jones (VIOLA; JONES, 2002) para reconhecer o gesto. Esse método usa as Haar-like features e a integral da imagem para extrair as informações de interesse, depois é usado o algoritmo AdaBoost para selecionar as features de maior peso (treinamento) e gerar um classificador. Por fim os melhores classificadores são combinados em cascata para fazer o reconhecimento do gesto. Ou seja, a solução classifica uma imagem como gesto através da decisão tomada por um conjunto de classificadores fracos. Na segunda etapa, é aplicada sobre o gesto classificado uma extração de feature que é invariante a translação, rotação e escala da imagem. As imagens são representadas como se estivessem sem variações. Por fim é aplicado o SVM sobre os resultados da segunda etapa. Esse método utiliza uma função linear em um espaço de alta dimensão

para estimar uma superfície de decisão. As imagens são representadas como vetores de suporte (SV) e a decisão do SVM é binária (é ou não é o gesto), depende de qual lado da função linear projetada na superfície de decisão o SV está;

- Reconhecimento de gestos a partir da detecção da pele e remoção de defeitos (KATHURIA, 2011): em sua proposta o autor primeiramente transforma a imagem de entrada para os espaços de cores Hsv e YC_rC_b, que são mais fáceis para detectar superfícies da pele. Depois aplica o algoritmo de detecção de pele em ambos e pega o melhor resultado. Em seguida é feita uma busca pela mão, seleciona-se a maior região que contenha pele e tenta identificar o gesto. Caso a região seja bem maior que o esperado e haja falha no reconhecimento, aplica um algoritmo que procura a camada ideal da mão e tenta eliminar partes da imagem que sejam defeitos. A proposta é lapidar o que foi detectado como pele para tentar chegar a um gesto próximo à mão;

- Reconhecimento de gestos a partir de detecção de blobs e ridges (LAPTEV; LINDEBERG, 2001): no artigo o autor tenta primeiramente destacar regiões que possam fazer parte da mão através da elevação delas. Representando na imagem, essas elevações (blobs e ridges) são tratadas como regiões mais claras. Depois o autor busca por padrões semelhantes aos de gestos da mão criados pelo conjunto dessas regiões destacadas. Nessa proposta podem haver regiões destacadas na imagem que não pertençam à mão;

- Reconhecimento de gestos usando o Hidden Markov Model (HMM) (MONI; ALI, 2009): no artigo é apresentado técnicas e estratégias para o reconhecimento de gestos da mão usando HMM. O modelo de Markov é um automato de estados finitos onde cada transição entre estados tem um determinado valor de probabilidade associado. O objetivo desse modelo é determinar os parâmetros ocultos a partir dos parâmetros observáveis. Nesta proposta o autor não leva em conta os ruídos da imagem.

Os artigos citados acima sugerem reconhecimento de gestos utilizando imagens ou vídeos sem perspectiva de profundidade (2D). Apesar de resolverem problemas de extração da mão de forma satisfatória, ainda persistem alguns problemas relacionados a ruídos do plano de fundo da imagem. Como são figuras sem profundidade, não é possível destacar a mão por estar em uma posição mais à frente de outros objetos da imagem. A mão e o plano de fundo são tratados como itens com mesma profundidade. Com isso, alguns elementos próximos à região de interesse podem atrapalhar na decisão. No caso da detecção de pele, por exemplo, se a região atrás da mão estiver em um tom de cor semelhante, não será possível achar a região de interesse correta, pois tudo será considerado como pele e não aparecerá o formato da mão na feature extraída. Em outros casos o ruído do plano de fundo pode ser considerado como mais um elemento da mão.

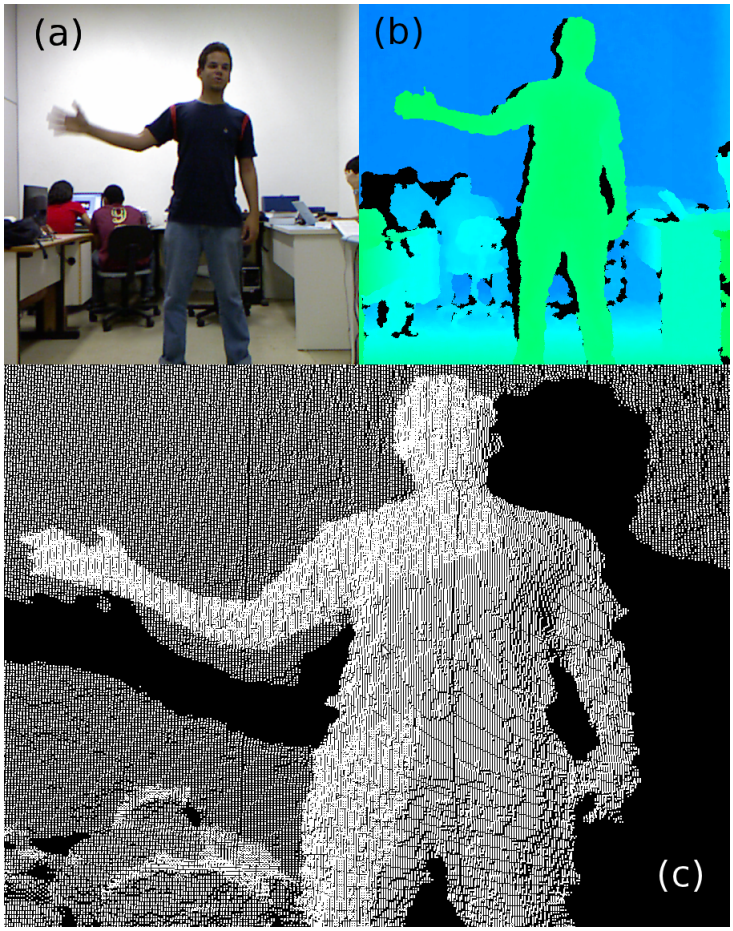

Figura 2.2: Imagem capturada por uma camera time-of-flight. Em (a) tem a imagem de uma câmera normal, que não apresenta profundidade. Em (b) a imagem apresenta um gradiente de cores que indicam profundidade. As áreas com tons de verde e amarelo indicam proximidade e os tons de azul indicam distancia. Em (c) é apresentada uma visualização 3D do ambiente a partir da imagem de profundidade.

Para solucionar esses problemas causados pela ausência de profundidade, é necessário que as soluções gastem mais processamento para tratamento das features. Precisa-se remover os ruídos em volta da região de interesse. Quanto mais detalhes tem a imagem, pode ficar pior de identificar um objeto e as vezes ser impossível de decidir.

Uma boa solução para melhorar a detecção de gestos e ignorar casos de ruídos no plano de fundo são as câmeras time-of-flight (TOF). A diferença desses dispositivos para as câmeras normais (2D) é que eles podem capturar toda cena do ambiente em 3D (Figura 2.2), ou seja, pode detectar a profundidade da cena capturada. O primeiro tratamento da imagem pode ser eliminar as camadas mais distantes e aplicar o estudo do reconhecimento apenas sobre objetos mais próximos. Dessa forma, possíveis ruídos que atrapalhariam a região de interesse durante uma detecção de gestos são eliminados.

Como exemplo de detecção de gestos utilizando câmera TOF. Existe um algoritmo que o

primeiro passo é limitar um intervalo de profundidade em que a mão pode está (KOLLORZ; PENNE; HORNEGGER, 2008). Neste trabalho o autor não faz muitos tratamentos para extrair o gesto. Qualquer região que aparecer na área limitada é aplicado o algoritmo para classificar a mão sem tratamento de ruídos, pois toma-se como base que outras características indesejadas (plano de fundo, o resto do corpo, etc) são excluídas por estarem fora dos limites especificados. A seguir são explicados alguns passos do algoritmo de forma resumida. No Capítulo 5.5 o algoritmo é melhor explicado juntamente com uma proposta de modificação. As etapas do algoritmo são:

- Seleção da mão e braço: o algoritmo delimita a distancia de detecção da imagem para pegar apenas o braço e a mão. Com uma menor profundidade, o plano de fundo e o resto do corpo do usuário não é selecionado;
- Região de interesse: a mão é separada do braço aproximadamente no pulso e uma região quadrada onde cabe apenas a mão é selecionada. Esta secção é chamada de região de bounding box;
- Suavização: devido a imperfeições presentes na captura, é aplicado uma gaussiana sobre a imagem para suavizar as bordas da mão;
- Projeções no eixo X e Y: o bounding box da mão é projetado no eixo X e Y criando um histograma;
- Classificação da mão: o algoritmo calcula a proximidade entre o gesto a ser identificado e os gestos ideais para um conjunto de frames. Quanto mais próximo, maior a probabilidade de ser o gesto correto. Esse processo utiliza a normalização das projeções e a profundidade para calcular a distancia. O intervalo do resultado deste calculo está entre 0 e 1.

Como foi visto no algoritmo citado acima, são feitos menos tratamentos para a extração de features da mão. A região de interesse é selecionada pela distancia, há uma secção na região do pulso, aplica-se uma gaussiana para suavizar e são calculados os histogramas das projeções. Essa menor quantidade de passos devido à maior facilidade em identificar o gesto acaba exigindo menos processamento do computador, o que deixa a máquina com mais recursos para executar outras tarefas ou tentar reconhecer o próximo gesto. Portanto, o reconhecimento de comunicação natural gestual, que são construídas com constantes sucessões de gestos, podem ser otimizados através do uso de câmeras TOF. Neste contexto, uma IHM baseada em interações com avatar que utilize comunicação não-verbal (gestos) pode ter seu desempenho melhorado

se usar o conceito por trás do time-of-flight. Os recursos não utilizados no reconhecimento pode ser redirecionado para outras atividades que funcionam em paralelo neste tipo de interface (sintetização de voz, animações do personagem, etc).

2.3 SISTEMAS SIMILARES

Antes de desenvolver o atual trabalho, foi necessário haver uma pesquisa de outros softwares parecidos para comparar e validar a existência dos novos conceitos apresentados. Com base na proposta de um motor genérico para IHM baseada em interações com avatar e nos tópicos debatidos anteriormente sobre interface homem-máquina com avatar (Seção 2.1) e reconhecimento de gestos (Seção 2.2), a pesquisa sobre soluções semelhantes identificou três contribuições existentes: o Maxine (BALDASSARRI; CEREZO; SERON, 2008), um software de interação com avatar criado por (UCHINO et al., 2007) e o AlexAvatar (DEMARA et al., 2008).

O primeiro trabalho a ser analisado é o Maxine. Um motor manipulado através de scripts que gerenciam ambientes 3D, sons e animações. Apesar de poder ser usado para diversas finalidades que envolvam ambientes virtuais, tem uma tendencia maior a facilitar trabalhos com avatares interativos. Com esse objetivo, ele fornece suporte a módulos de emoções, sincronização de lábios, voz sintetizada e outros. Estas funcionalidades ajudam a elaborar semelhanças com aparências e personalidades humanas no avatar. Com esse conjunto de recursos, o personagem consegue se aproximar mais da forma humana de exposição da informação. Ou seja, é uma boa solução para IHM com avatar quanto a apresentação da informação da máquina para o usuário. No sentido inverso, do homem para a máquina, o Maxine oferece um suporte a dispositivos de entrada. O módulo de entrada foi bem elaborado, disponibiliza suporte a várias formas de manipulação do sistema, teclado, voz, imagem e movimento. Essa atenção foi dada com o objetivo de aumentar a gama de usuários em potencial e melhorar a imersão das pessoas no sistema. Além disso, um maior número de sensores ajuda a melhorar a percepção do ambiente. Em resumo, o Maxine tem a característica de conseguir fornecer um suporte vasto à dispositivos de comunicação.

Outra característica relevante do Maxine é a sincronização labial, ela fornece suporte às línguas inglesa e espanhola. Para cada letra significativa de gesticulação da fala há um posicionamento labial presente no modelo 3D. Para cada palavra, o motor faz uma sequencia correta de gesticulações e periodicamente a animação é ajustada para não haver perda de sincronia. Além desses detalhes da fala, o motor aplica pequenas variações durante os movimentos da boca para

que o avatar pareça ser mais real. Ele tenta simular o ser humano, que não consegue manter um padrão mecanizado de gesticulação.

Apesar do Maxine ser uma ferramenta que consegue fornecer adequadamente o suporte desenvolvimento de interfaces com vários tipos de interações, ela foi elaborada para que apenas os recursos próprios sejam usados pelo desenvolvedor. A linguagem de script só permite a manipulação dos recursos deste motor. Portanto, se o desenvolvedor regular quiser inserir novos módulos ou substituir outros existentes é necessário haver uma programação mais baixo nível, que não é o objetivo do Maxine. O motor limita sua manipulação. Já no sistema apresentado por Uchino et al. (2007) é diferente. O motor também é uma tentativa de realizar a comunicação natural humana entre o homem e a máquina. Há o suporte ao reconhecimento e sintetização de voz e gestos e ambiente virtual. Mas o motor é mais modularizado.

O sistema elaborado por Uchino et al. subdivide seu funcionamento em módulos diferentes, onde cada um tem uma tarefa específica e funciona em um computador diferente. Um módulo é responsável pelo reconhecimento de padrões de imagens retiradas de uma câmera, outro é responsável pelo reconhecimento e sintetização de voz e um terceiro módulo é responsável pelo gerenciamento do ambiente virtual (modelo 3d e animações). Estes módulos se comunicam através de um hardware específico que cria uma rede entre os computadores, o que torna o sistema pouco portável. Além disso os módulos são fixos e também trabalha com tipos de hardwares específicos. O motor não fornece suporte a criação de novos módulos.

Outra proposta de interface homem-máquina é o AlexAvatar, o objetivo deste projeto foi desenvolver um um agente que conseguisse discutir e também aprender sobre um domínio específico enquanto mantem um diálogo com uma pessoa. Ou seja, seu motor faz uma tentativa de imitar ainda mais o homem em uma conversa através do fator aprendizagem. Para atingir este feito, sua estrutura para gerar conversas consta de um módulo de reconhecimento, um gerenciador de diálogos e um módulo de saída de áudio. O gerenciador de diálogos do AlexAvatar é subdividido em três partes: a desambiguação de discurso, que converte os sons para um conteúdo que possa ser compreendido e interpretado por uma pessoa; o gerenciador de dialogo baseado em contexto serve para gerar as saídas que serão faladas pelo avatar com base no gerenciador de conhecimento e pelas palavras-chave geradas pelo desambiguador de discurso. Além do sistema de Dialogo proposto, existe uma base para a construção de cenas 3D e layout, sintetizador de expressões e reconhecimento. Apesar disso, a forma de interação é mais limitada, pois o AlexAvatar apresenta apenas reconhecimento de voz, reprodução de voz sintetizada e imagem do avatar 3D. O motor não trabalha com o reconhecimento gestual presente na linguagem humana.

2.4 RELAÇÃO COM O TRABALHO PROPOSTO

A proposta da monografia tem como objetivo elaborar um motor genérico para o desenvolvimento de IHM baseada em interações com avatar. Por definição de motor, o sistema deve fornecer um ambiente de suporte ao desenvolvimento de um software. Portanto, o motor proposto deve fornecer um conjunto de funções que ajudem a elaborar a interface e as interações entre o avatar e o usuário. Só que para isso é preciso um mapeamento das necessidades do motor para ele que seja eficiente em sua tarefa. Para elaborar os requisitos e conceitos da proposta, foram usados alguns assuntos debatidos nas sessões anteriores (2.1, 2.2 e 2.3).

Um dos assuntos relevantes desta pesquisa foram os conceitos relacionados a suporte de desenvolvimento. Nos exemplos como Maxine (Seção 2.3), os motores fornecem recursos de entrada e saída, inteligencia artificial do avatar e ambientes virtuais. Eles dão a estrutura necessária para que o a comunicação natural (tanto gestual quanto verbal) possa ser elaborada. Esse é objetivo principal de uma IHM baseada em interações com avatar. Portanto, o motor proposto também deve ter esta característica. Deve fornecer recursos para elaborar o avatar e sua comunicação. Este sentido, pode-se subdividir um motor em três linhas de desenvolvimento: Avatar, Inteligencia Artificial e Reconhecimento de Linguagem Natural. Estes módulos podem ser definidos como:

- Avatar: O personagem virtual precisa realizar a comunicação humana. É necessário animações de movimentos corporais, gesticulações labiais e sintetização de vozes, portanto o motor precisa gerenciar um módulo relacionado apenas à funcionalidades da aparência do avatar. Essa parte deve fornecer ao personagem a sincronia entre lábios e voz, personalidade, emoções e outros recursos que tornem a interação mais próxima à humana;

- Inteligencia artificial: é a parte lógica de gerenciamento da comunicação e integração do software com a interface. Nessa etapa deve ser decidido o que o avatar vai falar, quando vai falar, qual funcionalidade deve ser executada. Essas informações são decididas com os recursos da entrada e reproduzidas com os recursos do avatar;

- Reconhecimento de Linguagem Natural: a entrada capturada de uma linguagem natural são imagens de gestos ou audios de voz. É necessário haver um processamento dessa informação para uma linguagem entendida pelo computador. Portanto um módulo com algoritmos de interpretação dessas entradas é essencial para uma IHM baseada em interações com avatar.

Apesar dessa divisão de módulos, é necessário haver uma integração das informações pas-

sadas entre eles. Por exemplo, se o sistema reconhece um gesto, interpreta e gera uma interação de saída, as etapas seriam executadas em módulos diferentes que não possuem os mesmos protocolos de comunicação. É necessário haver uma forma de integração independente destas estruturas. Nos motores pesquisados (Seção 2.3) há uma solução de padronização da comunicação entre os módulos (UCHINO et al., 2007). Porém os módulos são fixos e a comunicação é apenas realizada entre eles mesmos. Uma provável inserção de nova tecnologia nesse sistema causaria a alteração dos outros módulos para compreender suas funcionalidades. Ou seja, o sistema tem uma menor manutenabilidade. Ele é mais rígido a modificações. Nesse contexto, o trabalho proposto tenta apresentar uma solução para melhorar a generalização do motor. Tornar seus componentes substituíveis para que aumente a sua aplicabilidade.

Nessa mesma linha relacionada à generalização do sistema, pode ser citada a sincronia labial. Normalmente, avatares precisam ter uma sincronia nos lábios para se assemelhar à forma humana de comunicação. Esta gesticulação é fortemente ligada idioma. Os sons das letras variam de acordo com as línguas. Portanto, um motor de IHM baseada em interação com avatares deve tentar dar suporte a uma sincronia labial. No caso dos sistemas estudados, esse recurso se limita a gesticulações específicas de idiomas. Devido a isso, há uma limitação na aplicação do avatar a determinados públicos. Para tentar solucionar esse problema este trabalho busca apresentar uma solução de generalização da gesticulação para os idiomas.

3 VISÃO GERAL DO MOTOR PROPOSTO E DA PROVA DE CONCEITO

Este capítulo visa fazer uma apresentação geral que foi desenvolvido na monografia. Para realizar essa tarefa, o sistema do motor proposto é subdividido em partes resumidamente explicadas. Em seguida, é exposta uma forma de detecção de gestos utilizando câmera TOF e por fim é apresentado um modelo usado para prova de conceito, que utiliza o motor proposto e o reconhecimento de gesto discutido.

Como foi citado anteriormente, o objetivo principal da monografia era desenvolver um motor genérico de interface homem-máquina baseado em interações com avatar. Portanto alguns conceitos do motor foram criados e extraídos de outros trabalhos para suprir as necessidades de desenvolvimento de uma IHM. Com base nesse contexto, foi elaborada uma arquitetura (Figura 3.2) que aderisse e organizasse os conceitos da proposta. Resultado da arquitetura ficou seguinte forma:

- Gerenciador de Sistema: faz a administração geral do motor. Fornece um conjunto de funcionalidades básicas, gerenciamento de threads e módulos e comunicação entre partes isoladas do sistema. Além disso fornece o suporte à agregação ou substituição de outros módulos;

- Gerenciador de Sinais: a comunicação entre módulos isolados do sistema é feita através da emissão e recepção de sinais. O gerenciador do sistema é responsável por administrar esse processo;

- Módulo Reconhecedor: é responsável por capturar as formas humanas de entrada (voz e gestos), transformá-la em uma forma compreensível para o computador e repassar para o sistema;

- Módulo Lógico: é uma das principais partes do sistema, nele é desenvolvido a conexão

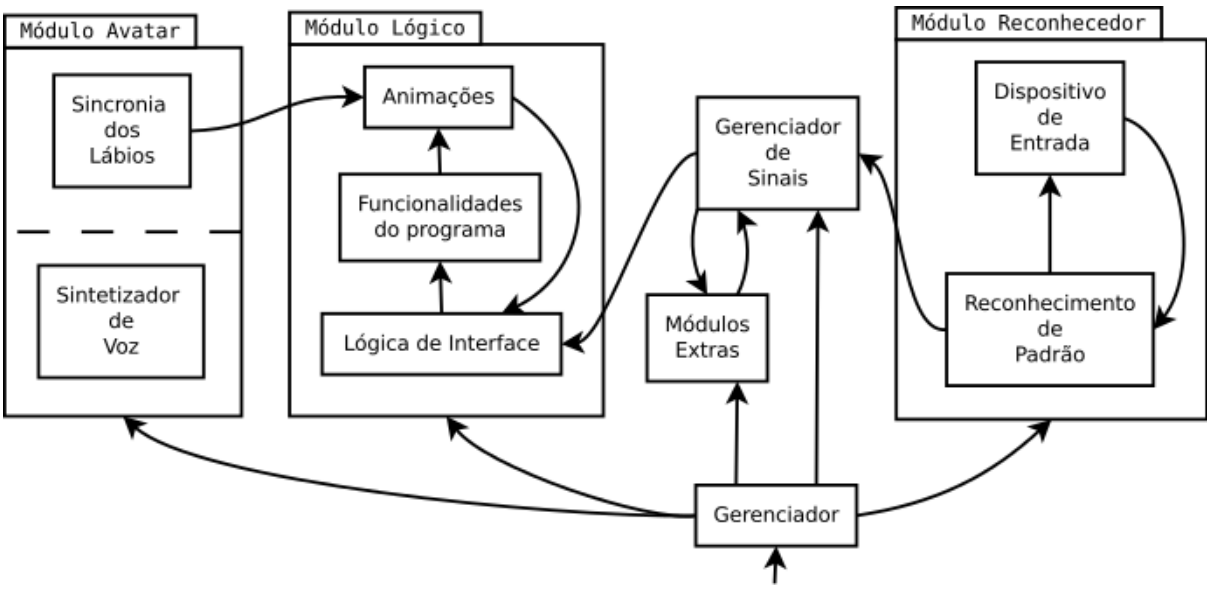

Figura 3.1: Arquitetura do motor proposto pela monografia. Existem três módulos importantes que são administrados pelo gerenciador: lógico, reconhecimento e avatar. O gerenciador de sinais auxilia a comunicação entre módulos diferentes que estão em threads separadas. O próprio desenvolvedor pode criar novos módulos (módulos extras) e associá-las a novas threads.

da interface com o as funcionalidades do sistema, é elaborado a inteligencia artificial do avatar e é gerado a saída (imagem e sons);

- Módulo do Avatar: é a parte do motor que implementa o funcionamento do avatar. Este módulo apresenta a sincronia de lábios e sintetização de voz. O módulo lógico e o gerenciador fazem a sincronização entre essas duas partes.

A arquitetura se diferencia das soluções encontradas devido ao gerenciamento de módulo. Qualquer nova funcionalidade deve ser um módulo isolado e utilizar os protocolos de comunicação presentes no gerenciamento do motor. Estes protocolos podem ser definidos por quem utiliza o sistema. Basta dar um nome como identificador e transmiti-lo junto a algum tipo de dado definido pelo desenvolvedor. Os dados emitidos não possuem uma limitação da plataforma. Isso é feito porque os módulos muitas vezes precisam passar diferentes tipos de informação de acordo com projeto. Mais detalhes sobre o funcionamento do motor pode ser visto no Capítulo 4.

Com a arquitetura discorrida acima, precisava-se uma prova de conceito do motor. Portanto, foi criado um modelo de aplicação que utilizava a arquitetura do motor proposto. Este modelo utilizou um avatar 3D, sintetização de voz, sincronia de lábios e o reconhecimento de gestos com base em um algoritmo existente (KOLLORZ; PENNE; HORNEGGER, 2008). A elaboração dessa etapa é discutida dos Capítulos 5.5 e 5.

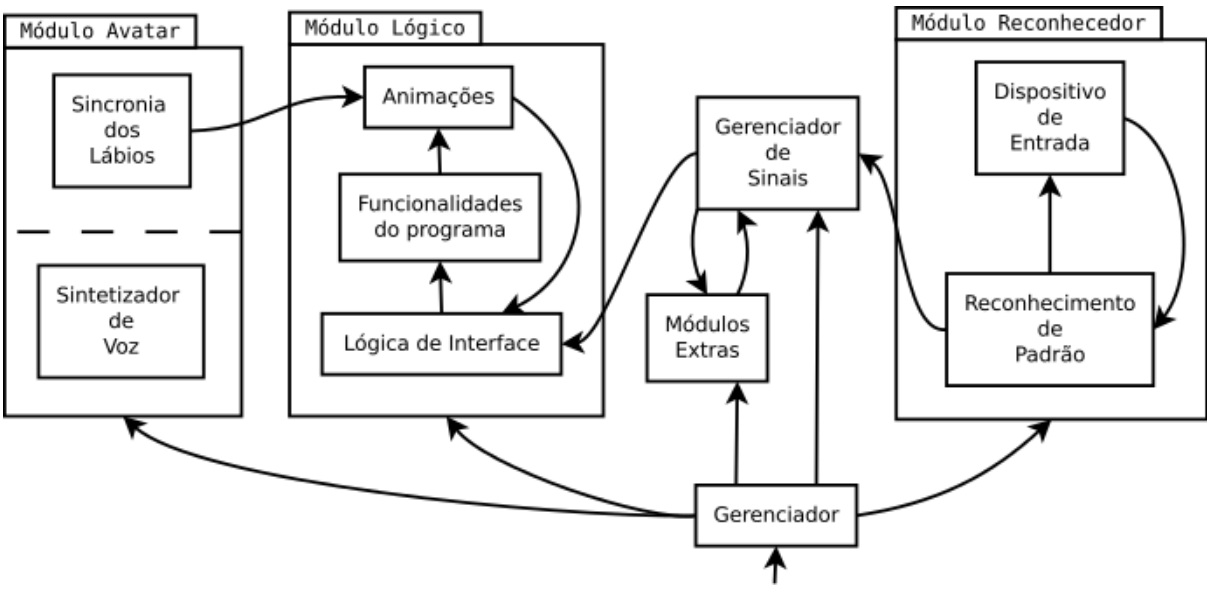

Figura 3.2: FIGURA DA PROVA DE CONCEITO USANDO O MOTOR E MODIFICANDO

4 MOTOR DE INTERFACE HOMEM-MÁQUINA BASEADA EM INTERAÇÃO COM AVATAR

Para desenvolver um aplicativo com interface homem-máquina braseada em interações com avatar, é preciso um sistema que forneça todos os recursos básicos para a implementação. Muitas vezes são funcionalidades de áreas distintas da computação que precisam ser integradas. Para isso, existe o motor. Ele é usado como uma camada de abstração destes recursos para facilitar o desenvolvimento. Este capítulo aborda todos os conceitos usados no desenvolvimento do motor de interface homem-máquina com avatar baseado no que foi pesquisado e exposto nos capítulos anteriores. Os assuntos sobre o motor proposto estão divididos em gerenciamento do sistema, comunicação entre módulos, módulo lógico, módulo de reconhecimento (dispositivo de entrada) e módulo do avatar.

4.1 GERENCIADOR DO SISTEMA

Em um sistema onde o avatar se comunica de forma semelhante à humana, é necessário haver execução de tarefas simultaneamente. O personagem precisa falar enquanto executa animações, reconhece voz ou gestos e desenvolve alguma tarefa como plano de fundo. Como foi citado anteriormente (Seção 2.1), o avatar deve passar a sensação de que está observando o usuário a todo momento. Parar uma animação para processar alguma atividade, pode não ser adequado. Acabará passando a sensação de um sistema não inteligente.

Para solucionar o problema de simultaneidade, o gerenciador do motor proposto apresenta um sistema de threads que cuida da inicialização, atualização, execução paralela e encerramento das funcionalidades de forma invisível ao desenvolvedor. Estas funcionalidades que precisam executar juntas são divididas em módulos por área de aplicação (reconhecimento de gestos, inteligencia artificial, avatar, etc). O gerenciador também apresenta formas substituir módulos existentes e de adicionar novos módulos extras à execução paralela. A arquitetura funciona

como um quebra-cabeça. O desenvolvedor pode ir incrementando, removendo e substituindo sua estrutura à medida que as necessidades do projeto vão aparecendo. Os módulos (peças) funcionam de forma separada.

Módulos podem funcionar como partes isoladas e independentes. Porém, apesar deste isolamento, muitas vezes é necessário que haja certa comunicação entre eles. A parte da estrutura que reconhece a comunicação natural, por exemplo, pode precisar informar à parte lógica da arquitetura a presença de algum gesto feito pelo usuário. Este tipo de situação é essencial para o funcionamento da IHM. Portanto o gerenciador administra parte da comunicação. Ele recebe e repassa informações dos módulos para o gerenciador de distribuição dela (Seção 4.3). Ou seja, o gerenciador do sistema funciona como um facilitador de acessos de recursos como a comunicação.

Além de prover o acesso a este recurso, o gerenciador também apresenta o suporte a outras funcionalidades necessárias ao desenvolvimento. Qualquer recurso básico que precise ser usado nas camadas dos módulos pode ser obtido através do acesso ao gerenciador. Isso é feito para evitar se criar dependências entre módulos e não tornar rígida a arquitetura do sistema.

4.2 EMISSÃO E RECEPÇÃO DE SINAIS

O módulos do motor precisam que haja uma comunicação de forma organizada. Muitas partes do software podem precisar distribuir e capturar informações sem que hajam conflitos. Inclusive módulos do sistema que dominam áreas de conhecimento diferentes e tem um protocolo de comunicação distintos precisam interagir. Em outras situações, mais de um módulo da aplicação precisam receber a mesma informação. Para contornar este problema, o motor possui estruturas de emissão e recepção de sinais, que possibilitam a troca de informação entre módulos distintos e bem separados.

Para realizar uma comunicação no motor proposto, a primeira etapa é a transmitir a informação. O módulo que precisa submeter algo emite um sinal passando o nome do sinal e um conjunto de dados. O nome do sinal serve como identificador e o conjunto de dados são informações extras utilizadas no processamento do sinal.

Os sinais emitidos são recebidos pelo gerenciador do sistema e repassados para o gerenciador de sinais, que é responsável por tratar e distribuir as comunicações realizadas entre o sistema. O gerenciador de sinais funciona como um disparador de eventos. Cada sinal recebido é emitido para que estiver ouvindo-o.

Como exemplo, pode-se citar o processamento da entrada de dados. O reconhecedor captura um gesto do usuário, transforma a informação em uma linguagem entendida pelo computador e emite um sinal informando o que o usuário fez. O gerenciador de sinais o recebe e o distribui para outros módulos. A partir deste ponto o módulo lógico captura e filtra o sinal, identifica que foi emitido pelo reconhecedor e decide o que deve ser feito a partir da determinada entrada.

4.3 MÓDULO DE RECONHECIMENTO

No motor de IHM com avatar que foi proposto, a linguagem precisa ser natural humana. O usuário não deve se adequar ao padrão imposto pelo dispositivo. O correto é a máquina fazer este papel: entender como o homem se comunica. Os dispositivos de entrada devem ser elaboradas com o reconhecimento de padrões de vozes, gestos ou expressões. Portanto uma das partes da estrutura do motor é o módulo de reconhecimento, ele é responsável por identificar a informação do usuário e repassar para módulos que tenham interesse, que interpretam os dados e geram uma resposta ao usuário.

Apesar das IHM normalemente se limitarem mais ao reconhecimento visual e sonoro, existe uma grande quantidade de dispositivos de entrada, que são escolhidos de acordo com a necessidade do projeto. O desenvolvedor pode usar uma ou mais câmeras, dispositivos TOF ou novas tecnologias para detectar a imagem por exemplo. O módulo de reconhecimento precisa ser maleável às adversidades do projetista. Para suprir essa necessidade de generalização, foi criada uma interface que o próprio desenvolvedor pode agregar um dispositivo ao software implementando as funcionalidades básicas de entrada requisitadas pelo gerenciador. Essas principais funcionalidades são: portabilidade de imagem; interpretação da entrada e atualização; emissão de sinais. Estas características definem um módulo reconhecedor.

Às vezes o software necessita que seja apresentada a imagem do usuário no monitor como ilustração para auxiliar comunicação. Mas a captura das imagens por um dispositivo qualquer é implementada sem haver compatibilidade alguma com o a parte gráfica do motor, pois são bibliotecas distintas, específicas para sua área e cada uma possui as próprias classes para implementar a imagem. A única possibilidade de apresentação da imagem é em janelas distintas. Apesar da incompatibilidade, várias bibliotecas usam padrões de representação de imagens comuns que podem ser facilmente convertidos de uma biblioteca para outra. No caso do motor proposto, o desenvolvedor precisa implementar uma interface para dá o suporte correto ao dispositivo. A integração é feita através da informação do tipo, tamanho e dados da imagem.

Existem vários formatos aceitos pelo motor. As principais características das representações do pixel da imagem são:

- Tamanho da representação: existe a definição de quantos bits são usados para representar uma cor ou uma das componentes da cor. Podem ser usados dois, três, quatro, cinco, seis, oito, dezesseis, vinte e quatro ou trinta e dois bits;
- Formato da representação: são aceitos os formatos RGB com ou sem transparência, apenas transparência, apenas luminância, apenas um componente isolado do RGB, profundidade e outros formatos menos conhecidos como PowerVR e DirectDraw Surface do DirectX.

Implementando essa funcionalidade, o motor poderá trabalhar com imagens na mesma tela aplicando-a como textura de um plano no ambiente virtual ou aplicar alguma transformações sobre ela.

Mas mesmo com esta compatibilidade, em uma IHM baseada em interações com o avatar, os tipos de entrada capturados (imagens e sons) do usuário não conseguem ser compreendidos pelo computador. A máquina não consegue entender que o ato de levantar uma mão na imagem possa ser entendido como um sinal de *parar* por exemplo. A principio ela nem consegue distinguir o que é uma mão. É necessário que haja uma interpretação dos gestos através de algoritmos de reconhecimento de padrões para tentar identificar a ação. Primeiro o computador precisa tratar a imagem para identificar regiões de interesse, depois procurar pela mão e por fim classificar o padrão como um gesto de parar. Além disso, é necessário que periodicamente seja verificada a atividade do usuário, pois muitas ações são denominadas por um gesto em movimento. A comunicação natural humana é realizada de forma dinâmica e contínua. A qualquer momento pode ocorrer um gesto que mude a decisão lógica da máquina. Nesse contexto, o reconhecedor deve atualizar seu resultado de reconhecimento constantemente. Inclusive quando estiver executando outras tarefas em paralelo. Essas informações devem ser sempre transmitidas para o resto do sistema no intuito de manter o comportamento do personagem condizente com a ação realizada pelo avatar.

4.4 MÓDULO LÓGICO

Para a elaboração de um software que utiliza IHM baseada em avatar, é necessário que haja uma parte lógica do personagem, onde as decisões sobre execução, falas e animações precisam ser tomadas. Com o intuito de solucionar este problema foi criado o módulo lógico.

Esta estrutura é responsável por controlar todo fluxo de execução de atividades do sistema. Nele pode-se implementar inteligencia artificial, leitura dos comandos de entrada e controle de animações e sons. No geral ele recebe a entrada interpretada pelo módulo de reconhecimento, decide o que deve ser feito pelo sistema e produz as saídas corretas.

As estruturas de decisão de uma IHM são as principais partes do software, pois é onde o serviço definido pelo desenvolvedor é ligado e administrado pela interface. O módulo lógico é responsável por analisar os dados de entrada, analisar os serviços da aplicação e decidir o comportamento da interface, enquanto todos os outros módulos são usados apenas para suprir o funcionamento da interface de comunicação natural. Ou seja, o módulo lógico é a parte inteligente do sistema que liga a interface e o serviço. Este módulo pode ser elaborado usando técnicas de inteligencia artificial ou sistemas puramente reativos a alguma interação humana. Pode-se aplicar recursos de expressão de sentimentos e análise de diálogos, gerenciamento da disposição dos modelos e ambientes 3D em relação à cena, seleção de sons e qualquer outra funcionalidade que influencie na resposta ao usuário. É importante ressaltar que esse módulo deve ser usado apenas como a porta de comunicação com os serviços. As funcionalidades do software implementado com o motor devem ter uma arquitetura à parte.

4.5 MÓDULO AVATAR

Como a proposta do trabalho é um motor de interface homem-máquina baseada em interações com avatar, existe a necessidade de funcionalidades que ajudem na elaboração do personagem. É preciso um módulo que tenha os recursos necessários para que a interação entre o homem e a máquina fique mais natural. Nesse contexto, foi elaborado o módulo do avatar para que agregasse estas funcionalidades do personagem.

Uma das tentativas de tornar a animação do personagem mais natural é fazer a sincronia labial. Para desenvolver este recurso, em primeiro lugar deve-se ter um modelo manipulável. O personagem deve poder fazer movimentos da boca e criar gesticulações das palavras. Assim pode-se tornar possível animar o avatar.

Apesar a possibilidade de movimentação, em uma sincronia labial, é necessário se saber os pontos em que os elementos da boca podem se posicionar. Para cada palavra falada deve haver certas gesticulações específicas que fazem o personagem parecer pronunciar a palavra correta. Ou seja, os lábios e mandíbula devem se movimentar até certos pontos para fazer a animação labial ideal. Para definir esses pontos em que a boca deve se movimentar, são criados estados em que se sabe as regiões do modelo do personagem que foram deslocadas. No caso da sincronia

labial, esses estados são posições dos lábios. Em um modelo 3D do avatar, deve ser criado alguns pontos de controles (shape keys) que são predeterminados no momento da modelagem. Cada shape key contem um conjuntos de pontos pertencentes ao modelo e um deslocamento aplicado sobre ele. Um exemplo desses pontos de controle é boca aberta. Em um avatar que inicialmente está em estado de repouso (boca fechada), o modelador deve selecionar regiões próximas à mandíbula, tracioná-los para baixo até a posição ideal de boca aberta e adicionar o estado atual da malha a um shape key. Dessa forma as posições podem ser indexadas e usadas em várias bibliotecas gráficas.O motor proposto faz a sincronia de lábio através dos pontos de controle. Para cada posição de alguma gesticulação relevante é feito um shape key. Depois todas as posições são carregadas pelo módulo do avatar e usadas para fazer a sequencias de gesticulação.

Apesar de poder se fazer a sequencia correta de shape keys para uma determinada palavra serem mostradas em determinado espaço de tempo, uma transição direta não passa a sensação de continuidade pois o ponto de controle é estático. Ele é apenas um posicionamento de elementos do avatar, não representa uma animação. Para solucionar esse problema de descontinuação, é usado um fator de multiplicação no momento da expressão do shape key (Figura 4.1). Este fator é expresso entre zero e um, onde o zero indica que nada de um estado de repouso para a posição do shape key é movido e um indica que todo o shape key deve ser apresentado.

Para realizar a suavidade na gesticulação, uma variável que cresce linearmente no tempo é multiplicada pelo shape key. Enquanto a variável faz com que a expressão facial de um determinado ponto de controle apareça, o shape key anterior é multiplicado pela inversa do atual, ou seja, a gesticulação anterior vai decrescendo em paralelo. Ocorre a transformação de uma expressão facial em outra de forma suave.

Mesmo com a solução de suavidade de gesticulação, as animações labiais ainda não ficam ideias. Existem gesticulações diferentes para sons diferentes da fala. Apenas um tipo de gesticulação ou gesticulações não correspondentes com o som não são adequados para a sincronia labial. É necessário que o personagem movimente a boca de acordo com o som que é produzido pela sintetização de voz.

Na criação das gesticulações, tentou-se elaborar formas genéricas da posição dos lábios para aumentar a gama de linguagens do sistema. Foram selecionados artigos que trabalham com sincronia de lábios e apresentavam imagens do posicionamento labial. Das figuras foram identificados padrões de postura da boca presentes em todos os trabalhos. Por exemplo, o padrão dos lábios para a pronuncia do ’u’ em inglês (OH et al., 2010) é parecido com o padrão do ’o’ e é comum a algumas gesticulações em espanhol (BALDASSARRI; CEREZO; SERON, 2008) e

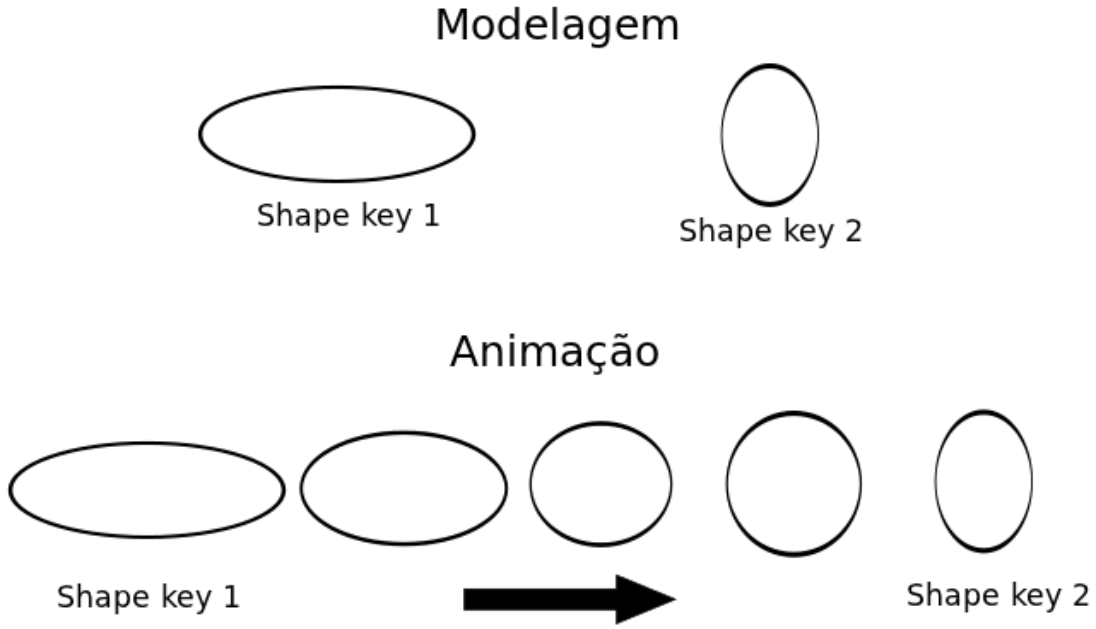

Figura 4.1: Processo de elaboração da animação labial. Primeiramente o personagem tem os shape keys modelados em um programa (Shape key 1 e Shape key 2). Depois os shape keys são usados como pontos de inicio e fim da animação. A transformação entre estes pontos de controle é realizada com um fator de multiplicação. Enquanto o fator do Shape key 2 cresce de 0 para 1, o fator do Shape key 1 decresce de 1 para 0. Esta transformação torna a animação mais suave.

chines (HUANG; YIN; ZENG, 2005), ou seja, a posição labial do ’u’ tem um grande potencial para ser um padrão comum.

Foram identificados quatro padrões diferentes para o posicionamento dos lábios:

- Repouso: usada para momentos em que não se fala nada ou transição entre duas animações labiais. Nesse estado, que é comum a todas as linguagens, a boca se mantem totalmente fechada e os lábios encostados um no outro;
- Abertura Pequena: posicionamento em que os lábios tendem a fazer formas esféricas que deixam a abertura da boca pequena. É semelhante à pronuncia do ’o’ ou do ’u’ em inglês;
- Abertura Média: leve abertura dos lábios onde algumas vezes mostra-se os dentes e outras não. Assemelha-se à pronuncia do ’a’ ou ’e’ em inglês;
- Abertura Grande: maior abertura utilizada na linguagem onde normalmente não se mostram os dentes. Pode ser representada pela pronuncia do ’i’ em inglês.

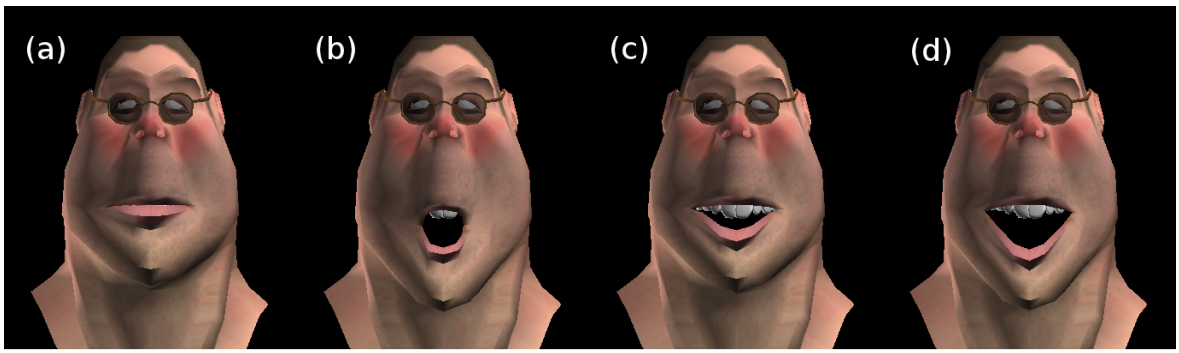

Figura 4.2: Padrões labiais do avatar do motor. A partir dos padrões selecionados de outros trabalhos, foram desenvolvidos os shape keys de (a) Repouso, (b) Abertura Pequena, (c) Abertura Média e (d) Abertura Grande, respectivamente da esquerda para a direita.

Cada padrão de posição labial citado acima é representado por um shape key da animação. Sobre eles são aplicadas as regras de suavidade e transição mencionados anteriormente. Estes passos resolvem o problema de suavidade da animação, porém há a necessidade de fazer uma animação contínua com um conjunto de shape keys quando o avatar falar alguma frase.

A implementação da sequencia de gesticulação para sincronizar com a voz sintetizada é feita com a seleção de letras mais expressivas da língua que possam substituir um bloco de letras da palavra por um posicionamento labial. Ou seja, uma palavra qualquer que possua duas letras mais expressivas terá dois blocos de gesticulação no momento da sincronia labial. Um exemplo dessa solução pode ser visto na Figura 4.3.

As letras dominadoras tem maior significância no posicionamento labial quando uma pessoa fala. Portanto essa solução tenta chegar próxima à gesticulação real através da apresentação da sequencia de animações mais relevantes. O desenvolvedor seleciona o conjunto de letras expressivas e classifica-as entre os padrões comuns do shape key. Precisa-se identificar se determinada vogal ou consoante relevante pertence aos conjuntos abertura pequena, abertura média ou abertura grande. Como foi sitado anteriormente o padrão de repouso é usado apenas para momentos onde não se fala nada ou há uma transição entre animações de letras dominantes.

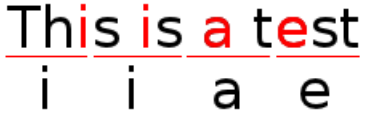

Figura 4.3: Exemplo de sincronização da gesticulação do personagem virtual com o sintetizador de voz. As letras expressivas (em vermelho) do texto utilizado pelo sintetizador são separadas e classificadas em padrões labiais. No momento da fala estas letras são usadas para fazer a gesticulação de um bloco de letras (sublinhado vermelho).

Essa solução para gerar a animação da sincronia de lábios não leva em conta o tempo que cada letra significativa deve ter na fala. Ela apenas determina um intervalo de tempo constante para todas as transições entre shape keys. Só a velocidade geral pode ser alterada para ficar próxima à velocidade de sintetização de voz. Com isso, à medida que o discurso é realizado, o avatar tende a perder a sincronia. Portanto, é recomendável usar essas animações para trechos de frases curtos.

4.6 DESENVOLVIMENTO COM O MOTOR

A implementação de uma interface homem-máquina baseado em interações com avatar tende a ser complexa. Há várias funcionalidades que precisam ser executadas simultaneamente e se comunicar entre elas. Consequentemente, um motor para desenvolver essa interface tende a ser complexo também. Ele precisa organizar todas esses recursos para não haver conflitos. Por causa disso, foi criado um conjunto de passos para se elaborar uma interface com base na arquitetura proposta. A seguir são explicados os passos:

- Definição do Sistema: antes de dar inicio à codificação, é recomendável definir as funcionalidades do sistema, possíveis animações do avatar, diálogos, gestos a serem reconhecidos, dados de comunicação entre módulos e outros fatores que influenciem em algoritmos mais complexos. Por exemplo, o reconhecimento do gesto da mão exige um equipamento específico e lógica de funcionamento não trivial;

- Criação Estrutural: com uma especificação do sistema, o desenvolvedor deve implementar um módulo de reconhecimento, onde utiliza-se dispositivos de entrada para capturar os gestos ou voz e aplica-se algoritmos para classificar a informação. Depois é necessário desenvolver um módulo lógico para organizar a ordem de apresentação e interação do sistema. Caso necessário, pode-se também implementar o suporte sintetizador de voz utilizando as funções do Gerenciador;

- Animações: um avatar já fornecido pelo motor pode ser usado na interface ou o desenvolvedor pode criar um novo personagem seguindo os padrões de animação. Caso seja usado a sincronia de lábios com voz sintetizada, o desenvolvedor precisa identificar as letras de expressividade e classificá-las nos padrões de animações labiais.

4.7 DISCUSSÃO

O capítulo apresentou a arquitetura proposta para um motor de interfaces baseada em avatar. Esta arquitetura contem o gerenciador do sistema, que organiza os módulos, fornece recursos básicos de desenvolvimento e administra as comunicações entre módulos. Toda estrutura tem como objetivo fornecer um conjunto de funcionalidades para a elaboração de uma IHM com avatar. Portanto, para validar este modelo, é necessário que haja um exemplo construído sobre a arquitetura como prova de conceito da monografia. O exemplo precisa apresentar uma IHM baseada em interações com avatar que utilize pelo menos parte da comunicação natural humana. A seguir, é apresentado os conceitos de um sistema para comprovar o funcionamento da arquitetura discutida neste capítulo.

5 PROVA DE CONCEITO

A arquitetura do motor apresentada no trabalho serve para desenvolver interfaces homem-máquina baseadas em interações com avatar. Como forma de comprovar o funcionamento do motor proposto, este capítulo apresenta um exemplo simples de comunicação homem-máquina, onde o usuário se utiliza de gestos da mão para interagir com o aplicativo e o software utiliza voz sintetizada e animações de um personagem virtual para interagir com o usuário.

5.1 FUNCIONAMENTO

Na prova de conceito, o usuário deve escolher uma das aplicações apresentadas pelo avatar através de um gesto. O personagem informa quais aplicações podem ser abertas e quais os seus respectivos gestos para executar a tarefa. O usuário deve posicionar a mão em frente ao dispositivo de entrada (reconhecimento) e articulá-la para ficar de acordo com o padrão dos dedos da aplicação desejada. Este processo de execução deve ser organizado corretamente pelo desenvolvedor. Por exemplo, o sistema não pode começar a execução de um programa enquanto o avatar está falando alguma informação, pois o usuário precisa saber quais comandos ele pode usar para os programas. Portanto, o software deve ser dividido em estados de tarefas do avatar.

Com a análise do fluxo de execução do programa, ele foi dividido em três estados: apresentação de aplicações, reconhecimento de seleção da aplicação e execução. A primeira etapa é falar a lista de programas que podem ser executados, depois identificar qual software foi selecionado e por fim executar a escolha. O automato de execução da aplicação é apresentado na Figura 5.1.

5.2 PONTOS DE CONTROLE

Mesmo com os estados bem definidos, é preciso saber qual o momento de ocorrer uma transição de um para o outro. Cada estado precisa receber a informação de que é o momento

de encerrar devido a algum evento. Estes acontecimentos podem ser identificados como fim de fala do personagem, reconhecimento de algum gesto de entrada específico e outros exemplos. Ou seja, isso depende da implementação da IHM. Portanto foi criando um sistema de controle do início, execução e fim dos estados do software. Eles são definidos da seguinte forma:

- START (início): é responsável por dizer que um determinado estado está sendo executado pela primeira vez em sua requisição atual. Quando ocorre uma transição, o estado que foi selecionado sempre passa por este ponto de controle;
- DOING (fazendo): normalmente os estados nunca terminam com apenas uma única iteração no programa, muitas vezes precisam esperar a finalização de tarefas mais demoradas como reconhecimento e sintetização de voz ou precisam executar tarefas mais longas. Este ponto de controle serve apenas para dizer que o estado ainda não terminou;
- DONE (feito): é informado sempre que a tarefa é finalizada e deve executar seus últimos detalhes. É neste momento que ocorre a transição entre estados.

Estes pontos de controles são manipulados pelos próprios estados, pois eles que sabem identificar quais tarefas devem ser executadas e o momento exato.

5.3 DIÁLOGOS

O andamento da execução da interface também precisa ser repassada para o usuário. Ele precisa saber qual os softwares podem ser executados, qual o momento em que ele pode tentar abrir o programa e se o gesto foi realmente reconhecido. Para isso, esta interação informativa é feita por fala do avatar.

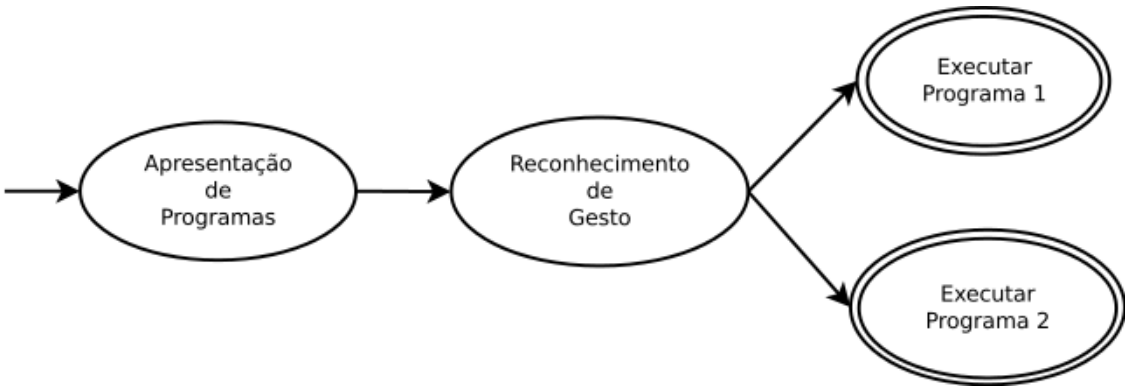

Figura 5.1: Automato de execução do sistema. Primeiramente o avatar apresenta os programas fornecidos através de figuras e fala, depois tenta fazer o reconhecimento do gesto feito pelo usuário e por fim é aberto o aplicativo 1 ou aplicativo 2 a partir da decisão do reconhecimento.

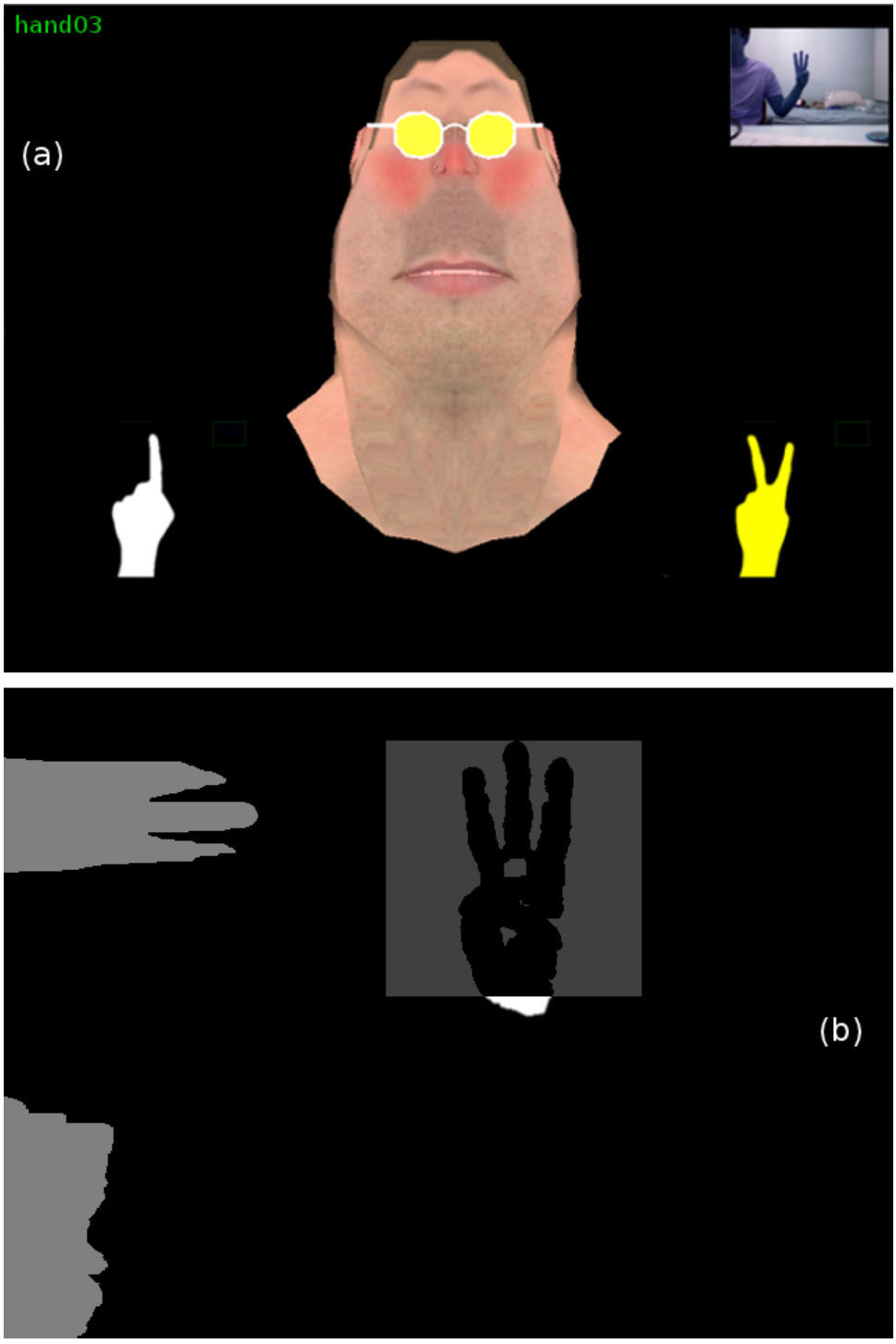

Figura 5.2: Imagem da prova de conceito do motor proposto. Em (a) é apresentado a interface visual do sistema. Ele tem um avatar 3D no centro, os gestos aceitos pelo programa nos cantos inferiores e a imagem da câmera no canto superior direito. Em (b) pode-se ver o sistema detectando a mão com o gesto de três dedos. Esta tela foi captura durante um teste do programa e (a) e (b) estão executando simultaneamente. O gesto de três dedos foi detectado e mostrou ”hand03” na parte superior esquerda de (a). Como não é um gesto referente a um aplicação, nada é executado.

O primeiro estado é a listagem de softwares. O programa precisa listar as aplicações disponíveis, os respectivos gestos e informar que o usuário deve fazer algum dos gestos. Assim que a frase termina ocorre a transição e o motor tenta apenas reconhecer. A frase selecionada é:

”Faça o gesto da esquerda para executar o *aplicativo*1 ou o gesto da direita para executar o *aplicativo*2.”

Onde *aplicativo*1 e *aplicativo*2 são os nomes dos programas usados. Como o sintetizador selecionado para elaborar o exemplo não tem suporte para português. A frase em inglês fica:

”Do the left gesture to execute *aplicativo*1 or the right gesture to execute *aplicativo*2.“

Após a listagem dos softwares, é feito o reconhecimento de gestos da mão. Este estado identifica qual o programa deve ser aberto a partir de uma postura da mão. Depois de entender o gesto, o computador precisa informar o usuário que compreendeu a informação e que vai abrir o programa. Portanto o avatar fala a seguinte frase:

”Ok! Executando o programa.“

Em inglês fica:

”Ok! Executing the program.“

5.4 SINCRONIA DOS LÁBIOS

Os diálogos do personagem já definidos precisam estar integrados à sincronia labial. Esta sincronia é feita com o módulo do avatar presente no motor proposto (Seção 4.5). Como foi definido anteriormente, é necessário selecionar um conjunto de letras expressivas da fala para criar a animação de gesticulação do personagem. O desenvolvedor deve agrupa-las de acordo com o tamanho da abertura de boca (Figura 4.2). O mapeamento das letras ficou definida da seguinte forma:

- abertura pequena: ’o’ e ’u’;
- abertura média: ’a’ e ’e’;
- abertura grande: ’i’.

5.5 RECONHECIMENTO DE GESTOS DA MÃO E ANÁLISE DE DESEMPENHO

A prova de conceito utiliza o reconhecimento de gestos da mão como forma do usuário interagir com o avatar. Para fazer este reconhecimento, foi usada a modificação de um algoritmo já existente. Na solução, o autor recorre à uma câmera time-of-flight como dispositivo de entrada para capturar a imagem do ambiente. O processo de extração de features da mão é realizada exatamente como no artigo, mas a classificação é diferente. A monografia apresenta uma proposta de modificação para tentar melhorar a categorização do gesto. Esta solução foi agregada ao módulo reconhecedor da prova de conceito.

5.5.1 ALGORITMO

Para o computador entender a linguagem humana, é necessário que ele consiga compreender os formas de expressões do homem. No domínio de reconhecimento de gestos da mão, por exemplo, o computador precisa identificar diferentes simbologias que seu posicionamento representa. A máquina precisa entender o significado do gesto. Porém, antes de conseguir definir o este significado, o computador precisa identificar o que é uma mão e distinguir seus diferentes gestos, pois este recurso não faz parte da linguagem natural da máquina. É necessário usar algoritmos de processamento de imagem e reconhecimento de padrões para compreender a entrada.

Com este objetivo de identificar o gesto da mão, existe um algoritmo que tenta reconhecer gestos da mão utilizando uma câmera time-of-flight (KOLLORZ; PENNE; HORNEGGER, 2008). Esta solução divide todo o processo de reconhecimento em cinco etapas. Na primeira fase deve haver a segmentação da mão e braço a partir de uma determinada distancia. Por exemplo, objetos que estejam próximos ao dispositivo são detectados, caso estejam distantes são ignorados. Desse modo basta estender o braço para a região de proximidade para isolar a parte de interesse. O autor enfatiza que o método não funciona se houver algo a mais do que a mão e o braço na imagem segmentada. Como nessa proposta há o uso de uma câmera time-of-flight, o ruído do plano de fundo pode ser eliminado facilmente.

Na segunda etapa deve-se determinar uma área quadrada mínima onde se localiza a mão e o braço (bounding box). Este processo consiste em calcular as projeções dos eixos x e y, respectivamente chamadas de P_x e P_y, e escolher as das extremidades de cada eixo projetado. Ou seja, a largura do bounding box é definida pelas projeções mais à esquerda (x_{min}) e mais a direita (x_{max}) do P_x, no caso de P_y considera-se a mais acima (y_{min}) e mais abaixo (y_{max}) (Figura

5.3).

De forma simplificada, uma projeção é a diferença entre o maior e o menor valor que ocorre em uma linha e a projeção de um eixo é um vetor de projeções para cada posição do eixo. Ou seja, cada valor do eixo (altura e largura) é o índice da projeção.

A próxima etapa consiste em fazer a extração da mão separando-a da região do braço. Um índice (y_{new}) na projeção P_y é calculado para definir a posição de corte. Para achar esse valor é necessário projetar uma mão de tamanho l mm a uma distância cog_z (cog: centro de gravidade do objeto segmentado; z: direção da câmera). A fórmula aplicada é:

$$y_{new} = y_{min} + \frac{12.0 \cdot l}{cog_z \cdot 0.04} \tag{5.1}$$

Onde 0.04 é o tamanho do pixel e y_{min} é a posição mais alta da projeção. Este tipo de segmentação pode ser aplicada em imagens de mãos que possuem anéis, relógios ou pulseiras.

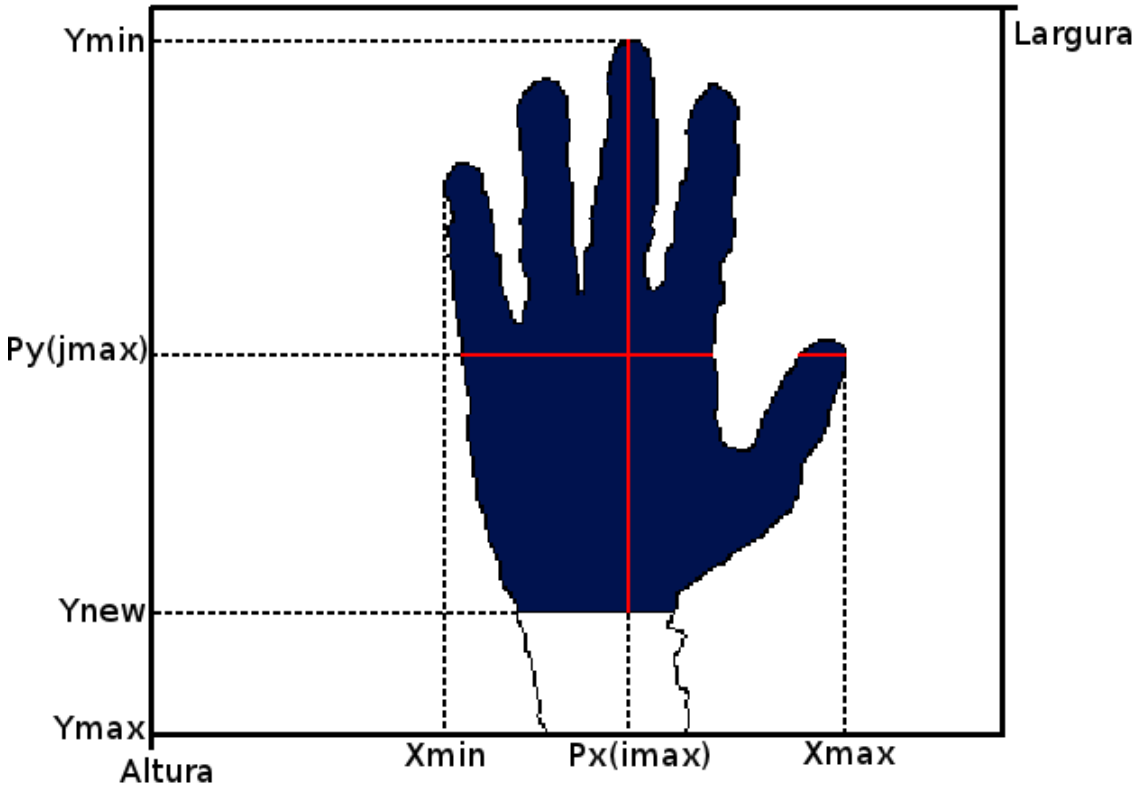

Figura 5.3: Exemplo da extração da região da mão de uma imagem. P_x e P_y são vetores de projeções (distancia entre *min* e *max*) da região de interesse nos eixos x e y, respectivamente. Os índices (x_{min},x_{max}) e (y_{min},y_{max}) indicam a região onde tem a maior projeção e delimitam o bounding box. O y_{new} define a divisão da região entre a mão e o antebraço. Ele substitui o y_{max} para a correta delimitação da região da mão.

A quarta etapa aplica sobre as projeções da região da mão um filtro para suavizar as bordas. Assim fica mais fácil de encontrar semelhanças nos padrões de mão. Depois, para cada pixel pertencente a mão é calculado a profundidade mínima, máxima e média, respectivamente chamados de a_{min}, a_{max} e a_{avg}. É importante ressaltar que outros pontos não pertencentes à mão mas que estão dentro do bounding box não são usados.

Na etapa final é feita a classificação da mão. Para cada gesto do banco de classificação, já devem estar guardadas as suas projeções suavizadas dos eixos e as suas profundidades minímas, máximas e médias. O gesto que será classificado e os gestos de referência (R) usado para classificar devem estar nas mesmas dimensões. Os seguintes passos são aplicados para achar a semelhança entre as mãos:

- Achar o índice da projeção do gesto de referencia equivalente ao índice dos gestos a ser classificado: para isso a formula abaixo deverá ser usada. O exemplo é para P_x, mas deverá ser aplicada a P_y também.

$$c = \frac{i - x_{min}}{x_{max} - x_{min}} \cdot (x_{R,max} - x_{R,min}) + x_{R,min} \tag{5.2}$$

  Onde i é o índice da projeção a ser testada (P_x); c é o índice da projeção de referencia $P_{x,R}$; $x_{R,min}$ e $x_{R,max}$ são os índices mínimo e máximo de $P_{x,R}$. Se c é um numero flutuante, deve a haver uma interpolação para tornar o numero do índice inteiro;

- Calcular a diferença da distância entre as projeções do eixo: é feito somatório da diferença das distâncias. Quanto menor o resultado da fórmula, mais próximos estão as projeções.

$$x_{diff} = \sum_i \left| \frac{p_x(i)}{p_x(i_{max})} - \frac{p_{x,R}(c)}{p_{x,R}(i_{max,R})} \right| \tag{5.3}$$

- Onde $p_x(i_{max})$ e $p_{x,R}(i_{max,R})$ são as posições de cada projeção que possuem o maior valor observado. A fórmula acima está calculando apenas P_x, mas deve também ser aplicada à projeção P_y. Calcular a distância entre as mãos: a fórmula utilizada calcula a soma das distâncias normalizadas da profundidade, P_x e P_y.

$$d = \left| \frac{a_{avg} - a_{min}}{a_{max}} - \frac{a_{R,avg} - a_{R,min}}{a_{R,max}} \right| + \frac{x_{diff}}{x_{max} - x_{min}} + \frac{y_{diff}}{y_{max} - y_{min}} \tag{5.4}$$

  Quanto menor a distância, mais semelhantes são as mãos. Pode ser visto que a mão a ser testada e a mão de referencia vem sendo normalizadas para o mesmo espaço desde a fórmula anterior. Todos os paços anteriores devem ser aplicados para algumas sequencias de quadros para ter certeza de que é o gesto procurado.

Na solução explicada acima o autor fez testes de classificação com e sem o uso da profundidade para calcular a distancia mínima entre gestos. Ou seja, apenas usou ou omitiu o a_{min}, a_{max} e a_{avg}. Em seus resultados, o calculo sem profundidade teve um desempenho ligeiramente inferior. Para um conjunto de 408 e imagens de 34 pessoas diferentes, apenas 8 fotos (aproximadamente 1%) foram acertadas a mais no teste que usou profundidade.

Neste algoritmo discutido, calcula-se a distancia entre todos os gestos e considera o mais próximo como o gesto correto da mão. A solução usa apenas uma classificação (a menor distância) para distinguir um gesto do outro. Não usa mais de um classificador para identificar uma mão. Para tentar melhorar o desempenho, foi feita uma modificação neste algoritmo. A nova solução utiliza todos as outras distancias calculadas entre mãos como classificadores. Ou seja, um gesto de teste deve estar em um intervalo de distancia $d1$ do gesto classificador 1, em um intervalo de distancia $d2$ do gesto classificador 2 e assim por diante. O uso da comparação de intervalos de distancia de vários classificadores tenta refinar e confirmar o processo de seleção.

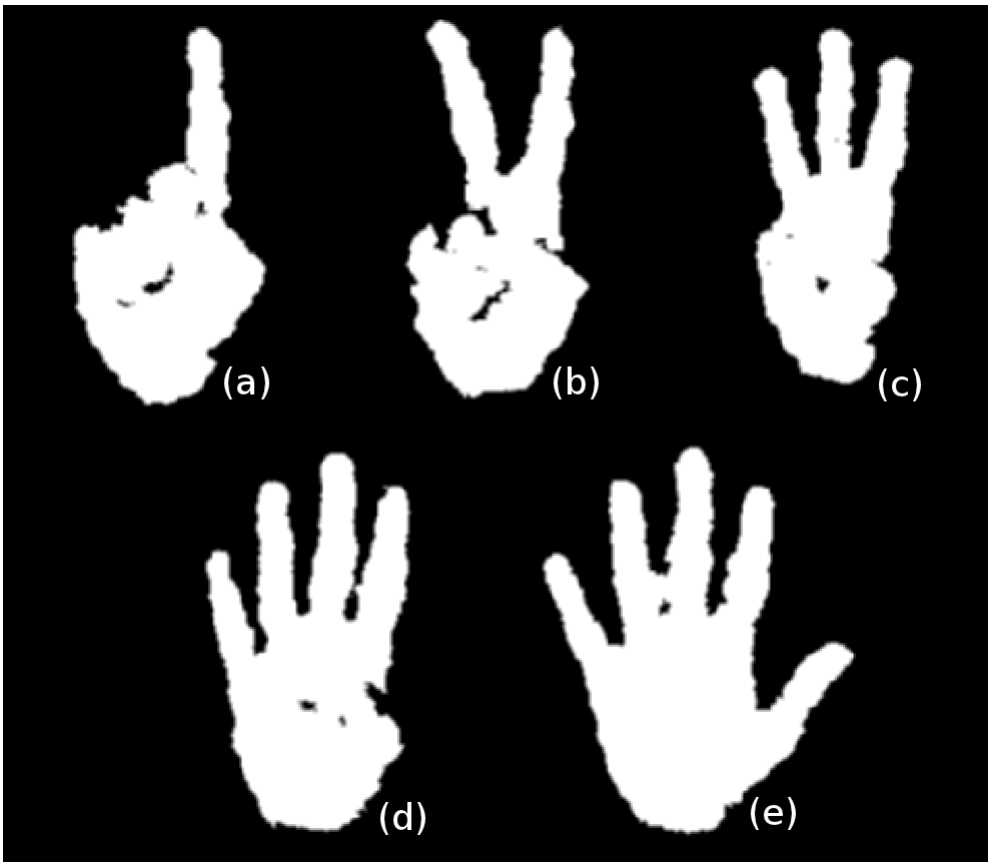

Figura 5.4: Imagens dos gestos reconhecidos pelo teste do algoritmo proposto. É possível reconhecer gestos com (a) um dedo, (b) dois dedos, (c) três dedos, (d) quatro dedos e (e) cinco dedos.

Tabela 5.1: Resultado do teste de reconhecimento de gestos. As colunas indicam os gestos reconhecidos e as linhas indicam os gestos esperados.

	Gesto 1	Gesto 2	Gesto 3	Gesto 4	Gesto 5
Gesto 1	0	0	0	0	0
Gesto 2	0	0	0	0	0
Gesto 3	0	0	0	0	0
Gesto 4	0	0	0	0	0
Gesto 5	0	0	0	0	0

5.5.2 TESTES

Um algoritmo de reconhecimento de gestos da mão precisa se mostrar eficiente em sua proposta. Ele precisa provar que consegue identificar de forma satisfatória as mãos e tenha uma baixa taxa de erro. Para tentar comprovar esta eficiência, é necessário que sejam feitos testes de reconhecimento. Portanto, para fazer essa análise da modificação proposta do algoritmo de reconhecimento, foram selecionados gestos para teste (Figura 5.4).

Para testar o algoritmo foram selecionadas X imagens de gestos de Y pessoas. Na Tabela 5.1 pode ser visto os resultados dos testes.

5.5.3 DISCUSSÃO

6 DISCUSSÃO E CONCLUSÃO

O objetivo do trabalho foi elaborar um motor de interface homem-máquina baseada em interações com avatar. A arquitetura da proposta precisava se adequar a possíveis substituições e adições de outras tecnologias para auxiliar o sistema, entre elas reconhecimento de gestos, inteligência artificial, etc. Através da prova de conceito, este objetivo foi alcançado, pois o modelo desenvolvido com o motor consegue fazer o reconhecimento de gestos da mão, fazer um avatar se comunicar através de voz e há a presença de sincronia labial. A aplicação consegue utilizar parte da comunicação natural. Mas para chegar a essa meta, foram necessárias a implementações de alguns outros recursos: reconhecimento de gestos da mão e sincronia labial. O reconhecimento gestual foi elaborado com a modificação de um algoritmo existente. Os resultados obtidos pelo teste foram satisfatórios para uma prova de conceito, portanto a solução foi utilizada. A sincronia labial também se mostrou satisfatória para a integração da voz sintetizada e da gesticulação do personagem, então também foi usada.

6.1 TRABALHOS FUTUROS

Apesar de haver uma prova de conceito como forma de avaliar a arquitetura, seria adequando haver testes de usabilidade do sistema para comprovar sua eficiência. A realização de teste de desempenho e substituição de módulos seria o ideal para verificar a viabilidade do sistema.

Não só a arquitetura, mas também as soluções de reconhecimento de gestos e sincronia labial precisam ser melhor estudadas. O reconhecimento precisa de um treinamento mais adequado, visto que o refinamento de cada posição da mão foi elaborado manualmente. O ideal seria utilizar técnicas com maior embasamento científico. Já a sincronia labial precisaria de um teste em larga escala. É necessário, pegar vários conjuntos de frases distintas e analisar a opinião de vários usuários sobre a sincronia.

REFERÊNCIAS BIBLIOGRÁFICAS

BALDASSARRI, S.; CEREZO, E.; SERON, F. J. Maxine: A platform for embodied animated agents. *Computers & Graphics*, v. 32, n. 4, p. 430–437, ago. 2008. ISSN 00978493. Disponível em: <http://linkinghub.elsevier.com/retrieve/pii/S0097849308000472>.

BOW, S. *Pattern recognition and image preprocessing*. CRC, 2002. Disponível em: <http://scholar.google.com/scholar?hl=en&btnG= Search&q=intitle:Pattern+Recognition+And+Image +Preprocessing#0>.

CAROLIS, B. D.; ROSIS, F. D.; CAROFIGLIO, V. Interactive Information Presentation by an Embodied Animated Agent. *Mind*, 1999.

DEMARA, R. F. et al. Towards Interactive Training with an Avatar-based Human-Computer Interface. *Education*, n. 8054, p. 1–10, 2008.

HERRERA, V. et al. Using an Emotional Intelligent Agent to Support Customers' Searches Interactively in e-Marketplaces. *2010 22nd IEEE International Conference on Tools with Artificial Intelligence*, Ieee, p. 15–22, out. 2010. Disponível em: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5671435>.

HUANG, X.; YIN, Y.; ZENG, G. Implementation of Speaking SoftMan in Game. *Neural Networks and Brain,*, p. 1403–1405, 2005. Disponível em: <http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1614893>.

KATHURIA, P. Hand Gesture Recognition. *Pattern Recognition*, 2011.

KOLLORZ, E.; PENNE, J.; HORNEGGER, J. Gesture recognition with a time-of-flight camera. *International Journal of Intelligent Systems Technologies and Applications*, x, n. 3, p. 1–8, 2008.

LAPTEV, I.; LINDEBERG, T. Tracking of Multi-state Hand Models Using Particle Filtering and a Hierarchy of Multi-scale Image Features. *Engineering*, p. 63–74, 2001.

Liu Yun, Z. P. An Automatic Hand Gesture Recognition System Based on Viola-Jones Method and SVMs. *2009 Second International Workshop on Computer Science and Engineering*, Ieee, p. 72–76, 2009. Disponível em: <http://doi.ieeecomputersociety.org/10.1109/WCSE.2009.769>.

LOMBARDO, V. et al. An Avatar-based Interface for the Italian Sign Language. In: *Complex, Intelligent and Software Intensive Systems (CISIS), 2011 International Conference on*. IEEE, 2011. p. 589–594. ISBN 9780769543734. Disponível em: <http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5989075>.

MONI, M. A.; ALI, A. B. M. S. HMM based hand gesture recognition: A review on techniques and approaches. In: . [s.n.], 2009. p. 433–437. Disponível em: <http://dx.doi.org/10.1109/ICCSIT.2009.5234536>.

NIXON, M.; AGUADO, A. S. *Feature Extraction & Image Processing, Second Edition*. 2. ed. Academic Press, 2008. Paperback. ISBN 0123725380. Disponível em: <http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20&path=ASIN/0123725380>.

OH, K.-g. et al. Real Time Lip Synchronization between Text to Speech(TTS) System and Robot Mouth. *19h IEEE International Symposium on Robot and Human Interactive Communication*, p. 620–625, 2010.

SNYDER, W.; QI, H. *Machine Vision*. Cambridge University Press, 2010. ISBN 9780521169813. Disponível em: <http://books.google.com/books?id=U15pcJMxRlwC>.

UCHINO, S. et al. VR Interaction in Real-Time between Avatar with Voice and Gesture Recognition System. In: *Advanced Information Networking and Applications Workshops, 2007, AINAW'07. 21st International Conference on*. IEEE, 2007. v. 2, p. 959–964. ISBN 0769528473. Disponível em: <http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4224230>.

VIOLA, P.; JONES, M. Robust real-time object detection. *International Journal of Computer Vision*, Citeseer, v. 57, n. 2, p. 137–154, 2002. Disponível em: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.156.4075&rep=rep1&type=pdf>.

WANG, Z.; XU, B. Construct A Naturalistic 3D Avatar With Live Help Interfaces Based On Multi-layered Representation. *Signal Processing*, p. 3509–3513, 2010.

WATERS, K.; LEVERGOOD, T. An automatic lip-synchronization algorithm for synthetic faces. *Proceedings of the second ACM international conference on Multimedia - MULTIMEDIA '94*, ACM Press, New York, New York, USA, p. 149–156, 1994. Disponível em: <http://portal.acm.org/citation.cfm?doid=192593.192644>.

XIAO, J.; STASKO, J.; CATRAMBONE, R. Embodied conversational agents as a UI paradigm: A framework for evaluation. *Embodied conversational agents-let's specify and evaluate them*, Citeseer, 2002. Disponível em: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.103.1086&rep=rep1&type=pdf>.