Interface Multimédia para Cadeira de Rodas Inteligente:

Relatório Final

Bruno Martins, Eduardo Valgôde

Universidade do Porto
Faculdade de Engenharia
FEUP

LIACC

Ciência.Inovação 2010 Programa Operacional Ciência e Inovação 2010
MINISTÉRIO DA CIÊNCIA, TECNOLOGIA E ENSINO SUPERIOR

Faculdade de Engenharia da Universidade do Porto
Departamento de Engenharia Electrotécnica e de Computadores
Rua Roberto Frias, s/n, 4200-465 Porto, Portugal

Agosto de 2006

Ciência.Inovação 2010 Programa Operacional Ciência e Inovação 2010

MINISTÉRIO DA CIÊNCIA, TECNOLOGIA E ENSINO SUPERIOR

Interface Multimédia para Cadeira de Rodas Inteligente:

Bruno Martins, Eduardo Valgôde

Orientador:
Professor Doutor Luís Paulo Reis

Co-Orientadores:
Engenheiro Pedro Miguel Faria

Faculdade de Engenharia da Universidade do Porto
Departamento de Engenharia Electrotécnica e de Computadores
Rua Roberto Frias, s/n, 4200-465 Porto, Portugal

Agosto de 2006

http://paginas.fe.up.pt/~ee99041/IMCRI/PT

http://paginas.fe.up.pt/~ee99041/IMCRI/EN

Resumo

Este trabalho tem como objectivo principal permitir a indivíduos, portadores ou não de deficiência, controlar uma cadeira de rodas com expressões faciais. Para tal são capturadas imagens do utilizador com recurso a uma câmara, a analise das imagens resulta em comandos que serão transmitidos à cadeira de rodas.

A aplicação, desenvolvida em Visual C++, utiliza técnicas de processamento de imagem de forma a obter a informação necessária da mesma. Entre estas técnicas incluem-se *Segmentação de cor*, *Filtros Gaussianos* e *Canny*.

Para processar a informação resultante do processamento de imagem é utilizada uma *rede neuronal*, cujas saídas permitem estabelecer o comando a ser transmitido a cadeira de rodas.

Neste relatório são apresentadas, de forma breve, todas as noções e técnicas que suportam o projecto, bem como as referências bibliográficas das quais os autores se socorreram.

Abstract

This work has as main target to allow individuals, with or without deficiency, to control a wheel-chair with facial expressions. In order to achieve that goal, user images (snapshots) are captured with a digital camera. From the analysis of those images results commands to be delivered to the wheel-chair.

The application, developed in Visual C++, uses image processing techniques in order to obtain the needed information from the snapshot. Among those techniques are *Colour Segmentation*, *Gaussian Filters* and *Canny*.

To process the information resulting from the image processing, a neuronal network is computed, whose outputs allow to establish the command to be inputted to the wheel-chair.

In this report are briefly presented all theories and techniques supporting the project, as well as bibliographic references that aided the authors.

Índice

Capítulo 1

1. Introdução

1.1 Enquadramento

Um dos recursos mais utilizados no auxílio de pessoas com deficiência motora, é a cadeira de rodas. O presente trabalho pretende estender a sua utilidade a um maior número de pessoas portadoras de deficiência motora, através de uma interface multimédia, que recorre à expressão facial para o seu comando.

Actualmente as cadeiras de rodas podem dividir-se em duas grandes categorias: manuais e eléctricas. As cadeiras de rodas movidas manualmente, dotadas em geral de grande robustez, destinam-se a ser movidas sobretudo pelo próprio utilizador, ou por outra pessoa. As primeiras têm como fonte de energia o próprio utilizador e o seu controlo é feito também por este. Isto requer alguma força de braços e mãos, uma boa destreza física e também uma boa capacidade de reconhecer o meio envolvente, quer a nível sensorial, quer a nível cognitivo. No segundo caso, em que as cadeiras são empurradas por outra pessoa, para além de requerer a disponibilidade de quem empurra, pode existir um impacto negativo no utilizador a nível psicológico devido à perda de privacidade e intensificação da sensação de dependência dos outros. Sobram portanto muitas pessoas com necessidades especiais a nível de locomoção, para as quais esta solução não serve (por exemplo: deficientes tetraplégicos, idosos com doenças degenerativas ou pessoas com paralisia cerebral).

As cadeiras de rodas eléctricas resolvem os problemas da necessidade de força física e de destreza manual, e deixa de ser preciso outra pessoa para a conduzir ou empurrar. Contudo, restam alguns problemas ainda, muitos deles já com boas soluções, nomeadamente nas áreas de: movimentação em zonas acidentadas ou de piso irregular, maneabilidade, autonomia, segurança, método de comando, etc. O presente trabalho procura resolver o problema de comando de cadeiras de rodas para as pessoas que não podem usar o convencional "joystick". Dentro deste campo algumas soluções já encontradas envolvem: a utilização dos movimentos da cabeça como se fosse um "joystick", a utilização do pen-

samento através de eléctrodos na cabeça, reconhecimento de voz, expressão facial, etc.. A solução implementada e aqui relatada recorre à expressão facial como método de comando.

Existem cerca de 60 movimentos individuais e distintos na cara e cabeça, visualmente perceptíveis, sendo que alguns destes podem ocorrer em simultâneo. Isto significa que existem muitas possibilidades diferentes na criação de uma linguagem de comando e que esta pode ser facilmente adaptável às limitações de cada um.

1.2 Motivação

Procura-se com este trabalho contribuir para a melhoraria da qualidade de vida daqueles que dependem da tecnologia para se deslocar de forma autónoma. Existem já alguns meios de apoio à movimentação muito bons, para o actual estado actual da tecnologia. No entanto, existe sempre a tendência, nas empresas que os produzem, de tentar alcançar o maior número de pessoas focando-se apenas no problema mais geral. O resultado é que sobram ainda muitas pessoas, que não podem utilizar essas soluções, devido a outras condições adicionais que limitam a sua capacidade de operação e controlo. Refere-se em concreto as cadeiras de rodas eléctricas, cujo método mais comum de comando é um joystick manual. Qualquer pessoa que não possa utilizar as mãos não vai ser capaz de operar este tipo de cadeira e este problema fica normalmente em segundo plano.

O trabalho executado caminha no sentido de adaptar os meios já existentes a essas relativas minorias. Resolver esta situação muitas vezes requer a criação de dispositivos especiais, que devido à tecnologia utilizada ou à falta de um mercado suficientemente grande, faz com que estes sejam demasiado caros para a maioria das pessoas. A maneira escolhida de contornar o problema é aproveitar os desenvolvimentos tecnológicos recentes no campo da informática e multimédia, assim como a sua popularidade, baixo preço e fácil acesso.

Em concreto, pretende-se tornar possível utilizar uma cadeira de rodas, de um modo prático e eficiente, recorrendo à expressão facial como método de comando e sem recorrer à utilização das mãos.

1.3 Objectivos

O primeiro objectivo deste trabalho é projectar e implementar uma interface de comando de uma cadeira de rodas eléctrica. O comando é feito através da expressão facial e o modo de adquirir a informação da expressão facial é através de uma câmara digital. O segundo objectivo será implementar um simulador gráfico de uma cadeira de rodas a ser controlada com esta interface de modo a poder validar o sistema implementado. Finalmente, pretende-se integrar o sistema desenvolvido numa cadeira de rodas real.

Objectivos:

- reconhecer expressões faciais relevantes para o comando de uma cadeira de rodas de forma robusta (capaz de reconhecer a expressão em qualquer tipo de cara, para várias condições de iluminação em interiores);

- estabelecer uma linguagem apropriada para o comando de uma cadeira de rodas;

- criar um modelo de uma cadeira de rodas e um espaço físico onde esta se deslocará;

- aplicação multimédia com interface baseada em expressões faciais que permita ao utilizador controlar a cadeira de rodas através da selecção das expressões correspondentes a cada comando de controle.

1.4 Estrutura do Relatório

O relatório está dividido em nove capítulos. No capitulo 2 é feita a apresentação do problema, nomeadamente explicando o sistema FACS, considerações sobre expressões faciais e restrições de diversos tipos resultantes do problema em causa e solução adoptada. No seguinte capitulo (3) é feita uma visita guiada pelos trabalhos já desenvolvidos nesta área, com referência as diferentes técnicas, métodos e abordagens utilizadas e respectivas referências bibliográficas. O capitulo 4 descreve o trabalho desenvolvido pelos autores, referindo a abordagem feita, as diferentes fases do processo de extracção de informação de uma imagem e a forma como a partir dessas informações é calculada a expressão facial que corresponde a imagem. No capitulo 5 são referidos todos os pormenores relativos à implementação do sistema, nomeadamente ao nível da interacção com o utilizador, quer no que toca a interface bem como à linguagem de comando, sendo também descrita a

implementação de um simulador básico que permite a verificação dos comandos. O sexto capitulo relata as conclusões obtidas e descreve os problemas encontrados durante o projecto, assim como refere as alterações feitas ao projecto inicial. No sétimo capitulo deste relatório, é indicado o caminho a seguir, isto é, como este projecto poderia evoluir, que funcionalidades, que métodos e que detalhes seriam desenvolvidos de seguida de forma a dar continuidade ao projecto e a torna-lo mais funcional e atractivo tanto a nível cientifico como pragmático. O capitulo 8 refere as referências bibliográficas mais importantes, que foram consultadas com mais frequência para a realização deste trabalho. No nono e ultimo capitulo são apresentados anexos que tratam de teoria de suporte ao projecto, bem como de esquemas e dados que possibilitam um entendimento melhor e mais aprofundado do trabalho.

Capítulo 2

2. Descrição do Problema

2.1 A Expressão Facial

A expressão facial é uma das primeiras linguagens que o ser humano utiliza na sua vida. Ao contrário de outras como a língua falada, ou da escrita, a expressão facial como meio de exprimir emoções é inata na espécie humana, ou seja, é independente do contexto cultural e independente de aprendizagem também. Isto é facilmente observável se se tiver em conta que pessoas cegas à nascença têm expressões faciais de alegria, tristeza, surpresa e de outras emoções básicas, iguais às de uma pessoa provida de visão.

A utilização da expressão facial é crucial desde os primeiros dias de vida de um ser humano. Enquanto ainda bebé, usa-a para dizer aos seus pais o que está a sentir, nomeadamente desconforto ou alegria, e lê também as faces das pessoas que o rodeiam.

Devido à sua forte relação com as emoções e à sua natureza instintiva, é frequente as pessoas fazerem algumas expressões faciais inconscientemente, que traduz o seu estado emocional no momento. Algumas destas expressões são muito breves e de pouca amplitude – microexpressões - pelo que é preciso estar atento para conseguir observá-las. Contudo, também é possível controlar razoavelmente bem, conforme a experiência da pessoa, a face independentemente do estado emocional. Por exemplo: na antiga cultura japonesa a mulher de um samurai quando recebia a notícia de que o marido tinha morrido em batalha, sorria em vez de chorar, de modo a não mostrar fraqueza. Outro exemplo, mais comum, é o trabalho de um actor. Uma técnica muito utilizada quando se quer representar um determinado estado emocional, é reviver na imaginação um determinado momento da vida em que se sentiu essa emoção. Isto faz com que a expressão representada seja mais perfeita do que se resultasse de um mero processo racional. Existem também muitas outras expressões que não têm nenhuma emoção associada (piscar o olho, levantar uma sobrancelha, etc.) e que para as fazer apenas interessa a concentração nos movimentos que a definem.

Outro aspecto importante sobre a expressão facial, é que não só pode ser o resultado de um estado emocional, como também pode induzir no próprio, de forma limitada, a emoção que traduz. Este efeito não fica por aqui, estende-se também ao observador. Quem olha para outra pessoa faz uma avaliação inconsciente e praticamente imediata, do estado emocional e físico da pessoa. A expressão facial é portanto um meio privilegiado de obter esta informação, e pode induzir também no observador emoções em resposta às observadas. Por exemplo se alguém encontra um amigo e este tem uma expressão triste, o primeiro também tem tendência a ficar triste ou preocupado, por oposição, se estiver a sorrir então provavelmente este também fica feliz.

Falta ainda lembrar que em muitas pessoas a sua expressão relaxada (cara normal), que supostamente não deve transmitir nenhuma emoção, parece estar a fazer uma determinada expressão, mais frequentemente: em sorriso permanente, ou em "cara zangada" permanente. Nestes casos só com o contacto habitual é que é possível começar a distinguir bem se de facto está a fazer uma expressão facial ou não.

2.2 Sistema FACS

O sistema FACS – *Facial Action Coding System* (sistema de codificação da acção facial) – foi desenvolvido por Ekman e Friesen em 1978, com o intuito de compreender como funcionam os movimentos da cara. Neste sistema são descritos todos os movimentos da cara visualmente perceptíveis e activados individualmente pelo menor número de músculos. O sistema está estruturado em unidades de acção (action units), que indicam que músculos foram contraídos para fazer uma determinada expressão. Isto permite distinguir movimentos de características fisionómicas do indivíduo. Foram determinadas 64 unidades de acção e qualquer expressão facial pode ser descrita como uma combinação destas unidades básicas. Este sistema permite também medir a intensidade de uma expressão facial e o seu "timing". Uma grande vantagem deste sistema é a sua objectividade. Apenas os movimentos são descritos e não é feita qualquer classificação relativa ao significado da expressão, ao contrário do que acontece em outros sistemas como o *Facial Affect Scoring Technique.*

2.3 Restrições

2.3.1 Expressões Faciais Detectáveis e Expressões Faciais Utilizáveis

Para este trabalho, o método de descrição ou decomposição de uma expressão facial, é uma versão baseada num resumo do sistema FACS. Esta escolha justifica-se pela necessidade, de um sistema mais simples devido ao nível de precisão que é previsto obter, numa fase inicial, e também à informação necessária para que os objectivos sejam atingidos.

Aproveitando o sistema FACS são descritas a seguir as restrições na escolha de movimentos da face utilizáveis e detectáveis.

Acções não utilizáveis: existe uma série de movimentos faciais, como levantar as sobrancelhas, franzir ligeiramente, ou sorrir que são feitas em circunstâncias muito diversas e que não devem ser utilizadas na linguagem de comando da cadeira de rodas;

Acções que só interessa a sua combinação: acções como por exemplo enrugar o nariz sem franzir o sobrolho são bastante difíceis de executar. Não se pretende que o utilizador tenha de ser particularmente dotado no controlo da expressão facial pelo que cero tipo de movimentos apenas a sua combinação interessa;

Acções difíceis de distinguir: Em geral todas as acções são tão difíceis de distinguir quanto menor for a sua intensidade e alguns movimentos mesmo na sua máxima intensidade são difíceis de distinguir. Por exemplo distinguir entre levantar as sobrancelhas e arquear as sobrancelhas;

Acções socialmente embaraçosas: qualquer combinação possível que envolva pôr a boca "em forma de beijo" é potencialmente embaraçosa para o utilizador da cadeira de rodas em público.

Medida da intensidade de uma expressão: para evitar erros ou falsos comandos, o sistema vai detectar apenas expressões com a intensidade máxima (os músculos contraídos ao máximo).

Cada expressão deverá durar mais de meio segundo de modo a confirmar a intenção de fazer essa expressão, assim como a evitar que uma eventual micro-expressão induza um comando não intencional.

Resumindo, interessa apenas identificar movimentos fáceis de fazer e fáceis de detectar, apenas com a intensidade máxima, de duração superior a meio segundo e que provoquem o mínimo de mal entendidos a nível social.

2.3.2 Limitações do Utilizador

Existem várias condições que limitam a capacidade de mexer a cara de forma controlada. Estas podem ser essencialmente: paralisia parcial ou total dos músculos da cara, motivada sobretudo por doenças como a paralisia de bell, a trombose, o tumor cerebral, ou a doença de parkinson; ou movimentos descontrolados da cara, que se devem sobretudo a doenças neurológicas como a esclerose múltipla, paralisia cerebral e coreia.

2.3.3 Limitações de Processamento e Temporais

Se tivermos em linha de conta que o objectivo do projecto é criar um interface que permita controlar em tempo real uma cadeira de rodas, facilmente se chega à conclusão da grande importância do tempo de processamento da imagem.

Foi estabelecido que para haver viabilidade numa perspectiva de utilização real do sistema, seriam processadas duas *frames* por segundo (i.e. a cadeira de rodas recebe comandos de meio em meio segundo). Este valor é um limite inferior, já que abaixo de duas imagens por segundo torna-se impossível controlar a cadeira num ambiente real.

Deste facto resulta que o tempo entre a aquisição da imagem (*frame*) até a obtenção do comando para a cadeira será, no limite, 500 ms. Durante este intervalo de tempo tem de ser processada a imagem, de forma a obter características que serão entradas da rede neuronal, cujas saídas darão o comando para a cadeira. A utilização de uma rede neuronal permite-nos considerar desprezável (negligenciável) o tempo de processamento das características para obtenção de comandos (uma vez treinada a rede).

Esta limitação temporal impede a utilização de métodos avançados de processamento de imagem, mais ainda tendo em conta que a aplicação deverá ser executada num "Lap Top". A imagem é organizada por *pixels*, tipicamente qualquer operação de processamento de imagem tem de percorrer todos os *pixels* pelo menos uma vez, a diferença de tempos de execução entre métodos de processamento prende-se com a necessidade de executar operações mais demoradas e de passarem pelo *array* de *pixels* diversas vezes.

A titulo de exemplo, foi verificado em testes preliminares num "Lap Top" com processador *Pentium centrino* a 1,80 GHz e 512 Mb de RAM, que operação de segmentar uma *frame* (i.e. fazer pertencer cada *pixel* a uma categoria pré-definida com a utilização de uma tabela de *Look Up*, como veremos mais adiante) demora cerca de 25 ms, ao passo que a utilização do filtro mediana para suavizar a imagem poderia demorar cerca de 500 ms, ou seja vinte vezes mais. Esta diferença deriva do facto da segmentação de cor passar uma vez por cada *pixel* e para cada um deles executar uma simples operação de procura numa tabela indexada (operação rápida). Já o filtro mediana passa só uma vez por cada *pixel*, (no teste feito), mas por cada *pixel* tem de construir um *array* com o próprio *pixel* e os vizinhos (vizinhança de 8, 24, ...), de seguida ordena-lo para calcular a mediana e por fim atribuir ao *pixel* pivot (i.e. central) o valor da mediana.

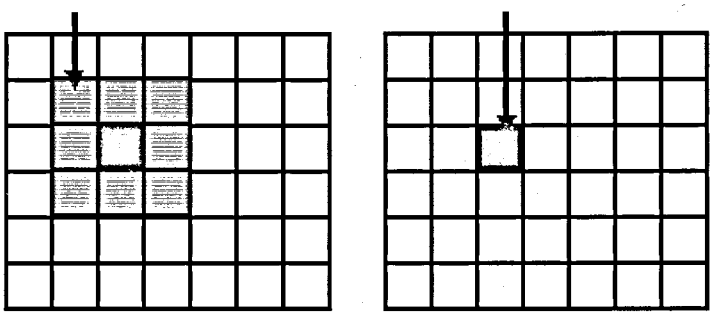


Figura 1- Vizinhança de um pixel

Com este teste verificamos o que seria de prever teoricamente, ou seja, para este projecto as limitações temporais impedem a utilização de processamento de imagem demorado o que elimina a possibilidade de utilizar métodos avançados.

No que toca ao tempo consumido com obtenção de características da imagem, podemos dizer que é diminuto quando comparado com o tempo total, já que tipicamente é feito com uma única passagem pela zona da imagem da qual queremos obter características, a partir do processamento de imagem já efectuado e simples operações matemáticas

2.3.4 Meio Físico

O sistema vai operar em cima de uma cadeira de rodas, e, por enquanto, apenas dentro de portas.

As condições de iluminação deverão ser as condições habituais de uma sala bem iluminada, com a ressalva de que a luz não devera ser de cor acentuada e sim, o mais perto possível da luz branca.

Não foi estudado o efeito da vibração no sistema, mas esta deve ser evitada de modo a garantir as melhores condições na captura da imagem. Deve-se, portanto, evitar travagens bruscas ou ambientes com níveis de ruído na banda dos graves e infra sons, muito elevado. Por exemplo: dentro de comboios, perto de estações de comboio ou zonas onde hajam obras com grande nível de vibração no solo. O suporte da câmara deve também estar colocado de maneira a amortecer ao máximo as vibrações mecânicas inerentes ao funcionamento normal da cadeira de rodas.

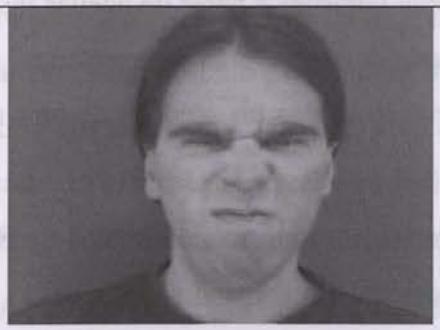
A cara do utilizador deverá ter atrás um fundo verde que cubra a quase totalidade do campo de visão da câmara. Este fundo destina-se a tornar a localização da cara mais fácil a evitar tomar outros objectos como fazendo parte da cara. Nesta fase o sistema ainda não é compatível com todos os tipos de cara. A cara não deve ter uma tez muito escura, e o cabelo deve ser escuro ou branco. O sistema não é capaz de reconhecer a cara correctamente se a pessoa tiver o cabelo de cor muito parecida com a da pele. O utilizador também não deve estar a utilizar óculos nem deve usar barba. Prevê-se que estas restrições possam ser eliminadas com o melhoramento do software utilizado.

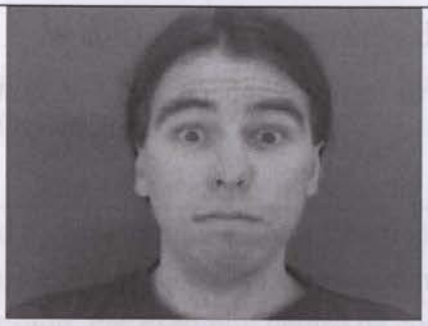
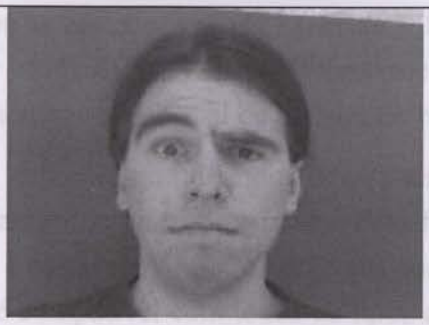
2.4 Linguagem de Comando

Uma vez identificada a expressão feita pelo utilizador, é necessário identificar que comando corresponde à mesma.

A linguagem de comando deverá poder ser editada de forma a se adaptar a diferentes utilizadores assim como a diferentes insuficiências que possam afectar os mesmos. De notar que *à priori* não são sabidas quais as expressões que são melhor detectadas, pelo que, uma vez testado o sistema deverá ser dada prioridade à utilização das expressões melhor identificadas, na medida do possível.

A linguagem base, estabelecida para efeitos de teste e comparação é a seguinte:

	VIRAR ESQUERDA
	VIRAR DIREITA
	ANDAR PARA FRENTE
	ANDAR PARA TRÁS

	**PARAR**
	**COMEÇA/PARA DE ACEITAR INSTRUÇÕES**

Linguagem de teste

A necessidade de um comando para o sistema começar e parar de aceitar instruções surge, como facilmente se entende, da possibilidade de serem "mal interpretadas" expressões feitas quer involuntariamente, quer voluntariamente mas no decorrer de uma conversa.

Caso a cadeira esteja equipada com sensores e controlo automático inteligente, podem ser estabelecidas expressões para serem associadas a comandos do tipo:

- Seguir parede à direita
- Passar pela porta em frente
- Seguir caminho pré-definido para casa
- Seguir elemento
- ...

Capítulo 3

3. Descrição geral dos trabalhos desenvolvidos na área

A grande maioria dos trabalhos já realizados nesta área (processamento facial) está ligada à área de reconhecimento facial (para identificação) ou reconhecimento de expressões para modelação facial.

A área na qual foram desenvolvidos mais trabalhos envolvendo reconhecimento de expressões faciais foi a área de interacção entre humano e máquina (computador/robot) e ferramentas para estudo psicológico. A principal diferença entre estes trabalhos e a grande maioria dos desenvolvidos nesta área de processamento facial prende-se com o facto de se precisar de identificar expressões faciais independentemente da face ao invés de identificar faces (indivíduos) independentemente das expressões faciais.

A partir deste ponto falaremos apenas dos trabalhos na área da identificação de expressões faciais.

O FACS, "*Facial Action Coding System*", sendo um sistema que lista todas as expressões voluntárias possíveis de fazer, serve de base a muitos dos trabalhos realizados [2, 3, 19, 20]. O primeiro passo é, na maioria dos casos, eliminar o fundo ou/e detectar a face numa imagem. Há duas aproximações a este problema, reconhecer a face como um todo com utilização de modelos [1, 6, 7, 20], ou detectando detalhes (olhos, boca, ...) e a partir dai o resto da face ou ficar apenas com a informação dos detalhes relevantes [1, 2, 4, 5, 15, 19]. De seguida há autores que optam por construir modelos da face [3, 12]. Uma vez tendo informações da imagem há uma grande variedade de métodos que podem ser usados, uma das aproximações bastante usadas é a de construir uma base de dados ou *eigenspace* onde possam ser guardadas imagens ou dados – tipo de cada expressão a detectar e de

seguida detectar as expressões por comparação [1, 7, 17]. Outra abordagem é, processar a informação recolhida com uma rede (neuronal, Bayesiana, ...) e depois de treinada a rede, a saída identificará a expressão efectuada [3, 4, 13, 15, 18]. Uma outra abordagem, utiliza classificadores locais ou globais (K *nearest neighbour, kernel, Gabor,* ...), [4, 5, 6, 7]. É também comum ver sistemas baseados, ou que utilizem durante o processamento, comparação e subtracção de imagens, sendo que as diferenças identificam expressões, ou eliminam informação não diferenciadora de expressão [2, 4, 7]. Por fim, há também trabalhos que utilizam modelos escondidos de Markov (*Hidden Markov Models, Pseudo 3D Hidden Markov Models*) que são maquinas de estado finitas, não determinísticas para a partir da informação da imagem identificar a expressão [10, 12]. Com os estudos na área da psicologia começaram a ser criados sistemas que têm em conta os movimentos musculares. Uma das aproximações nesta área é a partir de modelos faciais fazer representações dos tempos de activação dos músculos (*Motion energy*) que é característica de cada expressão [19].

Neste ponto do relatório, pretende-se apenas dar uma ideia de algumas das técnicas utilizadas em alguns dos trabalhos desenvolvidos até á data na área de reconhecimento de expressões faciais. Não se pretende explicar de forma pormenorizada nenhum dos métodos ou trabalhos, para tal ficam as referências bibliográficas.

Capítulo 4

4. Interface Multimédia

4.1 Abordagem do Problema

Método

A metodologia de desenvolvimento adoptada, foi do tipo iterativo. A partir dos objectivos gerais para a interface, foi definido um conjunto de requisitos mínimos necessários para que o sistema funcionasse. O primeiro objectivo é obter um protótipo, que cumpra esses requisitos básicos, e a partir daí ir redefinindo e aumentando os requisitos de modo a tornar o sistema mais prático, robusto e com funcionalidades acrescidas. Como ilustração de como se está a usar este processo na prática: o sistema está previsto funcionar com um fundo verde atrás da cara, de modo a tornar a implementação mais simples e fácil. Só num estado de desenvolvimento mais avançado é que se vai considerar remover esta restrição.

Esta metodologia não só é seguida para o sistema no seu todo, mas também para cada bloco lógico que compõe o sistema (com a diferença de que um bloco poderá ser eliminado e substituído por outro). Por exemplo a classificação de cores começou por ser feita e testada apenas para uma dada condição de iluminação e só quando ficou a funcionar bem, se passou à sua adaptação dinâmica para várias condições de iluminação.

Tendo em conta que cada protótipo se resume sobretudo a software, não há grande problema a nível de custos de materiais cada vez que se constrói um. É possível que este método não seja o mais eficiente a nível de tempo no caso em que os executantes são experientes e têm a certeza do que são as boas soluções de como atingi-las. Nesse caso uma abordagem mais directa do tipo formal poderia ser melhor. Não é o caso aqui, pelo que parece mais inteligente seguir esta abordagem com menor risco de falha global do sistema.

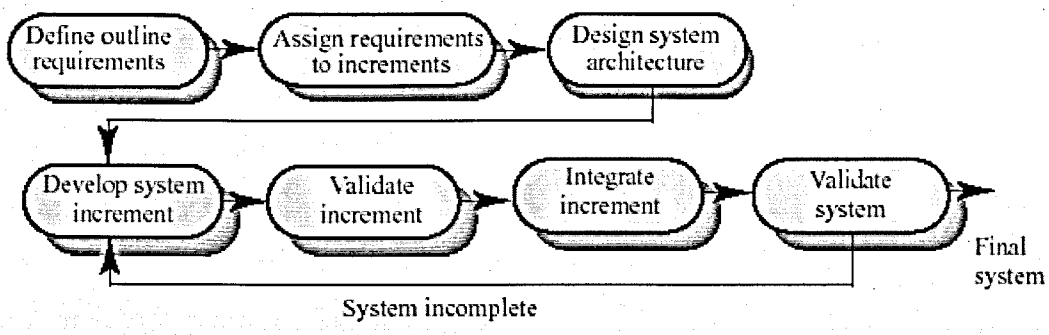


Figura 2 Método

Descrição geral

O sistema desenvolvido tem a arquitectura lógica exposta no diagrama de blocos e está organizado em 4 áreas: aquisição de informação, pré-processamento, identificação e interface.

A aquisição de informação trata de converter a informação do mundo físico, em concreto a informação visual da cara do utilizador, numa matriz de pontos RGB, de modo a ser possível tratar essa informação no computador.

A área de pré-processamento, dedica-se a extrair a informação relevante para identificar uma expressão facial (encontrar a cor de pele, os contornos da boca, etc.) e a organizar essa informação em estruturas de dados de forma compacta.

Na fase de identificação, a informação obtida do pré-processamento é processada de forma a deduzir qual a expressão facial expressa na imagem.

Finalmente, na parte da interface está definida a linguagem de comando da cadeira de rodas e é onde será implementado futuramente o porto de saída para interagir com o sistema de controlo da cadeira. É também fornecida ao utilizador a informação do que o sistema está a fazer, assim como a possibilidade de configurar o sistema (alterar a linguagem de comando, treinar a rede neuronal).

Aquisição de informação

Obtém-se a informação visual da cara do utilizador através de uma câmara digital, orientável. A câmara é dotada de uma matriz de sensores CCD, com alguns ajustes automáticos embebidos (tempo de exposição e "white balance" por exemplo).

Pré-processamento

Esta parte do sistema começa por filtrar o ruído da imagem que é introduzido no seu processo de aquisição. Segue-se a classificação das cores da imagem, que permite identificar conjuntos gerais como a cara ou o cabelo. Ainda a partir da imagem filtrada são extraídas as orlas presentes na imagem. Esta informação permite saber pormenores de cada um dos conjuntos definidos na classificação de cores. Segue-se a síntese e cruzamento dos dados obtidos nestes dois processos de forma a obter uma função discreta e unidimensional da imagem.

A informação da posição da cara na imagem poderá ser também aproveitada para ajustar a orientação da câmara.

Identificação

A fase de identificação é implementada por uma rede neuronal, que por natureza é capaz de fazer criar automaticamente as regras que permitem identificar um padrão. Este tipo de processamento é particularmente útil quando a quantidade de informação é demasiado grande para que a lógica que define a relação entre as entradas e as saídas, seja inferida ou deduzida individualmente. Para poder identificar correctamente é necessário uma fase inicial de treino, em que a partir de algumas expressões faciais a rede generaliza as regras que definem cada uma. Depois desta fase de treino o funcionamento limitar-se-á a receber a informação do pré-processamento e a deduzir qual a expressão facial.

Interface :

Aqui encontra-se implementada a função de transferência entre expressões faciais e os comandos correspondentes. Os comandos destinam-se a numa fase posterior serem convertidos em sinais eléctricos para o sistema de controlo da cadeira de rodas. Neste bloco inseriu-se a parte de interface com o utilizador. O utilizador tem a informação do que o sistema está a fazer: janela a mostrar as imagens capturadas, informação do que está a identificar e a linguagem que de comando que está a ser utilizada. Aqui será implementa-

do também algumas hipóteses de controlo do sistema de comando: iniciar, parar, mudar a linguagem de comando e calibrar a rede neuronal.

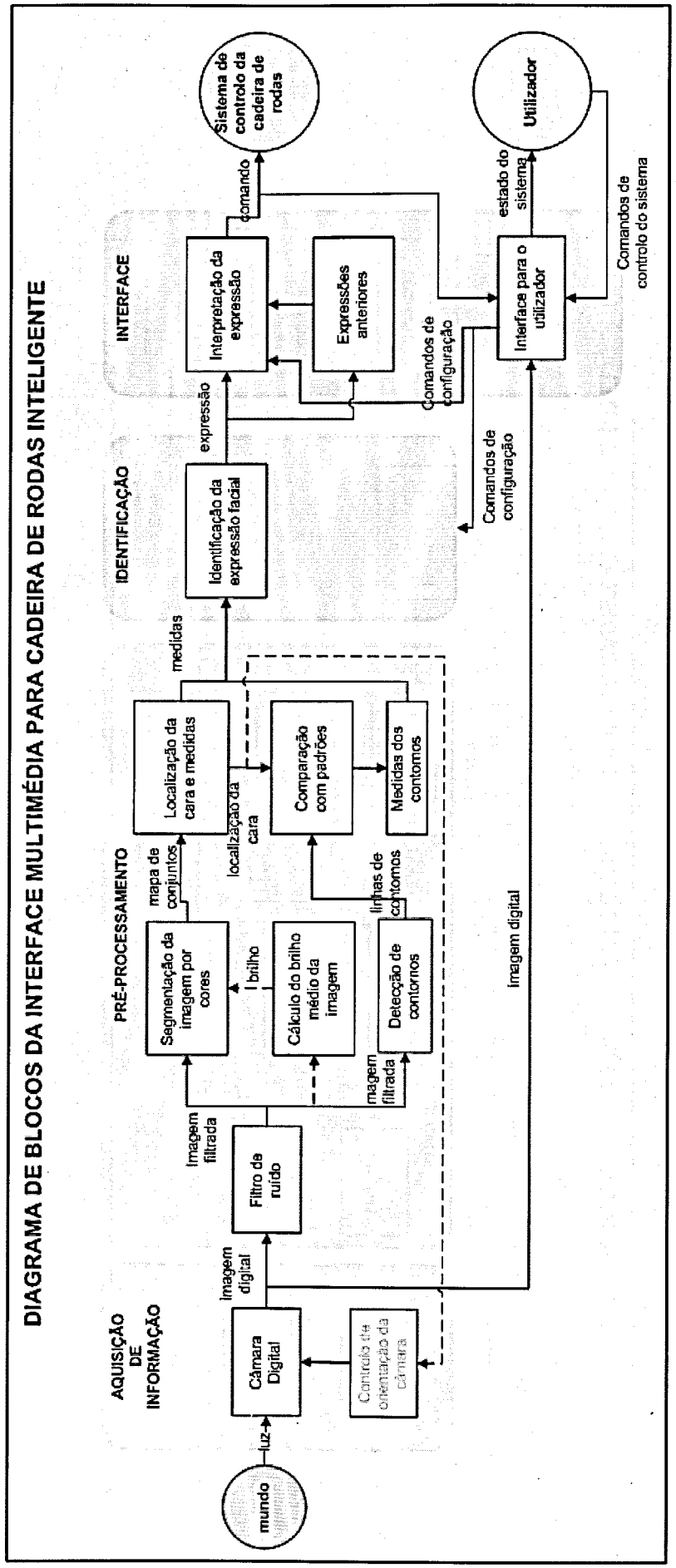
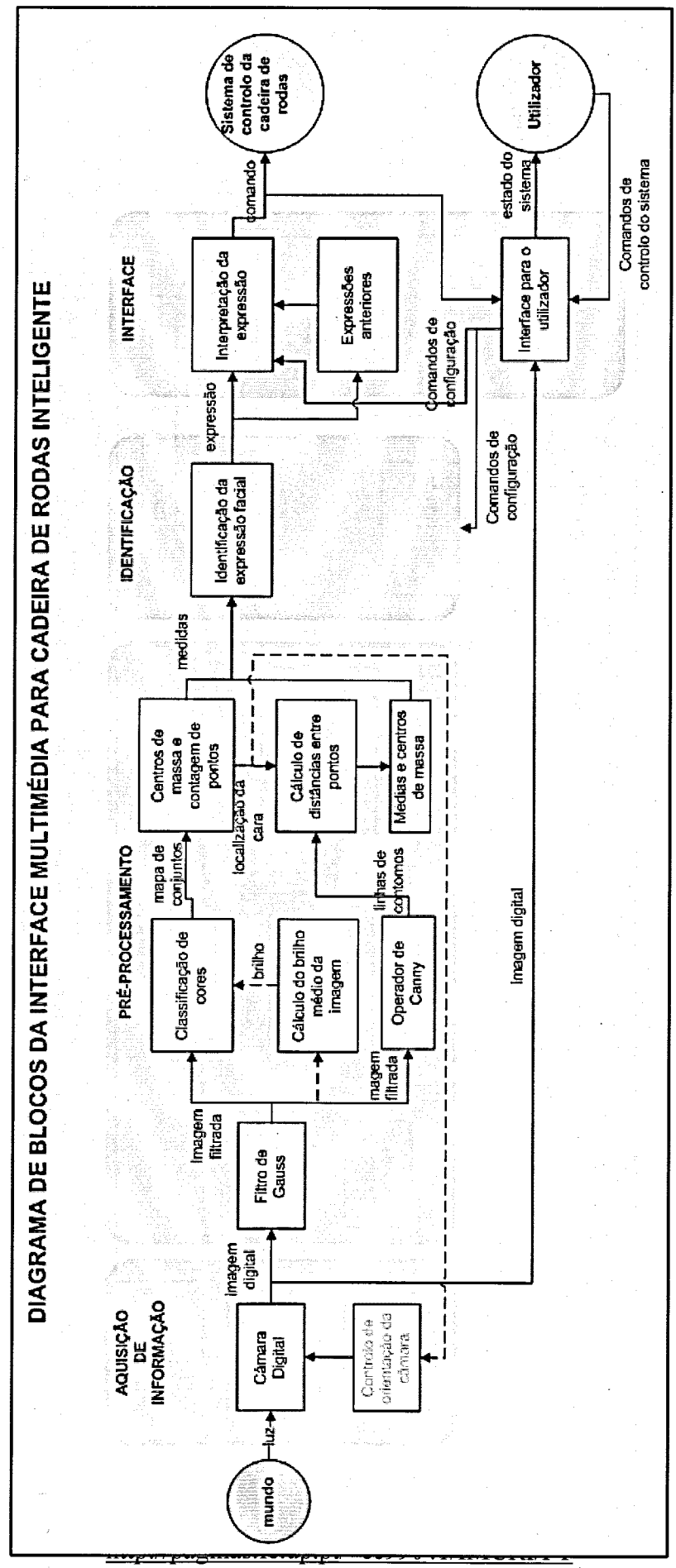


DIAGRAMA DE BLOCOS DA INTERFACE MULTIMÉDIA PARA CADEIRA DE RODAS INTELIGENTE
AQUISIÇÃO DE INFORMAÇÃO
PRÉ-PROCESSAMENTO
IDENTIFICAÇÃO
INTERFACE
mundo
luz
Câmara Digital
Controlo de orientação da câmara
Imagem digital
Filtro de ruído
Imagem filtrada
Segmentação da imagem por cores
mapa de conjuntos
I brilho
Cálculo do brilho médio da imagem
Imagem filtrada
Detecção de contornos
linhas de contornos
Localização da cara e medidas
localização da cara
medidas
Comparação com padrões
Medidas dos contornos
Identificação da expressão facial
expressão
Interpretação da expressão
Expressões anteriores
comando
Sistema de controlo da cadeira de rodas
Comandos de configuração
Comandos de configuração
Interface para o utilizador
estado do sistema
Utilizador
Comandos de controlo do sistema
Imagem digital



http://paginas.fe.up.pt/~ee99041/IMCRI/EN

4.2 Pré-Processamento

4.2.1 Filtro de Ruído

O processo de aquisição da imagem está sujeito a ruídos de vários tipos. O ruído não pode ser previsto devido à sua natureza aleatória, nem medido com grande precisão dado que não é possível obter uma imagem perfeita sem ruído como termo de comparação. O ruído é um factor sempre presente numa imagem digital e apenas se pode minimizar os seus efeitos, não é possível nem necessário eliminá-lo por completo.

O ruído que afecta a imagem e que pode ser prejudicial aos algoritmos de pré-processamento é introduzido ao nível do sensor fotoeléctrico. O ruído introduzido na transmissão da imagem da câmara para o computador, via USB não tem influência na qualidade da imagem recebida no PC. Das várias fontes de ruído que podem afectar um sensor CCD, foram consideradas como importantes para esta aplicação as seguintes:

- **ruído térmico**: a contribuição para a energia acumulada em cada pixel não provém só dos fotões incidentes. Os electrões do sensor também podem ser excitados pela temperatura, o que introduz um erro na imagem conhecido como ruído térmico. Uma maneira de medir o ruído térmico é capturar uma imagem sem qualquer excitação luminosa com o mesmo tempo de exposição utilizado para obter as imagens pretendidas. A imagem resultante é composta maioritariamente por ruído térmico. Para compensar este ruído pode-se usar a técnica de subtrair esta imagem às imagens que interessam. Salienta-se que este processo não vai eliminar o ruído térmico, dada a sua natureza aleatória, mas apenas compensá-lo. Por enquanto, com base na observação visual dos resultados, não é necessário usar esta técnica. Contudo, poderá vir a ser útil implementá-la quando se partir para ambientes com fraca iluminação. A redução da intensidade luminosa implica aumentar o tempo de exposição da câmara (feito automaticamente na webcam), o que aumenta também significativamente o efeito do ruído térmico;

- **variações de sensibilidade de cada pixel**: o chip CCD tem algumas imperfeições e como resultado a sensibilidade à luz de cada pixel não é igual. Para observar a contribuição deste tipo de ruído, pode-se capturar uma imagem de uma tela ou cenário com intensidade de luz muito uniforme, de preferência com um tempo curto de exposição de modo a minimizar a influência do ruído térmico, e observar as variações em cada pixel da imagem resultante. É também possível compensar este problema em específico.

- variações no número de fotões por unidade de tempo que atingem um determinado pixel: os fotões não apresentam uma distribuição espacial sempre uniforme e por isso alguns pixels precisariam de maior ou menor tempo de exposição para ter um valor 100% correcto;

- ruído aleatório ("salt and pepper noise"): este tipo de ruído é observado na imagem como pixeis cuja cor não tem qualquer relação com a dos pixels envolventes (mais facilmente visíveis quando são brancos ou pretos) e em geral acontece em poucos pixels por imagem.

Os tipos de ruído mencionados afectam a imagem tanto em termos de brilho como em termos de cor e a sua influência pode variar de câmara para câmara no que toca à periodicidade espacial e amplitude. O ruído para este tipo de sensor pode em geral segue o modelo:

$\eta(x,y) = \sqrt{g(x,y)} \cdot \eta_1(x,y) + \eta_2(x,y)$ para cada canal R G e B

η_1, η_2 ruído branco com média nula

$g(x,y) = \alpha_{x,y} \cdot w^{\beta_{x,y}}$ - resposta de cada pixel do sensor CCD, com □e□constantes dependentes da construção de cada sensor em particular;

$\sqrt{g(x,y)} \cdot \eta_1(x,y)$ - ruído associado aos fotões, depende do sinal recebido

$\eta_2(x,y)$ - ruído térmico independente do sinal recebido, o desvio padrão, que é função da temperatura a que está o CCD dá uma ideia da amplitude média do ruído. É possível estimar esta componente se se eliminar o sinal de entrada.

Em relação ao hardware utilizado é observável sobretudo uma presença considerável de ruído térmico, sobretudo em condições de fraca iluminação como seria de esperar. O padrão de ruído observado apresenta alta frequência espacial, um desvio pouco considerável em termos de cor e amplitude relativamente baixa em termos de brilho também. Concluiu-se que a relação sinal ruído devia ser melhorada antes de se proceder à identificação e extracção de características da imagem.

Através das imagens capturadas pretende-se retirar características que ocupam uma percentagem muito razoável da imagem e para as quais o erro de alguns pixels não deverá ser muito significativo, ao contrário de outro tipo de aplicações como por exemplo as imagens usadas em astronomia, ou quando se quer encontrar padrões de pequeno tamanho. Deste modo e com vista em tornar o decorrer dos trabalhos mais rápido o método de ava-

liar a qualidade da imagem antes e após a redução de ruído baseou-se na simples observação da imagem.

O método de redução de ruído da imagem foi o filtro passa baixo. Dado que as componentes de ruído consideradas mais importantes (e descritas acima) apresentam sobretudo uma elevada frequência espacial, a remoção das altas frequências retirará grande parte do ruído da imagem. Este processo também vai retirar parte do sinal que interessa também, sobretudo a nível de contornos. Contudo, tendo em conta o algoritmo que se segue de detecção de contornos, este resultado até vai ser útil.

Função do filtro:

O sistema implementado no domínio das frequências tem a forma:

$$I_{filtrada}(u,v) = F(u,v) \times I(u,v)$$

em que u e v são as frequências espaciais do brilho de cada uma das componentes R,G e B segundo a direcção horizontal e vertical respectivamente. I é a transformada de Fourier discreta da imagem original F a função de transferência do filtro passa baixo e I filtrada a imagem filtrada.

O filtro escolhido foi um filtro gaussiano com a seguinte função de transferência:

$H(u,v) = e^{\frac{-D^2(u,v)}{2\sigma^2}} = e^{\frac{-D^2(u,v)}{2D_0^2}}$ em que D0 é a frequência de corte

$D(u,v) = \sqrt{\left(u - \frac{M}{2}\right)^2 + \left(v - \frac{N}{2}\right)^2}$ assumindo que a imagem foi centrada (M é a altura da imagem e N a largura)

Uma das propriedades dos filtros gaussianos é que a sua transformada inversa é também uma função de gauss. Quer no domínio espacial quer no domínio das frequências, a frequência de corte é dada pelo desvio padrão. No entanto, no domínio espacial esta diminui com o aumento do desvio padrão ao contrário do que acontece no domínio das frequências. Esta característica e o facto da forma da função e os seus efeitos na imagem serem simples e fáceis de prever, torna a utilização deste tipo de filtro bastante intuitiva.

Optou-se por implementar o filtro de gauss no domínio espacial através de uma janela de convolução, dado que esta forma é mais simples e rápida de executar. Verificou-se que com um σ de 1 para as direcções de x e y se obtiveram bons resultados.

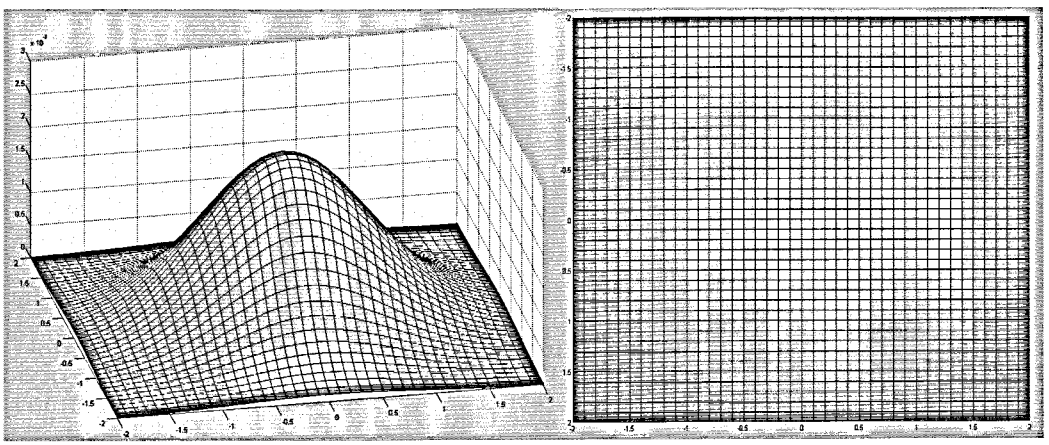


Figura 3 – Função de Gauss

$i_{filtrada}(u,v) = j(u,v) \otimes i(u,v)$ $\quad j(x,y)_a = \sqrt{2\pi}\sigma \cdot A \cdot e^{-2\pi^2\sigma^2\cdot(x^2+y^2)}$, em que A é constante e. De forma a ser mais intuitivo a expressão utilizada para a janela de gauss foi da forma: $j_a(x,y) = \sqrt{2\pi}\sigma \cdot A \cdot e^{-\frac{2\pi^2\cdot(x^2+y^2)}{N^2\cdot\sigma^2}}$ $\quad -\frac{N}{2} \le \frac{x}{N}, \frac{y}{N} \le \frac{N}{2}$ e em que N é a dimensão da janela (igual em x e y), u e v são as coordenadas do ponto da imagem, x e y são convertidos em u e v através da divisão inteira pela dimensão horizontal e vertical da janela, respectivamente. $A = \left[\sqrt{2\pi}\sigma \cdot e^{-\frac{\pi}{\sigma^2}}\right]^{-1}$ e finalmente $j(u,v) = \frac{j_a(u,v)}{\sum_{u=-\frac{N}{2}}^{N/2}\sum_{v=-\frac{N}{2}}^{N/2} j_a(u,v)}$; em que a principal diferença está no desvio padrão ser inversamente proporcional à frequência de corte. Esta alteração permite que quando se aumenta o desvio padrão, aumenta-se também o número de componentes filtradas. Isto torna-se ainda mais intuitivo se se pensar neste filtro como uma média pesada dos pixels envolventes, em que é atribuído maior peso ao pixel central e em que a relação de pesos é tanto mais ténue quanto maior for o desvio padrão. Por outras palavras quanto maior for o desvio padrão maior mais importante será a contribuição dos pixeis envolventes no cálculo da média

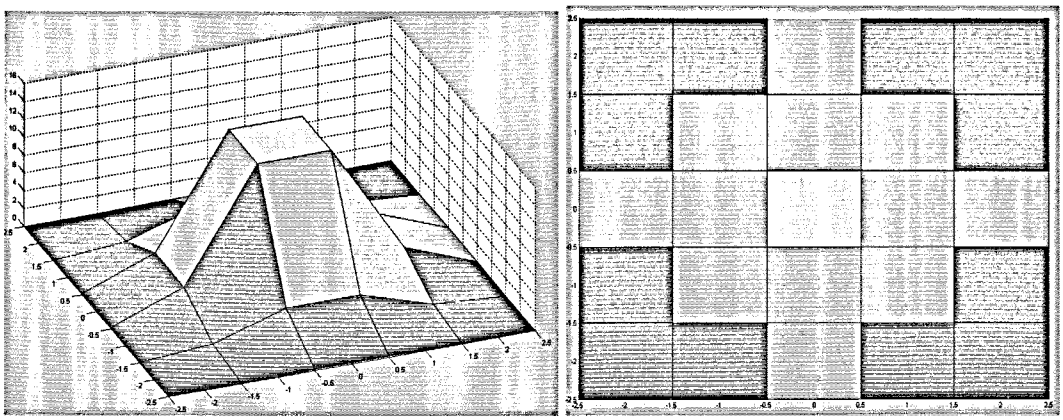


Figura 4 – Função de Gauss discretizada

A convolução idealmente deveria envolver um número infinito de valores de ambas as funções, janela e imagem, o que obviamente não é possível. Foram obtidos bons resultados limitando o tamanho da janela de Gauss a 5x5 pixels.

4.2.2 Segmentação de Cor

Para que sejam identificadas as zonas de interesse na imagem, é necessário segmentar a cor. Segmentação de cor [21, 22] é uma técnica de processamento de imagem em que, no nosso caso, se faz corresponder uma dada região de cores, no cubo RGB, a uma categoria de cor.

- modelo RGB

O modelo de cor RGB é um modelo aditivo, onde são combinadas as componentes Vermelho, Verde e Azul de diversas formas, de maneira a reproduzir as restantes cores.

O nome RGB deriva do Inglês *Red, Green e Blue* que são as componentes com as quais o modelo constrói as cores.

Há vários espaços RGB que usam o modelo RGB de cor, uma vez que este modelo não define o que é vermelho, verde ou azul, e as cores resultantes não serão exactas a menos que a contituição espectral das componentes seja definida e associanda a cada uma delas.

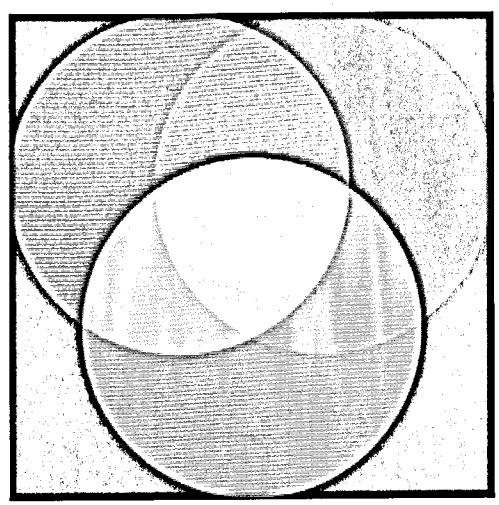


Figura 5 - Adição das componentes RGB

Normalmente representam-se as cores definidas em RGB no "cubo RGB" como se pode ver na figura seguinte.

Assim, a imagem deixa de ser vista como um conjunto de *pixels* e passa a ser vista como um conjunto de pontos que só podem ser de um dos conjuntos definidos previamente.

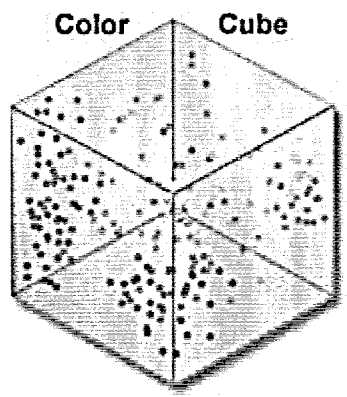


Figura 6 - Cubo RGB com conjuntos seleccionados

Este método permite identificar, por exemplo, todos os *pixels* que são da zona da cara como pertencentes ao mesmo conjunto, o que permite de seguida identificar de forma expedita as diferentes zonas de interesse.

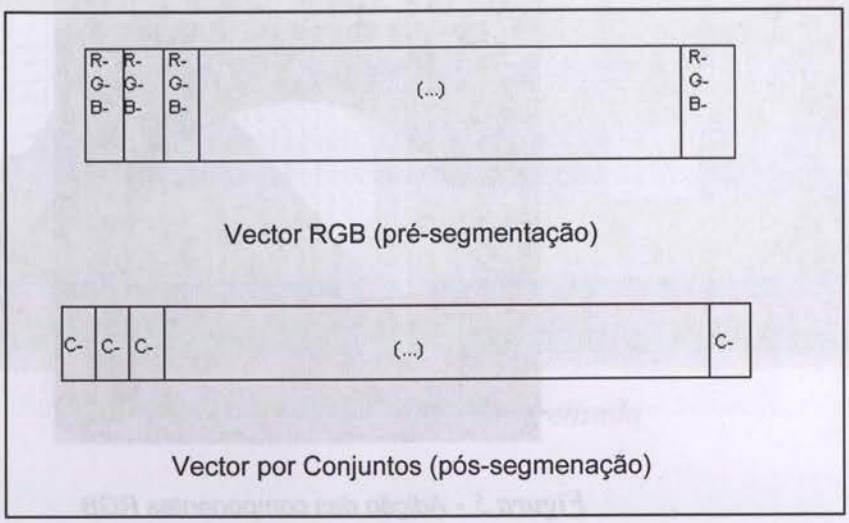


Figura 7 -(a)Imagem ; (b) Imagem Segmentada

Para que a imagem possa ser segmentada rapidamente (já foi frisada a importância da questão temporal do processamento de imagem no ponto 2.3.3.), é construída uma tabela de consulta (*Look Up Table*) que faz corresponder cada cor RGB ao um conjunto. São também guardadas informações relativas ao grau de confiança da pertença de cada cor ao conjunto e à contagem de cada cor na imagem. Esta tabela é construída inicialmente e depois caso seja necessário é alterada para melhorar a performance e tem a seguinte constituição:

Cor			Conjunto	Confiança	Contagem	Posição
R	G	B				
0	0	0	0	1	237	0
0	0	8	3	0.6	540	1
0	0	16	1	1	783	2
(...)	(...)	(...)	(...)	(...)	(...)	(...)
247	247	247	0	0	0	32385

Look Up Table

De notar que a tabela é agrupada de 8 em 8 bits para que não se torne muito extensa. Esta particularidade permite que o processo se torne mais rápido e gaste menos recursos, sendo negligenciável a perda de informação dele resultante.

Esta particularidade faz com que seja necessário calcular em que posição a cor se encontra na tabela (quer para inserir, quer para ler conjunto a que a cor pertence), assim a posição de uma cor RGB na tabela é calculada com a ajuda da seguinte formula:

$$Posição = \frac{R}{8} \times 1024 + \frac{G}{8} \times 32 + \frac{B}{8}$$

Sendo que os valores das divisões são inteiros por truncatura.

Na criação desta tabela são definidos pontos tipo das diferentes zonas a identificar (cara, cabelo, boca, fundo, ...) e é calculada a zona a afectar na tabela a partir de uma vizinhança, variável, do *pixel* escolhido e do tamanho da zona de cor em volta da cor do mesmo a afectar. Para uma vizinhança de 8 pixels, temos uma média de 9 pontos, segundo a equação (para R):

$$R(médio) = \frac{\sum_{i=0}^{9} Ri}{9}$$

Quanto maior a zona a afectar (distancia) definida, menor a precisão, maior a afectação e vice-versa. A zona envolvente do ponto é calculada como sendo a distancia tridimensional, ou seja, uma esfera com raio variável, onde todos os pontos nela contidos (cubo RGB) farão parte do conjunto a ser definido.

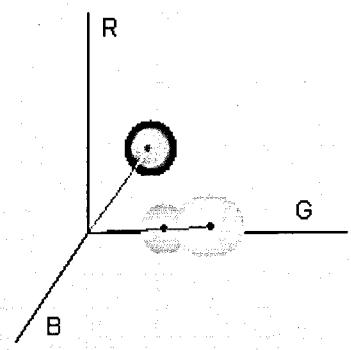


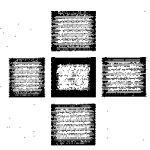
Figura 8 - Duas categorias no cubo RGB

As zonas afectadas são calculadas portanto segundo a equação que nos da a distancia de cada ponto RGB na tabela á média dos pontos em torno do ponto pivot que queremos introduzir como sendo de uma dada zona, ou saber a que zona pertence:

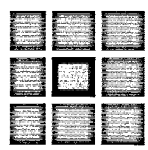
$$D^2 = (R - \overline{R})^2 \times (G - \overline{G})^2 \times (B - \overline{B})^2$$

De notar que é usado sempre o quadrado da distancia de foma a não terem de ser calculadas raízes quadradas, o que aumentaria significativamente o tempo de processamento.

Como é muito difícil saber a cor do *pixel* exacto seleccionado, a cor central da esfera é calculada como sendo a cor média da vizinhança (*8 connected*) do ponto seleccionado (como explicado anteriormente). Durante todo o trabalho usaremos esta definição preterindo a *4 connected.*



4 connected



8 connected

4.2.3 Correcção de Iluminação

Com a variação da iluminação as cores captadas pela camera mudam, pelo que a calibração com a qual a tabela de *Look Up* foi feita deixa de ser válida. Com isto, os con-

juntos pré-definidos deixam de ser bem identificados, já que o que era cor de fundo pode mudar, deixando de ser identificado como tal, resultado das diferenças de iluminação em relação a altura da construção da tabela.

Uma tabela *Look Up* não pode conter à partida todas as cores possíveis para a cara, uma vez que dependendo da iluminação e da pigmentação particular de cada individuo, este seria um conjunto muito extenso, que se acabaria por sobrepor aos restantes conjuntos, passando então a haver má classificação de zonas. O mesmo se passa em relação aos restantes conjuntos, pelo que a solução passa por fazer uma calibração da tabela em função das condições de iluminação, de forma a esta se manter válida.

O modelo de cor RGB não se presta a esta correcção, uma vez que a iluminação não pode ser variada de forma directa, dado não ter nenhum parâmetro directamente associado. Assim recorre-se ao sistema de cor HSI, este sistema tem como componentes *Hue, Saturation, Intensity*, sendo que basta variarmos a intensidade para variar o brilho de uma cor.

O sistema tem normalmente a seguinte representação espacial:

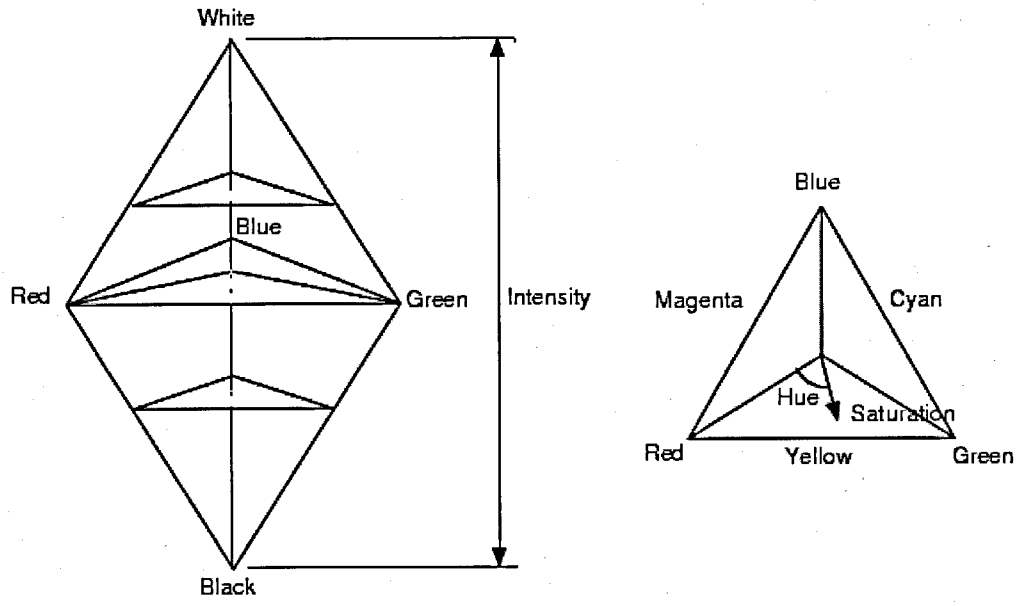


Figura 9 - Representação espacial do sistema de cor HSI

Para poder-mos aplicar este método de correcção de iluminação, teremos de estabelecer correspondência entre os dois sistema de cor, RGB e HSI, dado que a imagem é guardada num array RGB, mas que a correcção é feita em HSI.

A transformação de uma cor RGB numa HSI é dada pelas seguintes equações:

$$I = \frac{1}{3} \times (R \times G \times B)$$

$$H = \cos^{-1}\left[\frac{(R-G)+(R-B)}{\sqrt{(R-G)^2+(R-B)\times(G-B)}}\right]$$

se B > G então H=360-H

$$r = R/(R+G+B) = R/3I$$
$$g = G/(R+G+B) = G/3I$$
$$b = B/(R+G+B) = B/3I$$

$$S = 1-3\times\min(r,g,b)$$

Atendendo a que os valores de R,G,B estão entre [0-1].

Já a passagem de HSI para RGB é dada por:

$0<H<120$
$b=(1-S)/3$
$r=\frac{1}{3}\times\left[1+\frac{S\cos(H)}{\cos(60-H)}\right]$
$g=1-b-r$

$120<H<240$
$H=H-120$
$r=(1-S)/3$
$g=\frac{1}{3}\times\left[1+\frac{S\cos(H)}{\cos(60-H)}\right]$
$b=1-g-r$

$240<H<360$
$H=H-240$
$g=(1-S)/3$
$b=\frac{1}{3}\times\left[1+\frac{S\cos(H)}{\cos(60-H)}\right]$
$r=1-g-b$

$$R = 3\times I\times r$$
$$G = 3\times I\times g$$
$$B = 3\times I\times b$$

E temos os valores RGB entre [0-1], tendo em conta que H varia entre [0-360] , S e I variam entre [0-1].

Estas transformações consumiriam muito tempo de processamento, pelo que é inviável transformar cada cor RGB em HSI, corrigir I e passar novamente para RGB. Assim a partir destas equações derivou-se a forma como R,G,B se alteram com uma variação de I. A equação é dada seguidamente:

$$r \times 3I = R \Rightarrow r \times 3(I+i) = R' = \frac{R}{3I} \times (3I+i) \Leftrightarrow R + \frac{R \times i}{3I} = R'$$

Onde R é a componente vermelha original (obtida pela imagem da câmara), I o valor da intensidade dado pela equação já apresentada, i o valor da variação de intensidade luminosa e R' a componente vermelha com iluminação corrigida.

De igual modo, temos para as restantes componentes:

$$G + \frac{G \times i}{3I} = G'$$

$$B + \frac{B \times i}{3I} = B'$$

O próximo passo é saber que correcção (i) utilizar. A ideia, é a quando da construção da tabela de *Look Up*, calcular o I médio do frame com o qual a tabela foi feita e gravar junto com a tabela. De seguida, em cada segmentação feita utilizando a tabela é comparado o I e a diferença entre os I médios da frame e guardado com a tabela, é utilizado para corrigir todos os *pixels* da imagem (i).

O fluxograma seguinte pretende descrever o funcionamento do sistema.

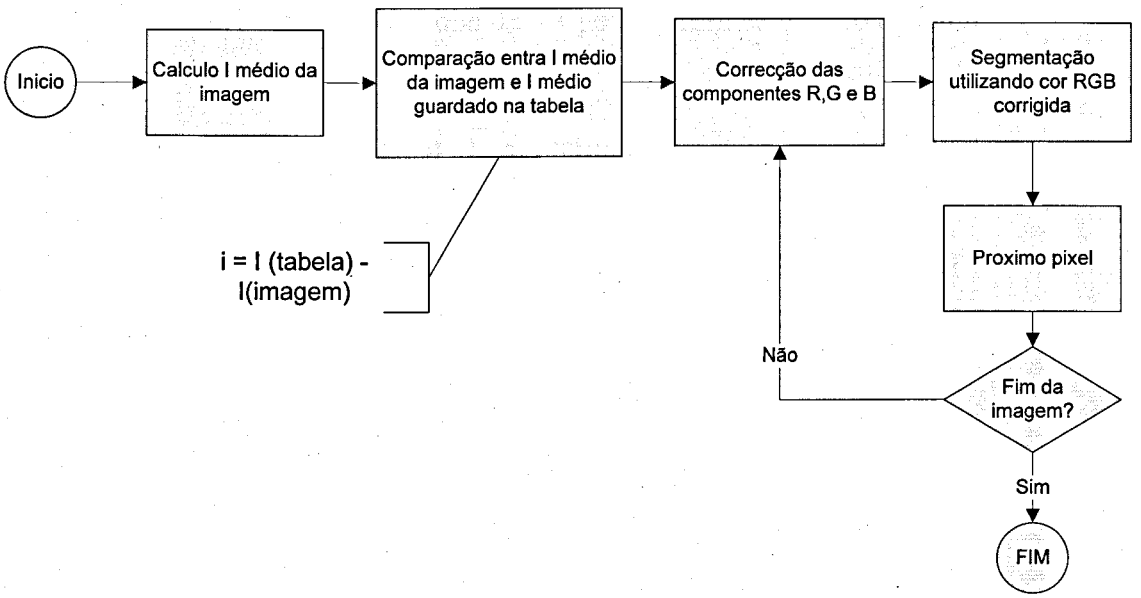


Figura 10 – Correcção da Iluminação

4.2.4 Extracção da Informação da Segmentação

Uma vez a imagem segmentada, é necessário retirar informação que sirva de entrada para a rede neuronal que calculará a saída.

Primeiramente é calculado o centro de massa da cara, para tal é feita a média da posição (x,y) de todos os pontos pertencentes ao conjunto "cara", segundo a formula:

$$Centroide_{cara}(x,y) = \left(\sum \frac{x_i}{i}, \sum \frac{y_i}{i}, \right)$$

Obtem-se portanto o centroide da cara utilizando a equação dada e procedendo como descreve o seguinte fluxograma:

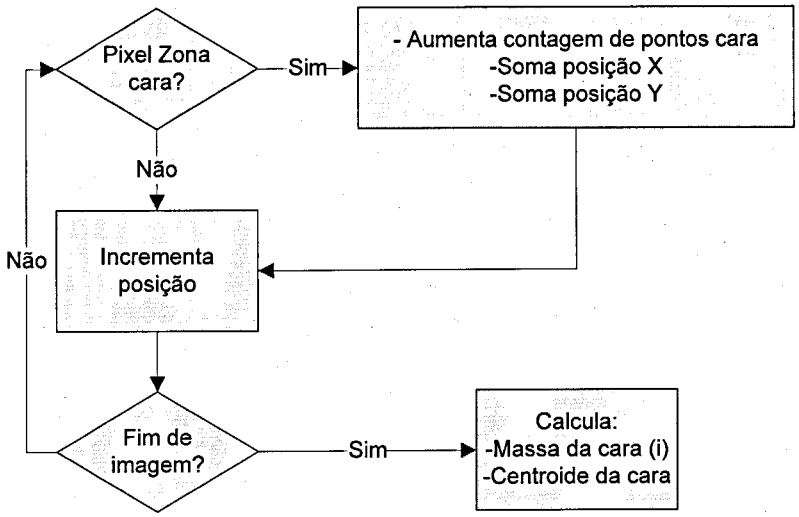


Figura 11 – Cálculo do Centróide

De seguida a partir do centroide da cara, são calculados limites, esquerdo, direito, superior e inferior de forma a obter um rectângulo envolvente da cara.

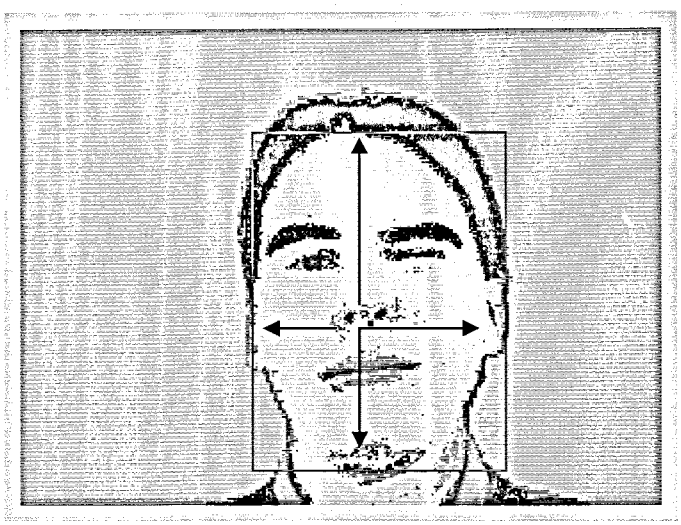


Figura 12 – Detecção de limites da cara

Neste momento a cara está identificada e delimitada na imagem, seguidamente são calculados pontos limite da cara de forma similar aos quatro limites encontrados, mas ago-

ra são calculados a partir dos limites do rectângulo envolvente, caminhando para o interior da cara.

Podemos agora ter uma delimitação mais correcta da cara unindo os limites calculados com rectas.

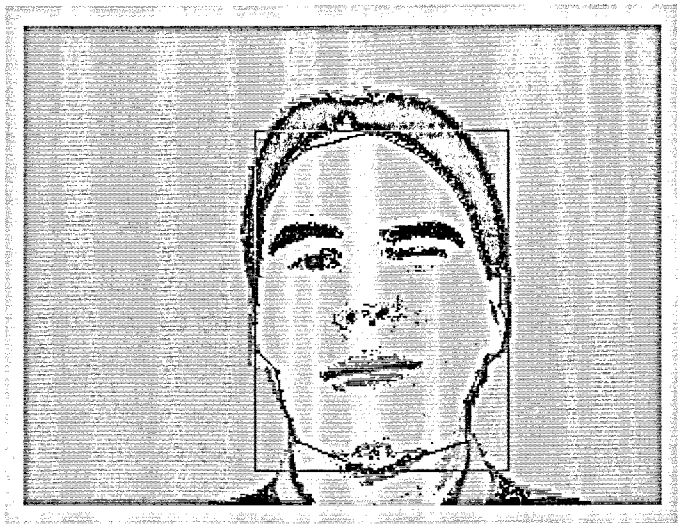


Figura 13 – Enquadramento da cara

O próximo passo é estabelecer zonas características da cara de onde possam ser extraídas informações relevantes. Dividindo o rectângulo envolvente sempre em iguais proporções obtemos zonas onde tipicamente se encontram as sobrancelhas, boca e olhos. Estas zonas têm em conta os limites calculados de forma a estarem completamente contidas na cara.

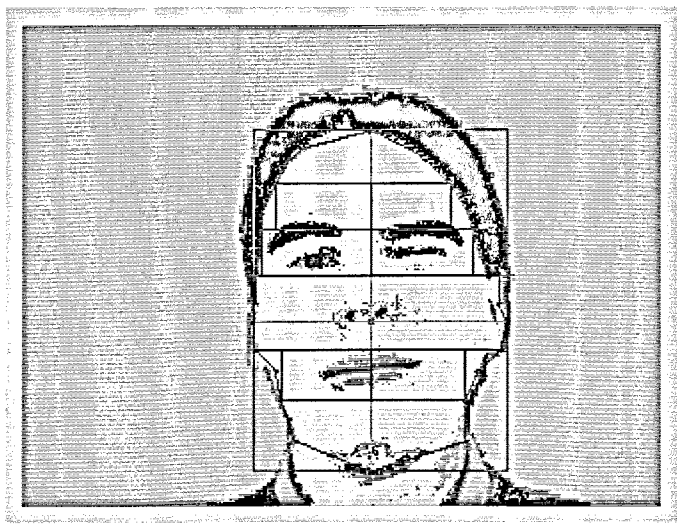


Figura 14 – Definição de zonas de interesse

http://paginas.fe.up.pt/~ee99041/IMCRI/PT

http://paginas.fe.up.pt/~ee99041/IMCRI/EN

De cada uma destas seis zonas são contados os pontos de cada conjunto: cara, cabelo/sobrancelhas, boca, sem zona. Também são calculados os diferentes centros de massa (por zona).

É apresentado em anexo, o ficheiro *Medidas.txt* para a imagem mostrada como exemplo, com toda a informação retirada da segmentação de cor.

De notar que as seis zonas coincidem tipicamente com olhos, sobrancelhas, e boca (parte direita e esquerda), numa expressão normal (neutra), mas o facto de não ficar enquadrado desta forma é por si só indicador da expressão ou posição facial.

Como se verifica no exemplo seguinte.

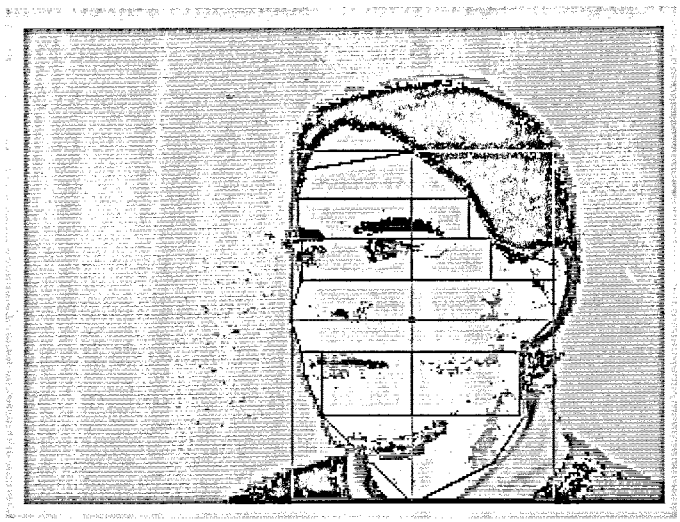


Figura 15 – Exemplo de posição desfasada com as zonas

4.2.5 Detecção de Contornos

Muita da informação sobre as características da cara encontra-se no padrão de contornos das sobrancelhas, olhos, boca e das rugas que se formam devido ao movimento da pele. Este tipo de informação encontra-se nas componentes de alta frequência espacial da imagem, ou seja, nas transições, mais ou menos abruptas, de brilho de cada uma das componentes R,G e B. Interessa então isolar estas componentes com informação valiosa do resto da imagem. Apesar de ser possível diferenciar orlas em cada "canal" R,G e B, neste

caso obteve-se melhores resultados utilizando a imagem em escala cinza e tratando apenas as variações de brilho.

O método de detectar contornos escolhido foi o Detector de Contornos de Canny. Canny especificou 3 aspectos que um detector de contornos deve tratar e em que este método em particular tenta ser óptimo:

- erro: um detector de contornos deve responder apenas aos verdadeiros contornos e deve encontrá-los todos;

- localização: a distância entre os pixels reconhecidos como contorno e os do verdadeiro contorno deve ser a menor possível

- resposta: não se deve assinalar vários pixels quando apenas um único contorno existe.

Destes critérios pode concluir-se que, a serem cumpridos, a informação obtida é apenas a informação relevante relativa aos contornos. Este aspecto é particularmente útil para esta aplicação, dado que o processamento que se segue é basicamente um processo de generalização, pelo que não é conveniente sobrecarregá-lo com informação pouco relevante. É de referir ainda que mesmo que estes critério sejam todos cumpridos, os contornos obtidos poderão não ser informação directa sobre a expressão facial, por exemplo quando um foco de luz proveniente de uma janela incide em parte da cara gera uma variação de luminosidade que será assinalada como contorno.

Uma orla é definida como uma variação com um determinado declive (em relação ao número de pixels) mínimo (e/ou máximo) no brilho da imagem. Em geral para se considerar um pixel como sendo uma orla, considera-se também a sua vizinhança imediata em direcções paralelas àquela que está a ser calculada, de forma a evitar que pequenos pontos mais brilhantes ou escuros devido ao ruído sejam considerados como orla.

O processo implementado segue a seguinte estrutura:

- "amaciar" a imagem;

- cálculo da derivada;

- supressão dos "não-máximos" locais;

- classificação

O amaciamento da imagem tem dois objectivos: eliminar o ruído de alta frequência espacial (mais alta ainda que a das orlas) e fazer com que as orlas apareçam como máximos após o cálculo da derivada. Para cumprir o segundo objectivo é necessário que o método de "amaciamento" da imagem seja a convolução com uma janela de Gauss. Já foi relatado e explicado de que maneira a janela de Gauss pode melhorar a SNR da imagem no ponto...., relembra-se que este processo é o primeiro de todos e é aproveitado também pela classificação de cores. Falta explicar de que maneira a convolução com a janela de Gauss faz com que as orlas sejam máximos.

Numa orla a função do brilho em relação ao espaço (índice do pixel) pode ser aproximada por uma rampa de declive constante. A derivada directa da orla será, portanto constante, e as partes da imagem em que não há grande declive de brilho será 0. Se se fizer a derivada da convolução da orla com a janela de Gauss (ou a convolução com a janela de gauss com a derivada da imagem, neste caso é equivalente), obter-se-á um máximo no centro da rampa:

Rampa no domínio das frequências: $\frac{1}{s^2}$ Função de Gauss unidimensional no domínio das frequências: $G = A \cdot e^{-s^2/2\sigma^2} \Rightarrow G' = A \cdot \frac{-s}{\sigma^2} \cdot e^{-s^2/2\sigma^2}$

$I \times G = \frac{declive}{s^2} \times A \cdot \frac{-s}{\sigma^2} \cdot e^{-s^2/2\sigma^2} = A \cdot \frac{-1}{s \cdot \sigma^2} \cdot e^{-s^2/2\sigma^2}$, ou seja o resultado vai ser o integral multiplicado por uma constante da função de gauss o que dá uma função com a mesma forma.

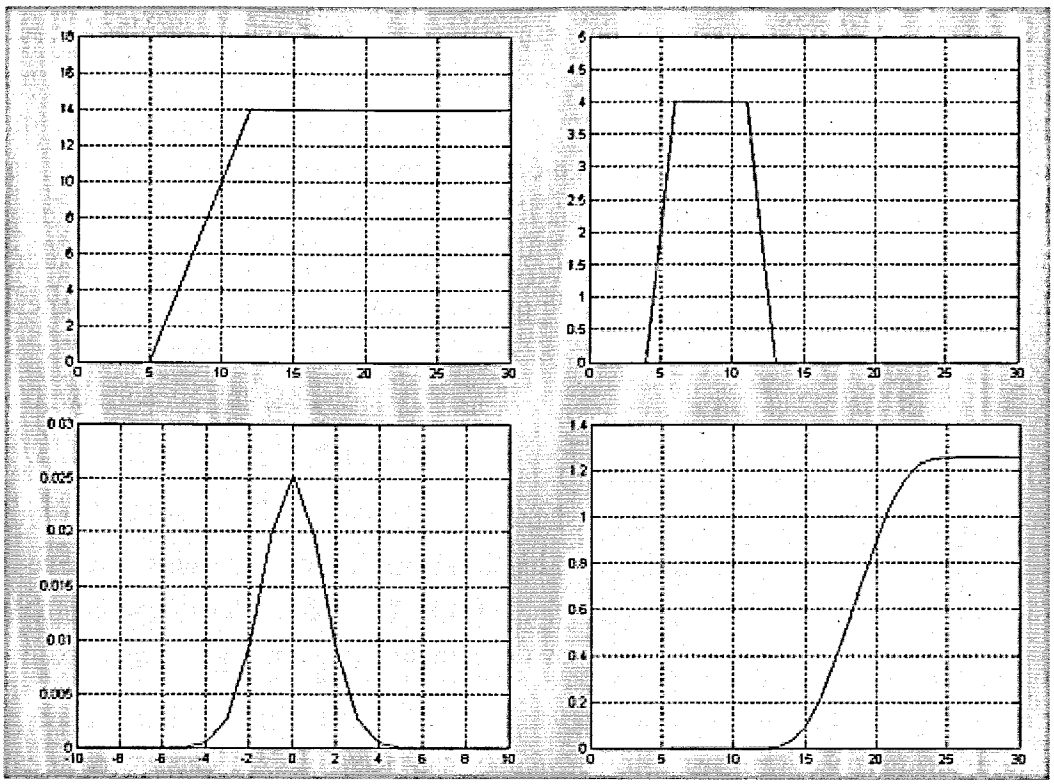


Figura 16 –Orla, Derivada da orla /Janela de Gauss, Orla filtrada

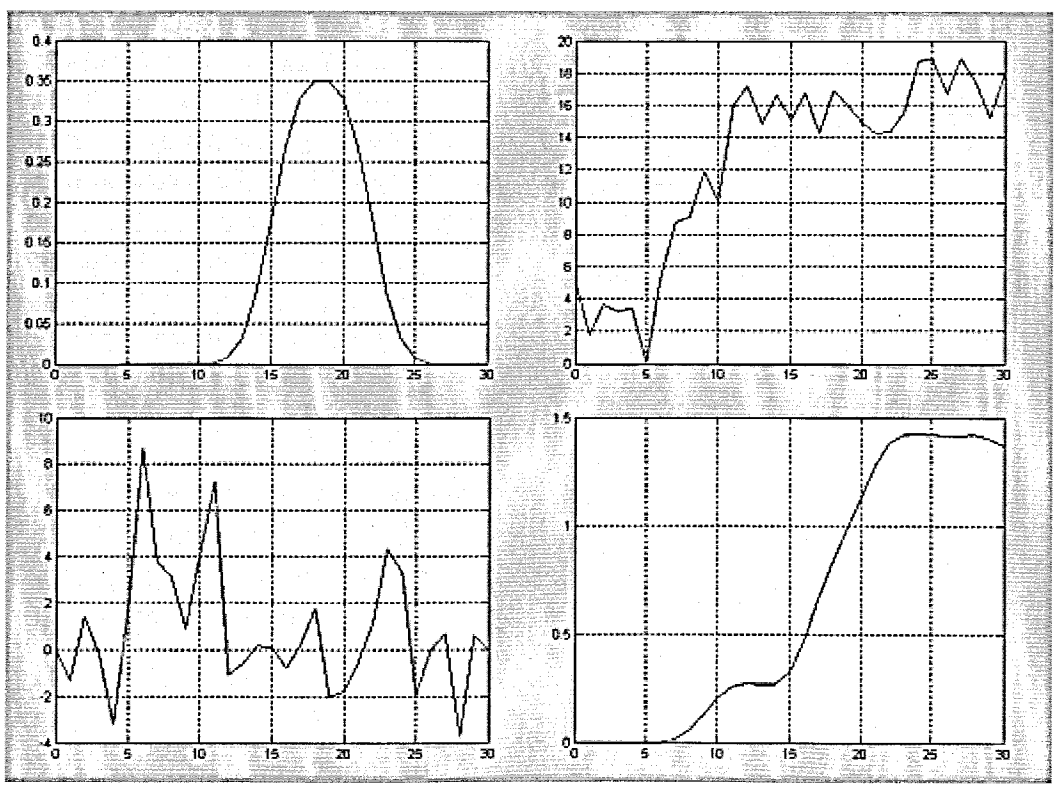


Figura 17 – Derivada da orla filtrada, Orla com ruído / Derivada da orla com ruído, Orla filtrada

http://paginas.fe.up.pt/~ee99041/IMCRI/PT

http://paginas.fe.up.pt/~ee99041/IMCRI/EN

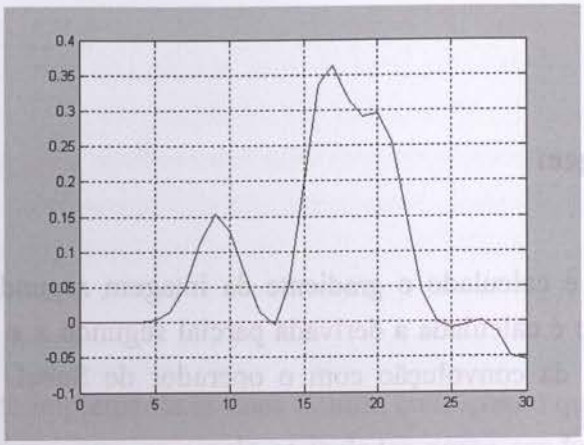


Figura 18 - Derivada da orla filtrada

Esta filtragem também vai eliminar alguns pormenores, sem ser ruído, da imagem, o que até certo ponto é desejável. Por exemplo pequenas texturas ou pormenores que não são relevantes para definir expressões faciais serão eliminados neste processo

Figura 19 – Imagem, Imagem filtrada com uma janela de Gauss 5x5 dp=1,8

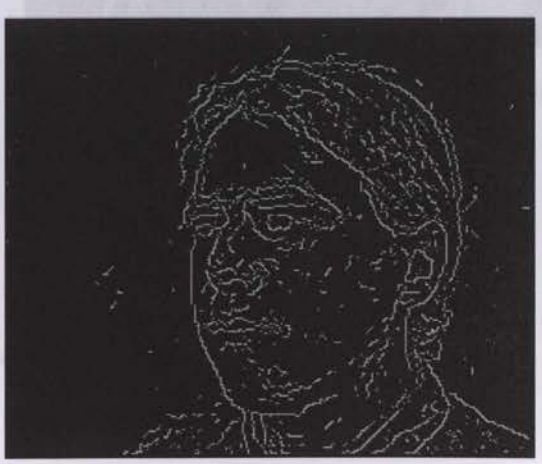


Figura 20 - Canny sem filtrar a imagem

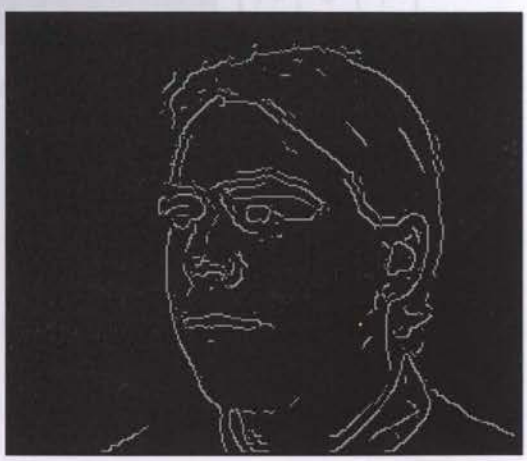


Figura 21 - Canny com filtragem

- derivada da imagem

De seguida é calculado o gradiente da imagem segundo as direcções vertical e horizontal. Primeiro é calculada a derivada parcial segundo x e depois a derivada parcial segundo y, através da convolução com o operador de Sobel. Calcula-se de seguida o módulo da derivada:

$$\nabla I = \frac{\partial I}{\partial x} + \frac{\partial I}{\partial y} \Rightarrow |\nabla I| = \left|\frac{\partial I}{\partial x} + \frac{\partial I}{\partial y}\right| \approx \left|\frac{\partial I}{\partial x}\right| + \left|\frac{\partial I}{\partial y}\right|$$

A seguir são calculados para cada pixel o ângulo da respectiva derivada parcial: $\alpha(x,y) = atg\left(\frac{\frac{\partial I}{\partial y}}{\frac{\partial I}{\partial x}}\right)$, que permite saber a orientação da orla.

-1	0	+1
-2	0	+2
-1	0	+1

Gx

+1	+2	+1
0	0	0
-1	-2	-1

Gy

Figura 22 – Janela de Sobel

Figura 23 – Sobel

- máximos locais

O passo seguinte implementa as duas últimas condições a que se propõe este método: minimizar a distância entre os pixels assinalados como sendo orla e as verdadeiras orlas, e também impedir que sejam assinalados vários pontos para a mesma orla. A ideia é pegar no gradiente da imagem e apenas deixar os valores máximos numa determinada vizinhança.

Associado a cada pixel está o módulo do gradiente o ângulo do gradiente. Para cada pixel verifica-se segundo a direcção perpendicular à do seu ângulo, se existe nos n pixels adjacentes algum com módulo superior. Se existir esse pixel não é um máximo pelo que deve ser apagado. Se não existir esse pixel é um máximo local e deve ser deixado ficar.

Figura 24 – Máximos locais

- classificação

Finalmente a classificação permite filtrar apenas a informação mais importante dos resultados obtidos até aqui. Define-se um nível máximo e mínimo de comparação. Todos os pontos com luminosidade superior ao nível máximo (o critério mais exigente), são considerados orlas. Seguidamente todos os pontos adjacentes aos pontos que cumprem o critério mais exigente, que se encontram na direcção +-alfa(xi,yi) e que têm um valor superior ao nível mínimo (critério menos exigente), são também considerados orlas. O resultado é uma imagem binária que contém a localização exacta das orlas importantes.

Figura 25 – Imagem inicial

Figura 26 – Resultado operador de Canny

Existem alguns métodos formais de avaliação dos resultados obtidos, mas estes não foram usados neste trabalho. Os resultados foram avaliados com base na sua observação visual.

4.2.6 Codificação da Informação de Contornos

A informação obtida a seguir a processar a imagem com o algoritmo de Canny encontra-se numa forma muito pouco prática de se utilizar com o método principal de reconhecimento que se segue, que é uma rede neuronal. A introdução da imagem toda numa rede neuronal, implica ter uma rede com milhares de entradas, o que não é realizável do ponto de vista prático. Deste modo é necessário um método que organize resuma e represente a informação. O ideal será identificar os contornos todos, como sendo do nariz, boca, olhos, etc; assim como as sua características relativamente a um estado normal.

Para este efeito foram estudados alguns métodos como:

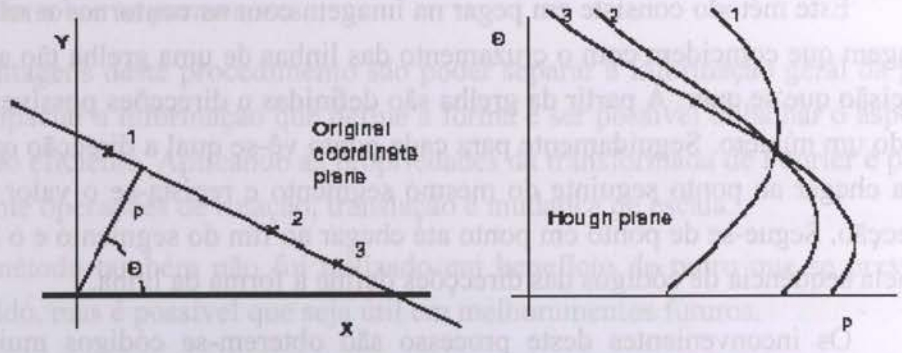
http://paginas.fe.up.pt/~ee99041/IMCRI/PT

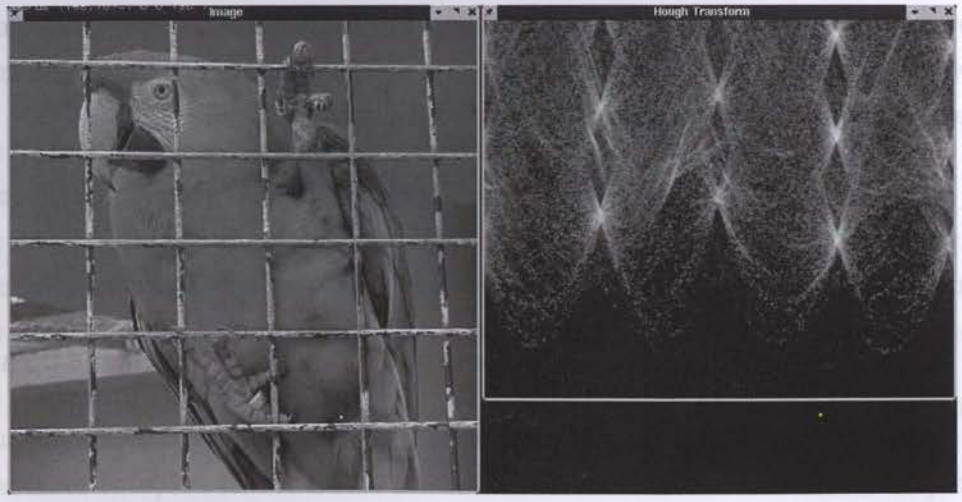
http://paginas.fe.up.pt/~ee99041/IMCRI/EN

- transformada de Hough

A transformada de Hough tem como objectivo identificar formas geométricas, numa imagem, preferencialmente binária, minimizando o número de computações necessárias. O tipo de formas geométricas que é possível identificar com esta técnica está limitado a formas que possam ser descritas por funções de parâmetros constantes: g(v,c)=0 em que v é o vector de coordenadas e c é o vector de parâmetros.

Resumidamente, a técnica consiste em criar um espaço de parâmetros da função que se pretende encontrar e para cada ponto importante da derivada da imagem calcular as várias variantes da função que pode passar naquele ponto variando os seus parâmetros. Cada intersecção no espaço paramétrico é contada e no final as intersecções com maior contagem indicam a função que encaixa nos pontos associados às rectas ou curvas que passam na intersecção. O espaço paramétrico é discreto com intervalos entre parâmetros tão grandes quanto a precisão pretendida. Uma precisão muito grande resultará num número maior de computações necessárias. O número de computações necessárias é: n*(k-1), em que n é o número de pontos da imagem e k o número de parâmetros da função.





Transformada de Hough à direita à procura de rectas descritas na forma: x*cos(teta)+ysen(teta)=rho

Isto pode ser útil para procurar certas curvas que estão directamente relacionadas com as características da face. No entanto, tendo em conta que a cara pode estar em quase qualquer posição, seria necessário adaptar as funções a procurar em função da posição. Por exemplo numa cara de frente a pupila dos olhos pode ser aproximada por um círculo, mas se estiver de lado já será uma elipse. De modo a tornar este método utilizável para este fim seria necessário identificar muito bem a posição da cara primeiro e adaptar as funções que se pretendem encontrar a essa posição. Este processamento prevê-se demasiado moroso quando se pretende obter resultados em quase tempo real, com o hardware descrito, pelo que não se considerou implementá-lo.

- chain codes

A codificação em cadeia permite obter um número ou uma sequência de números que identificam uma forma

Este método consiste em pegar na imagem com os contornos e retirar os pontos da imagem que coincidem com o cruzamento das linhas de uma grelha tão apertada quanto a precisão que se quer. A partir da grelha são definidas n direcções possíveis às quais é atribuído um número. Seguidamente para cada ponto vê-se qual a direcção que se deve tomar para chegar ao ponto seguinte do mesmo segmento e regista-se o valor dessa respectiva direcção. Segue-se de ponto em ponto até chegar ao fim do segmento e o número composto pela sequência de códigos das direcções define a forma da linha.

Os inconvenientes deste processo são obterem-se códigos muito longos como resultado e ser muito sensível a pequenas perturbações ao longo da linha devido a ruídos ou a imperfeições na extracção das linhas. Por estes motivos e pelo facto de que é difícil obter um critério fiável que permita separar as informações das linhas nas entradas da rede neuronal que se segue, este método não é utilizado.

- descritores de Fourier

Os descritores de Fourier são uma outra forma de representar e tratar a informação relativa a formas bidimensionais.

Esta técnica começa organizar os pontos da imagem como se fossem uma sequência unidimensional. Por exemplo pode-se ordenar os pontos todos a começar no canto inferior esquerdo da imagem e seguindo para a direita e para cima. A cada ponto da imagem que seja parte do contorno é associado um número complexo na forma x+jy.

Imagem: $I = \{(x_0, y_0), (x_1, y_1) ... (x_N, y_N)\}$ é passada para
$S = s[0], s[1], ..., s[N], s[k] = x_k + jy_k$

Em que x e y são as coordenadas desse ponto na imagem. Depois de fazer isto para toda a imagem, ou para todos os pontos pretendidos, é aplicada a transformada discreta de Fourier à série resultante.

Coeficientes da série de Fourier: $a(u) = \frac{1}{N} \sum_{k=0}^{N-1} s[k] e^{-j2\pi uk/N}$

Após esta operação a informação relativa à forma passa a estar codificada nos coeficientes complexos da série de Fourier. Nesta forma é possível distinguir os traços mais gerais que caracterizam a forma, dos pormenores. As componentes de baixa frequência espacial dão informação relativa ao aspecto básico da forma, e as de alta frequência contêm informação sobre os pormenores.

As vantagens deste procedimento são poder separar a informação geral da pormenorizada, compactar a informação que define a forma e ser possível trabalhar o aspecto da forma de modo eficiente. Aplicando as propriedades da transformada de Fourier é possível fazer facilmente operações de rotação, translação e mudança de escala.

Este método também não foi utilizado em benefício de outro que se prevê mais simples e rápido, mas é possível que seja útil em melhoramentos futuros.

- signatures

Este método descreve uma forma através de vectores com direcções fixas, em relação ao centro do contorno. Como resultado a informação fica representada através do ponto central e de uma função cujo domínio é o conjunto de ângulos igualmente espaçados (quanto menor o intervalo maior a precisão); e com f[α] a distância relativa a que se encontra do ponto central o ponto do contorno mais próximo seguindo segundo a direcção de alfa.

A função resultante descreve a geometria da forma sem variar com o tamanho desta e permite fazer rotações facilmente. Para rodar a forma basta transladar a função (de

forma "circular"): f(α)=f(α -rotação). Estas características acompanhadas da simplicidade de implementação e rapidez de execução, justificam a escolha deste método.

O principal problema encontrado nesta abordagem é a informação resultante não estar apresentada de forma compacta. A aplicação destes valores directamente numa rede neuronal é impraticável, seriam necessárias cerca de 100 entradas de modo a conseguir obter um bom nível de detalhe. Outro problema é que ainda não está implementado um método que garanta que determinadas linhas estão associadas a uma parte bem definida da cara. O que se consegue obter muito bem é a região da cara apenas.

Foram introduzidas algumas alterações de modo a resolver este último problema e foi introduzido um bloco adicional de forma a compactar a informação. No entanto os resultados obtidos foram maus, e no bloco de identificação a rede neuronal não convergia com este tipo de informação.

- solução escolhida:

Uma forma simples e directa de caracterizar qualquer coisa desconhecida, é comparando com outra coisa conhecida. Foi este método o escolhido para caracterizar os contornos obtidos.

Obtenção de imagens padrão:

Na fase de configuração para um determinado utilizador, são guardados vários exemplos para cada expressão facial, de modo a serem comparados durante a fase de execução com a expressão feita nesse momento. Para cada expressão são guardados 4 exemplos padrão. As imagens exemplo são sucessivamente "somadas" à medidas que são guardadas. Por somar aqui entende-se assinalar todos os pontos comuns e não comuns de cada imagem dentro da região da cara. De referir ainda que antes de somar é necessário redimensionar a zona da cara da imagem de modo a ficar do mesmo tamanho da primeira. Isto é feito utilizando coordenadas relativas em relação à altura e largura de cada rectângulo envolvente. Existe alguma perda de conectividade entre pontos, mas não é muito significativa. Ao somar imagens pretende-se evitar que no processamento posterior sejam considerados pormenores não idenfiticativos da expressão em causa.

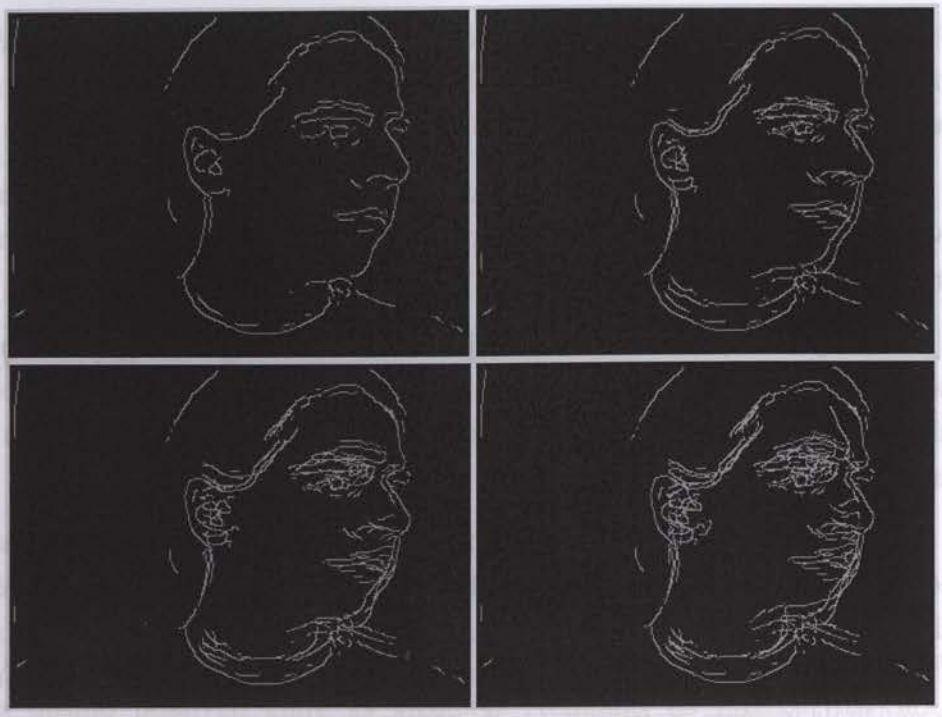


Figura 27 – Sequência de soma de contornos

Comparação de imagens:

O método de comparação de imagens escolhido foi baseado na distância de Hausdorff. A distância de Hausdorff inicialmente definida para ser aplicada na área da topologia, é também bastante útil para comparar a geometria de duas imagens ou de uma imagem e de um modelo.

Uma maneira de caracterizar dois conjuntos (fechados) iguais seria, dizer que a mínima distância de cada ponto de um conjunto a qualquer outro ponto do outro é 0 e vice-versa (necessariamente). Supondo que existe o conjunto A e B e que para cada um dos pontos de cada conjunto é associada a mínima distância a qualquer outro ponto do outro conjunto, a distância de Hausdorff é definida como sendo a máxima dessas distâncias.

Dados dois conjuntos de pontos finitos: A={a1,....,ap} e B={b1,...,b2q}, a distância de Hausdorff é definida como: H(A,B)=max(h(A,B),h(B,A)) em que h(A,B)=max min||a-b|| a pertence a A e b pertence a B || || norma euclidiana por exemplo

Através deste cálculo e recorrendo a operações como a rotação e redimensionamento de uma das imagens de contornos, é possível obter uma boa medida da semelhança no que respeita à forma.

O método realmente implementado tem em conta estes princípios, mas trata-se de uma versão simplificada, com base em certos pressupostos e com vista a minimizar o número de operações por cada padrão.

As imagens a serem comparadas são necessariamente caras e o redimensionamento óptimo para a comparação dos dois conjuntos é feito anteriormente. Portanto, é possível considerar apenas uma das distâncias (imagem 1-> imagem2 ou vice versa) em vez de ambas.

Como medida de semelhança foram tomados 2 valores: a média das distâncias e as coordenadas do centro de massa dessas distâncias. Experimentalmente, através do treino da rede neuronal utilizada, verificou-se que a média neste caso era uma medida melhor do que o máximo das distâncias. As coordenadas do centro de massa permitem ajudar a localizar a região da imagem mais diferente do padrão.

Ao executar este algoritmo verificou-se que o tempo de processamento era ainda assim muito elevado: à volta dos 10s conforme a quantidade de pontos. A implementação deste algoritmo, tem complexidade temporal O(pq), com p e q o número de pontos em cada conjunto. Deste modo limitou-se o número de pontos de cada imagem recorrendo a uma grelha de 3% da altura e largura. Contudo, esta técnica reduziu significativamente a precisão de cada imagem. O ideal para tornar esta técnica mais praticável, seria utilizar apenas alguns pontos chave dos contornos, em vez de todos os pontos de forma igual. Este objectivo não foi conseguido e fica como um aspecto a melhorar.

Figura 28 – Soma de contornos e contorno após grelha

No total são utilizados 4 padrões de contornos relativos a expressões difrentes, cada um composto pela soma de 4 imagens de contornos. Em cada ciclo, durante a execução, são calculadas as medidas das diferenças entre cada um deles e a imagem actual.

4.3 Identificação

4.3.1 Introdução

O bloco lógico da identificação tem a responsabilidade de estabelecer uma relação, tão certa quanto possível, entre as "features" retiradas da imagem da cara no pré-processamento e as expressões faciais que se pretende identificar.

Este processo, dada a natureza das medidas e das saídas, é feito num contexto de grande imprecisão. Começando pelas saídas, estas podem ser descritas como conjuntos vagos. Por exemplo a saída "Sobrancelha Direita Levantada", pode ser levantada 1cm em relação à esquerda ou 1,5cm... não é prático definir com exactidão o quanto é que a sobrancelha está levantada ou não. O que faz mais sentido neste tipo de saída é definir conjuntos (sobrepostos ou não), no qual a expressão facial pode ser definida, com mais ou menos certeza, como sendo A,B,... etc.. Quanto às entradas, a mesma medida (por exemplo percentagem de pontos "cor de boca") pode apresentar valores muito distintos, para a mesma saída, de modo não linear. Dada toda esta complexidade torna-se muito difícil estabelecer analiticamente as relações entre as entradas e saídas.

O problema exige que o software de identificação imite o modo como o ser humano avalia a expressão facial: com base em conhecimento prévio e em medidas vagas, a partir de nova informação ser capaz de generalizar correctamente.

A solução passa por recorrer ou a lógica difusa ou as redes neuronais artificiais. A diferença entre os dois métodos pode ser ténue, considerando que é possível utilizar redes neuronais para deduzir funções de pertença, ou implementar a inferência utilizadas em lógica difusa. A opção escolhida foi a rede neuronal. Com este método é possível criar a relação entre entradas e saídas de forma automática dado um conjunto de treino. Tendo em conta o número de entradas e de saídas, assim como a complexidade das relações, seria impossível fazê-lo intuitiva ou dedutivamente.

4.3.2 Considerações Teóricas

Uma rede neuronal é uma implementação simplificada que tenta seguir o comportamento verificado no cérebro humano. Tal como o cérebro humano a rede neuronal é composta por neurónios interligados. Um modelo por camadas de um cérebro simples é apresentado de seguida:

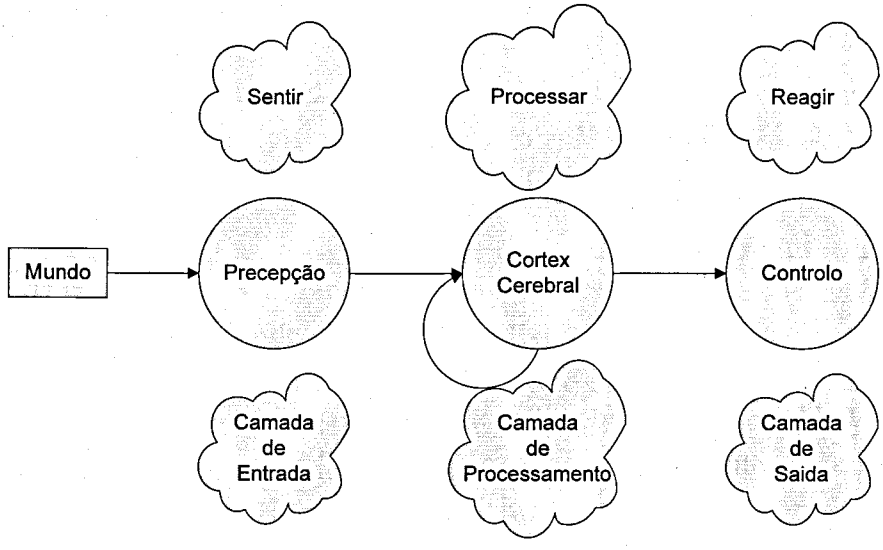


Figura 29 – Esquema de um cérebro simples

Um neurónio, numa rede neuronal, pode funcionar sozinho, tendo entradas, pesos, uma entrada de calibre (*bias*) e uma saída.

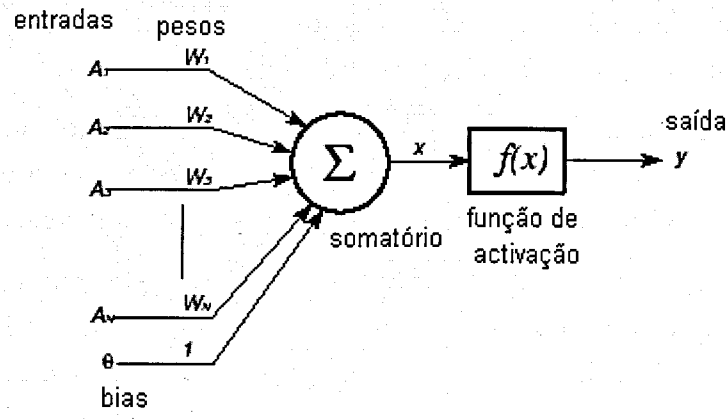


Figura 30 – Neurónio artificial

u_i – entradas, w_i- pesos, w_0 – "bias", O – saída

http://paginas.fe.up.pt/~ee99041/IMCRI/PT

http://paginas.fe.up.pt/~ee99041/IMCRI/EN

O valor do neurónio é de seguida calculado segundo a equação: $x = \sum_{1}^{m} w_i a_i + \theta$

A partir desta equação é calculado o valor de saída do neurónio a partir de uma função de activação, normalmente a função sigmoide:

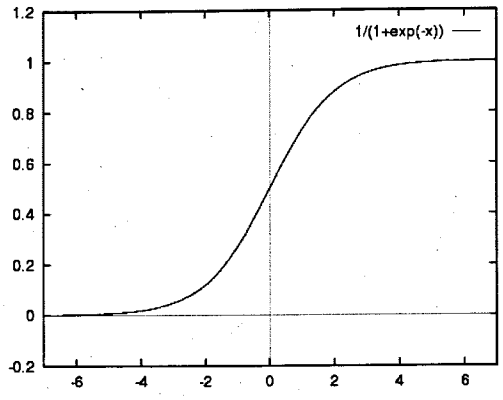


Figura 31 - Sigmoide

O perceptrão é a forma mais simples de uma rede neuronal e é usado para a classificação de padrões linermante separáveis. É constituído por um único neurónio, com pesos ajustáveis nos arcos que ligam às entradas, e uma entrada de calibre. Foi provado que dados dois padrões retirados de duas classes linearmente separáveis, este algoritmo converge de modo a situar a superfície de decisão sob a forma de hiperplano, entre as duas classes.

A saída do perceptrão $-1 < y < 1$ o hiperplano que separa as duas regiões é dado por $\sum_{1}^{m} w_i a_i + \theta = 0$; y<0 para todo o *a* pertencente a C1 e y>0 para todo o *a* pertencente a C2.

Uma rede neuronal composta por apenas um neurónio não é suficiente para o problema em causa. É necessário que a rede seja capaz de distinguir mais do que duas classes que (provavelmente) não são linearmente separáveis. Por exemplo só um neurónio não é capaz de fazer a operação "ou exclusivo".

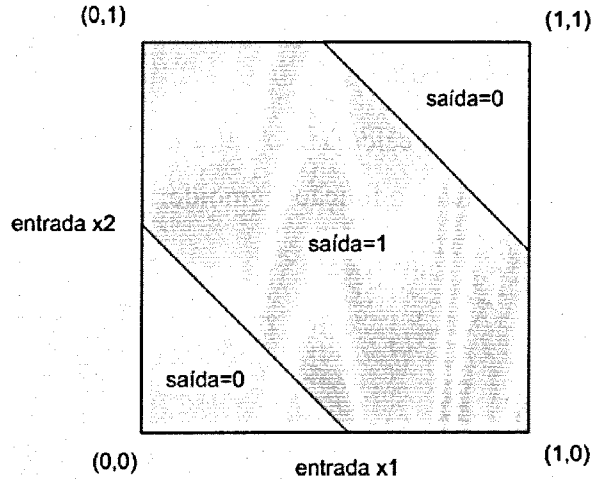


Figura 32 - XOR

Em resposta às limitações do percetrão surgiu o perceptrão multi-camada, e foi esta a arquitectura utilizada neste trabalho. Estas redes são uma generalização do perceptrão original e são capazes de classificar conjuntos não-linearmente separáveis.

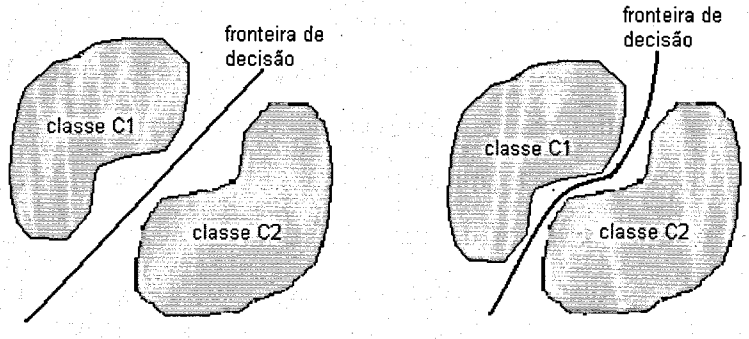


Figura 33 – Conjuntos linearmente separáveis e não- linearmente separáveis

As características principais desta rede são: um modelo de neurónio que inclui uma função de activação não-linear e "suave"; a rede contém uma ou mais camadas de neurónios escondidos (neurónios que não fazem parte das entradas ou saídas); todos os neurónios de uma cada estão ligados a todos os neurónios das outras camadas e não há ligação entre neurónios da mesma camada.

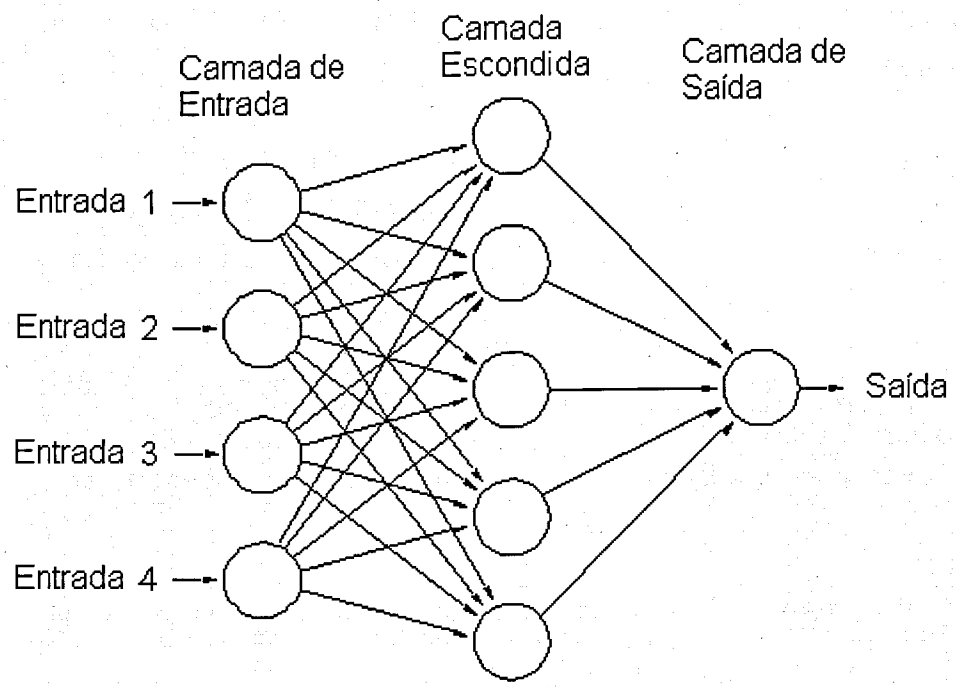


Figura 34 - MLP

A capacidade de generalização de uma rede neuronal depende sobretudo do número de neurónios em cada camada escondida e do número de camadas escondidas. Existe também alguma relação com a função de activação, mas esta não é tão importante desde que a função seja contínua e não-linear. Um perceptrão com uma única camada escondida e um número suficiente de neurónios é suficiente para fazer o mapeamento não linear das entradas. O número ideal de neurónios, neste caso será o menor número que permita fazer a generalização dos padrões correctamente. Utilizar mais neurónios do que os necessário implica maiores tempos de treino, e se forem demasiados a rede não será capaz de generalizar.

Usando as equações definidas anteriormente e utilizando o conceito *feed-forward*, cada neurónio alimenta com a sua saída os neurónios da camada seguinte, assim com as entradas dos neurónios de entrada podem ser calculados em cascata os valores de todos os neurónios até calcular, por fim a saída (ou saídas).

$$\sigma_{0=}(C_i-\mu)\mu(1-\mu);$$

C_i- Saida desejada; μ-Valor neuronio

Para todas as células da camada escondida:

$$\sigma_i=(\sum_{m:m>i} \omega_{m,j} \cdot \sigma_0)\mu_i(1-\mu_i)$$

m-celulas ligadas ao neurónio escondido

A partir destes valores são corrigidos os usando as equações:

Para pesos que ligam camada escondida a saída: $\omega^*_{i,j}=\omega_{i,j}+\rho \cdot \sigma_0 \cdot \mu_i$

Para pesos que ligam camada escondida e entrada: $\omega^*_{i,j}=\omega_{i,j}+\rho \cdot \sigma_i \cdot \mu_i$

Onde ρ é a taxa de aprendizagem (ou passo), quanto mais pequeno for, mais limitada é a variação a cada interacção, se for demasiado elevado pode levar o sistema a não convergir.

Este processo é repetido até ao erro ser satisfatoriamente minimizado, normalmente o erro é calculado para cada nó, como sendo o erro quadrático médio:

$$e.q.m.=0,5(Saida_{pretendida}-Saida_{obtida})^2$$

4.3.3 Características da Informação de Entrada

A informação de entrada deve ter uma relação o mais directa possível com as saídas de modo a tornar a generalização mais fácil. Esta preocupação foi tida em conta nos blocos de pré-processamento.

Estão a ser utilizadas como entradas:

http://paginas.fe.up.pt/~ee99041/IMCRI/PT

http://paginas.fe.up.pt/~ee99041/IMCRI/EN

- as contagens de pontos das diferentes categorias (cabelo/sobrancelhas, boca, cara, sem zona) nas seis zonas definidas e referidas anteriormente na segmentação de cor – 11 entradas.

- a média das distâncias entre os contornos da imagem com 4 contornos padrão, assim como as coordenadas x,y do ponto à volta do qual estão concentrados os maiores valores de distâncias (ver...) – 12 entradas.

Cada entrada antes de ser utilizada são normalizadas num valor entre 0 e 1. Esta normalização é obrigatória, dado que a função de activação está construída de modo a que o mínimo seja 0 e o máximo seja 1.

X=[média das distâncias com padrão 1, x_dif1, y_dif1, ..., média das distâncias com padrão 4, x_dif4, y_dif4,contagem1,... ,contagem12]

Existem algumas outras restrições que podem melhorar o funcionamento da rede, tais como: garantir que cada entrada não é correlacionada com as outras, ou normalizar de modo a que a média de cada entrada seja o mais próxima possível de 0,5. Contudo, os resultados obtidos sem estas considerações foram bastante satisfatórios.

As entradas da rede neuronal podem provir da segmentação e da obtenção de contornos, sendo que podem vir de so um dos métodos, ou de ambos simultaneamente. A decisão sobre que entradas usar dependedos testes a efectuar com diversos indivíduos e expressões a detectar. Fica assim adiada para o período após realização de testes, a escolha definitiva, de forma a termos as entradas que melhores resultados permitam obter.

Assim, estamos nesta fase a utilizar como entradas as contagens de pontos das diferentes categorias (cabelo/sobrancelhas, boca, cara, sem zona) nas seis zonas definidas e referidas anteriormente na segmentação de cor, ou seja, temos 24 entradas (ver ficheiro anexo *Medidas.txt*) .

Os valores destas entradas tem de ser normalizados de forma a conseguirmos ter uma maior homogeneidade das grandezas dos valores de entrada, uma vez que têm valores típicos e excursões distintas, o que daria maior relevância a alguns parâmetros e consequentemente prejudicaria o bom funcionamento da rede.

4.3.4 Características da Informação de Saída

No que se refere as saídas da rede neuronal, foi estabelecido um grupo *winner takes all*, que faz com que seja seleccionada como expressão de saída a que tiver o respectivo neurónio de saída com valor mais alto e apenas essa.

A lista de expressões a detectar é:

1. Abrir boca
2. Franzir testa
3. Franzir testa e nariz
4. Inclinar para a direita
5. Inclinar para a esquerda
6. Normal (neutral)
7. Sobrancelha direita levantada
8. Sobrancelha esquerda levantada
9. Ambas as sobrancelhas levantadas
10. Virar direita
11. Virar esquerda

Assim a saída da rede neuronal é um ficheiro, que tem uma série de 14 valores (um por cada expressão a identificar), que estão a 1 ou a 0. Estarão a 1 sempre que seja identificada a expressão correspondente e a 0 caso não seja identificada a mesma.

4.3.5 Tipo de Rede Neuronal Implementada

A rede utilizada é do tipo multilayer perceptron, com uma camada escondida.

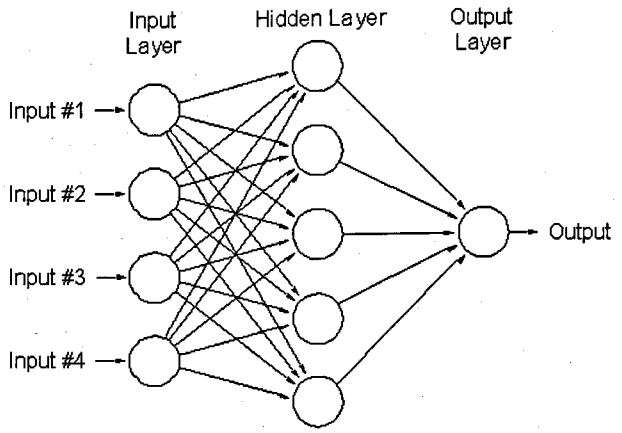


Figura 35 – Rede neuronal com uma cada escondida

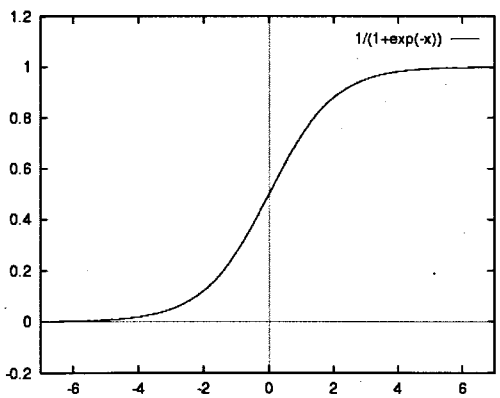
Características

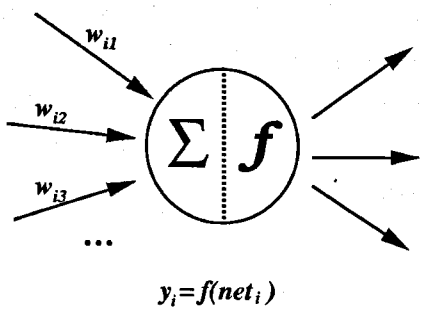
O modelo de cada neurónio deste tipo de rede inclui uma função não-linear de activação, neste caso: a função sigmoide. Contém uma camada escondida de neurónios que não faz parte nem das entradas nem das saídas da rede. Estes neurónios fazem com que a rede seja capaz de aprender tarefas complexas extraindo progressivamente os aspectos mais importantes dos padrões de entrada. Cada neurónio escondido possui ainda um "bias weight": uma entrada fixa que não depende do conjunto de entradas da rede.

Possui um elevado grau de conectividade entre camadas. Todos os neurónios de entrada estão ligados a todos os neurónios da camada escondida e todos os neurónios da camada escondida estão ligados a todos os neurónios da camada de saída. Não existe ligação entre neurónios da mesma camada e a informação circula apenas das entradas para as saídas.

A informação do erro da rede é a única que é percorrida da saída para a entrada durante a fase de treino.

Os valores ao passar de neurónio para neurónio são multiplicados pelo valor associado a cada arco de ligação. Em cada neurónio são somados todos os valores à sua entrada. Por sua vez essa valor é aplicado à função de activação 1/(1+exp(-x)), de modo a obter um valor entre 0 e 1 na saída, com uma zona de "indecisão" reduzida mas contínua.





Os neurónios escondidos são os responsáveis por distinguir os aspectos que tendem a definir um padrão no conjunto das entradas. O que se pretende com a rede é que seja capaz de calcular uma saída correcta para um padrão de entrada que nunca tenha sido introduzido antes, ou seja que tenha capacidade de generalização. O número de neurónios escondidos deve ser o mínimo que permita bons resultados. O aumento dos neurónios escondidos aumenta o tempo de processamento, sobretudo na fase de treino.

De modo à rede funcionar bem é necessário que o conjunto de treino seja o melhor possível. Neste caso o conjunto de treino é criado e avaliado pelo utilizador da interface. Este quando selecciona a opção de criar um novo conjunto de treino deve fazer a expressão que lhe é pedida e gravar as medidas (carregando em ...) quando entender que a expressão é a desejada. São pedidos 9 padrões de treino por expressão.

4.3.6 Método de Treino

O método de treino da rede utilizado foi a retro-propagação sequencial. Este algoritmo tem como função ajustar os pesos dos arcos da rede de modo a minimizar o erro quadrático médio para todos os padrões de treino apresentados na entrada.

Quando são colocadas as entradas na rede, a saída dificilmente será a desejada. Como tal, existe um erro, que é a diferença entre a saída obtida e a desejada. Este erro tem de ser minimizado de forma a obtermos saídas mais próximas do desejado. O *Backward error propagation*, ou simplesmente *backpropagation* ajusta todos os pesos dependendo do erro obtido.

As principais vantagens deste método de treino são a sua simplicidade a nível de computação e robustez - o facto da rede ser inicializada com pesos aleatórios não o impede de minimizar o erro de forma óptima.

A execução do algoritmo que implementa a rede é bastante rápida, no entanto o seu treino demora algum tempo. São utilizados no total 99 padrões de treino e cada um deles é usado com o algoritmo de retropropagação, 10000 vezes. No total está a demorar cerca de 1 minuto a treinar, de modo a obter um valor de erro pequeno. Este valor não foi considerado exagerado e por esse motivo não foram analisados os métodos de aceleração da convergência.

A velocidade de convergência pode ser controlada através do valor do coeficiente de aprendizagem. Para garantir a convergência da rede e evitar oscilações do erro durante o treino, o coeficiente de aprendizagem deve ter um valor relativamente baixo. O valor utilizado foi de 0,05.

4.3.7 Resultados

De modo a avaliar a identificação foram tiradas algumas medidas a partir da fase de treino da rede neuronal. As medidas tiradas foram o erro quadrático médio ao longo das 10k iterações e os resultados do teste com o conjunto de treino. Estes últimos resultados são antes do algoritmo "winner takes all" e permitem saber o erro médio ao identificar cada saída separadamente. Deste modo é possível avaliar quais as expressões que vão ser melhor identificadas e, consequentemente, o nível de relação entre as entradas e cada expressão. O erro quadrático médio permite saber se a rede no seu todo converge ou não e quão bem o faz.

Foram gerados gráficos recorrendo a um script de Matlab, de modo a ser mais rápido de avaliar o coportamento geral.

Características da rede utilizada:

Rede Neuronal com uma cada escondida;

Função de activação: sigmoide (entre 0 e 1);

nº de neurónios de saída: 11;

algoritmo de aprendizgem: retropropagação simples;

coeficiente de aprendizagem: 0,05;

Erros de teste $e[i] = 0{,}5 \cdot \left(y[i] - y_{esperado}[i]\right)^2$ $erro = \frac{1}{99}\sum_{1}^{99} e[i]$ e i é o índice no conjunto de treino: 1<i<99.

Teste utilizando apenas a segmentação de cor:

Erro quadrático médio:

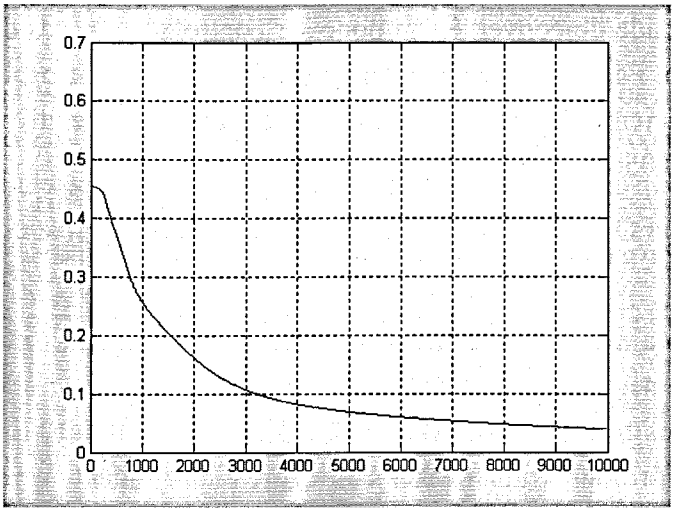


Figura 36 – Erro de Treino (segmentação)

Erro quadrático médio final: 0,0039383

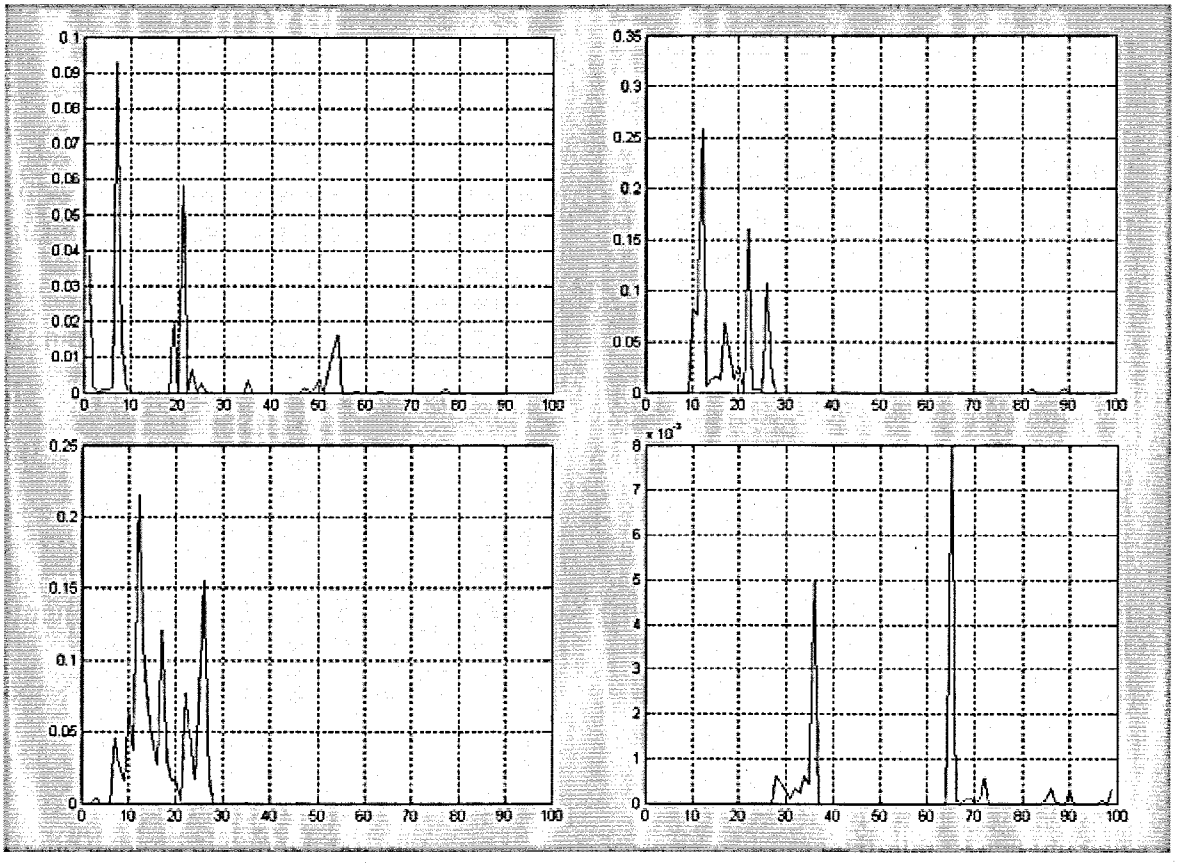


Figura 37 – Erro saídas: boca aberta, cara franzida / sobrolho e nariz f., cara inclinada dir.

http://paginas.fe.up.pt/~ee99041/IMCRI/PT

http://paginas.fe.up.pt/~ee99041/IMCRI/EN

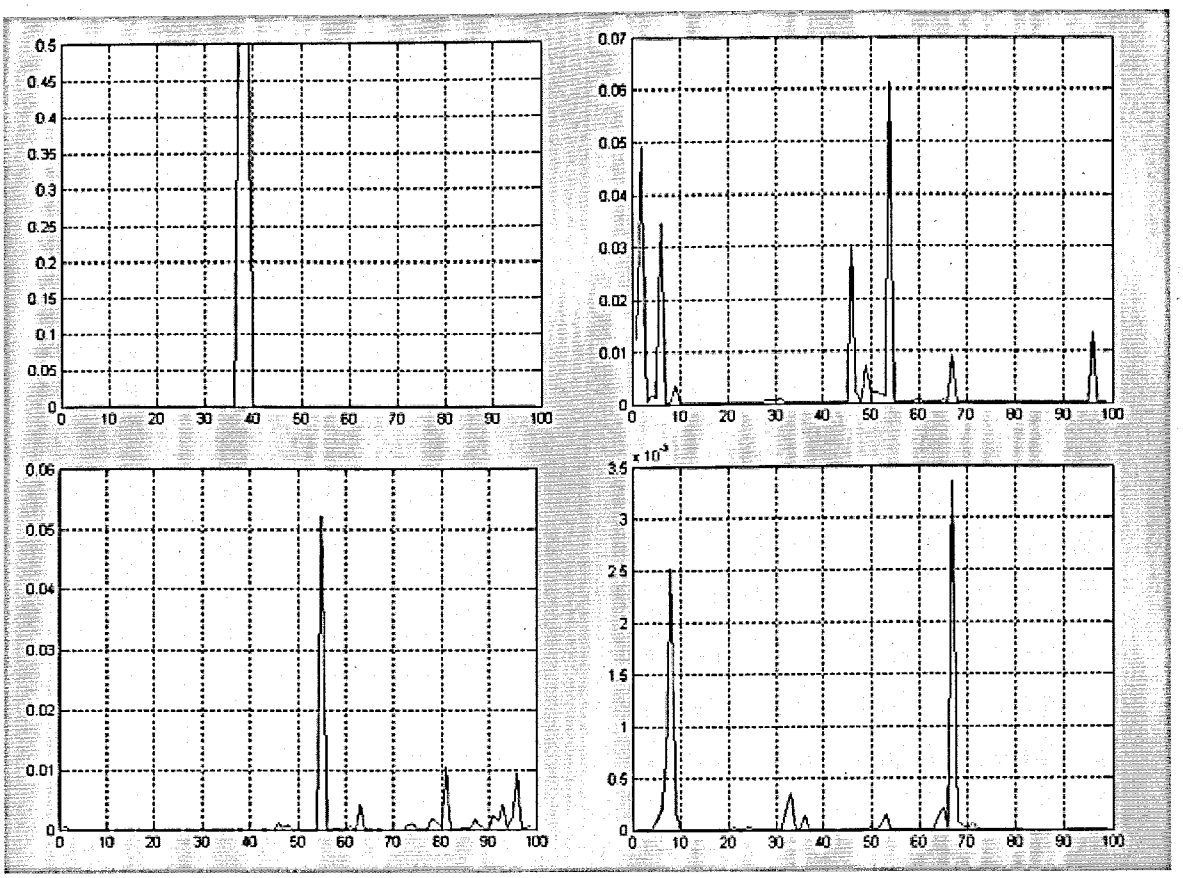


Figura 38 - Erro saídas: cara inclinada esq., normal / sobrancelha dir., sobrancelha esq.

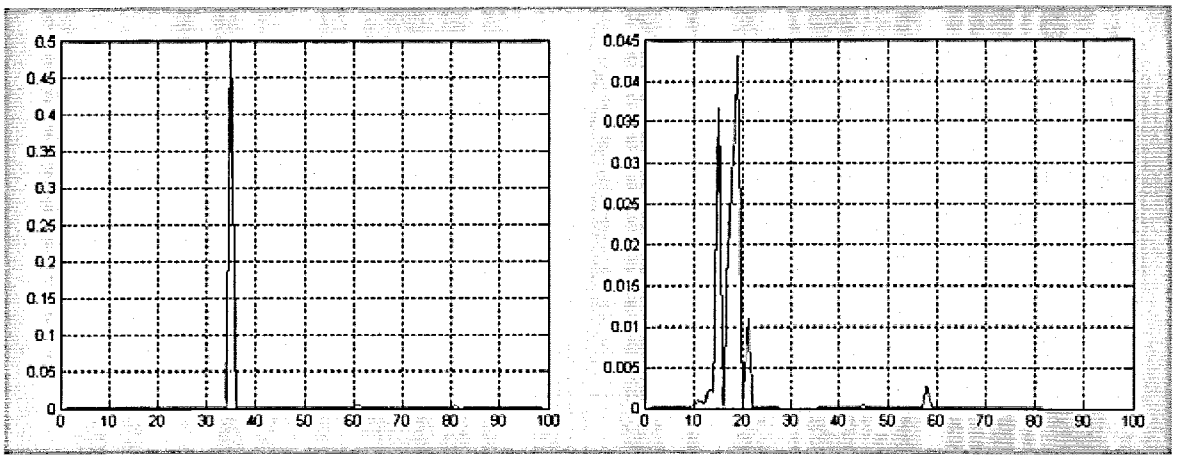


Figura 39 - Erro saídas: sobrancelhas levantadas, virada dir

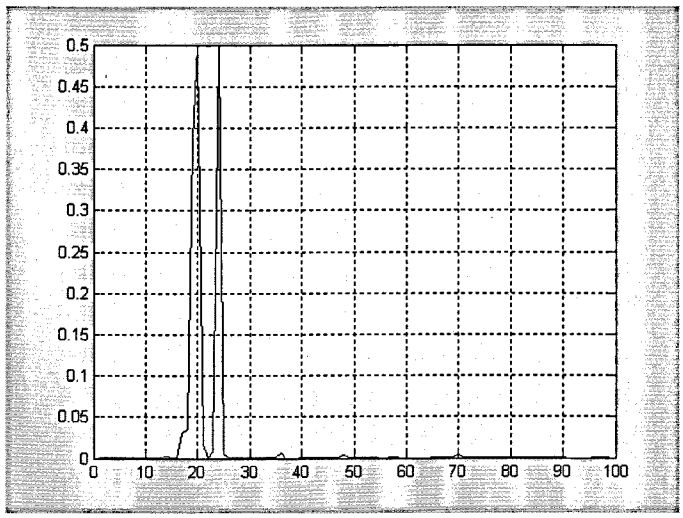


Figura 40 - - Erro saídas: virada esq.

Teste utilizando somente contornos

Características da rede utilizada:

Rede Neuronal com uma camada escondida;

Função de activação: sigmoide (entre 0 e 1);

nº de neurónios de saída: 12;

algoritmo de aprendizagem: retropropagação simples;

coeficiente de aprendizagem: 0,05;

Erro quadrático médio:

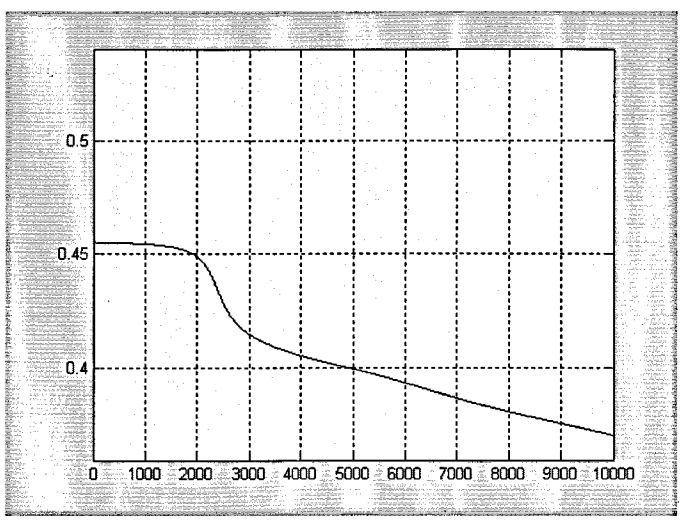


Figura 41 - Erro de Treino (contornos)

http://paginas.fe.up.pt/~ee99041/IMCRI/PT

http://paginas.fe.up.pt/~ee99041/IMCRI/EN

Erro quadrático médio final: 0,0039383

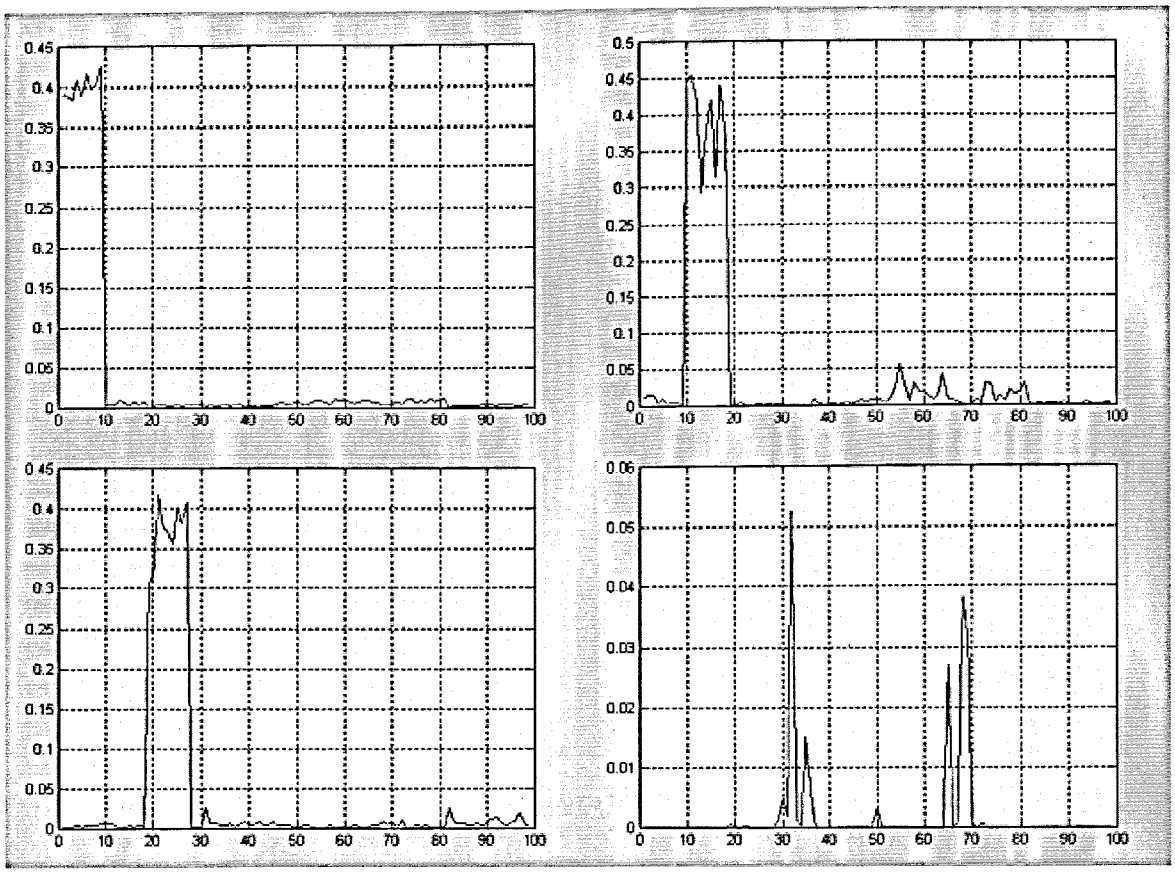


Figura 42 - Erro saídas: boca aberta, cara franzida / sobrolho e nariz f., cara inclinada dir.

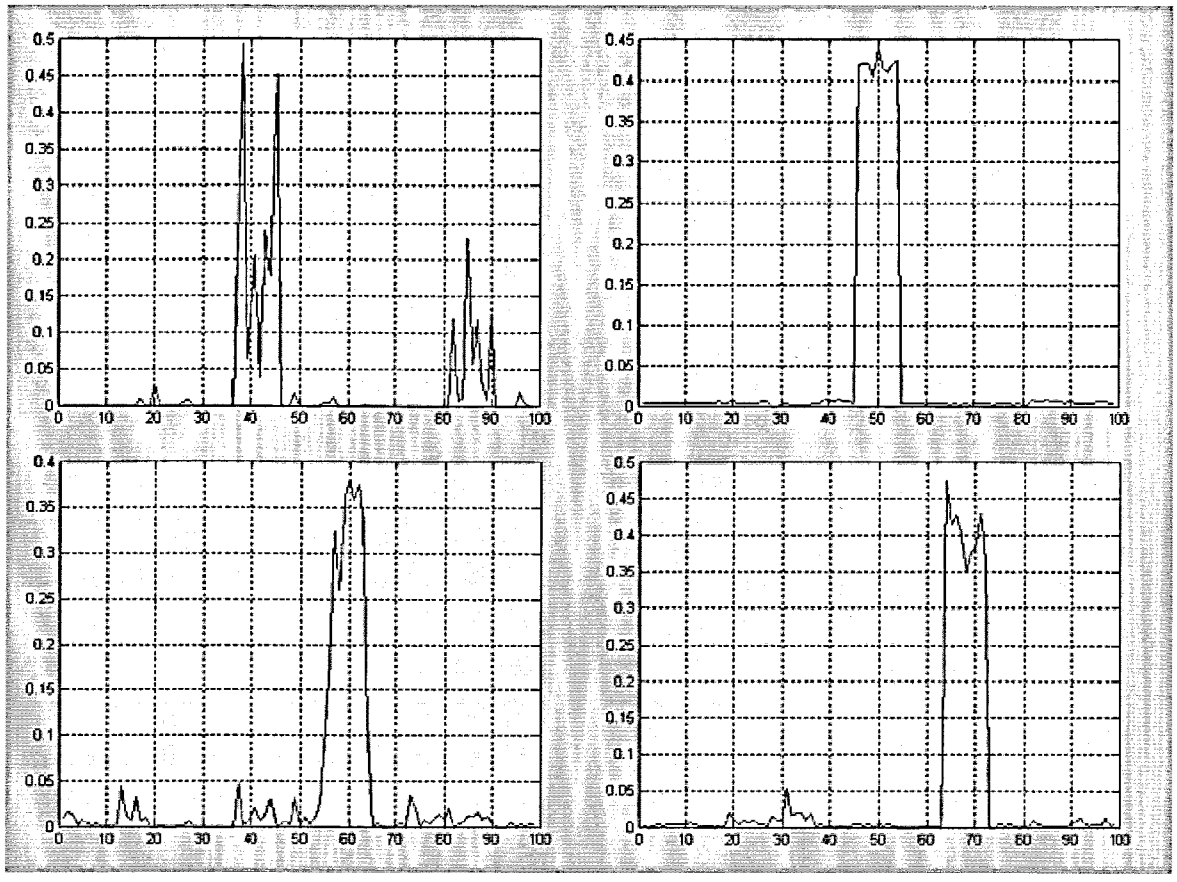


Figura 43 - Erro saídas: cara inclinada esq., normal / sobrancelha dir., sobrancelha esq.

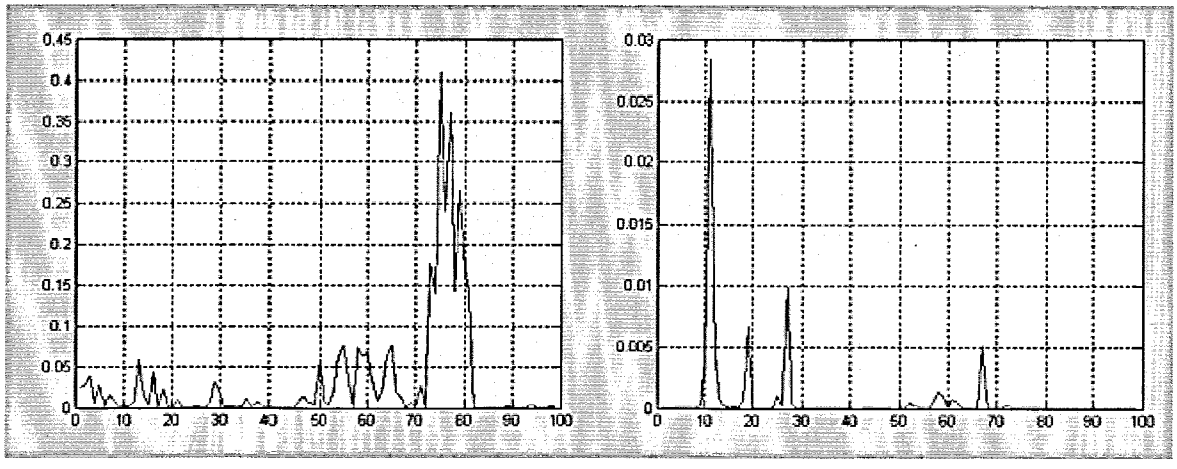


Figura 44 - Erro saídas: sobrancelhas levantadas, virada dir

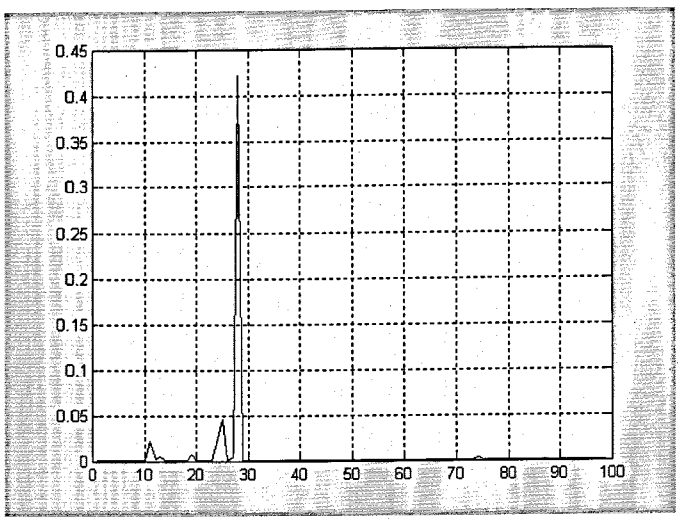


Figura 45 - Erro saídas: virada esq.

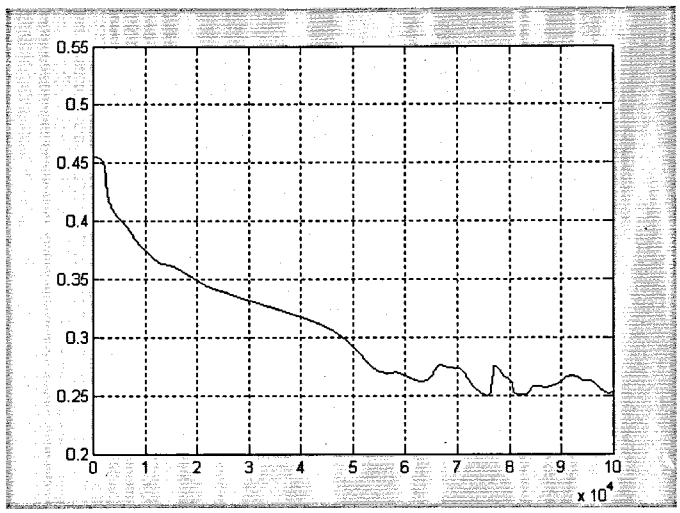


Figura 46 – Erro de treino para 100000 iterações (contornos)

Teste utilizando a segmentação de cor e contornos:

Erro quadrático médio:

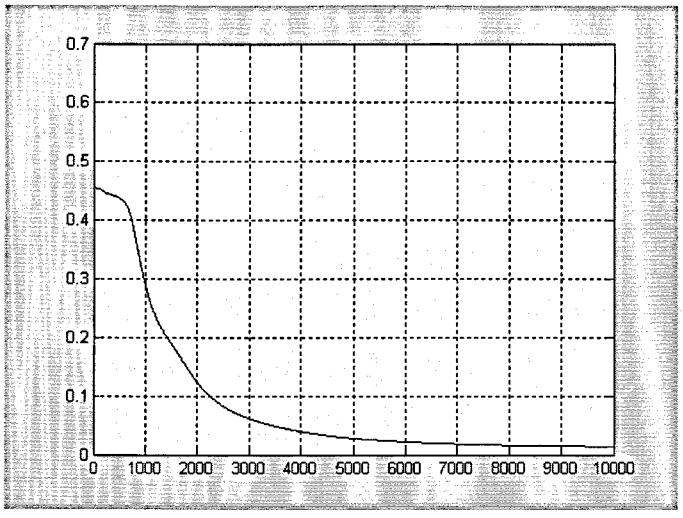


Figura 47 – Erro de treino (combinação)

Erro quadrático médio final: 0,01385

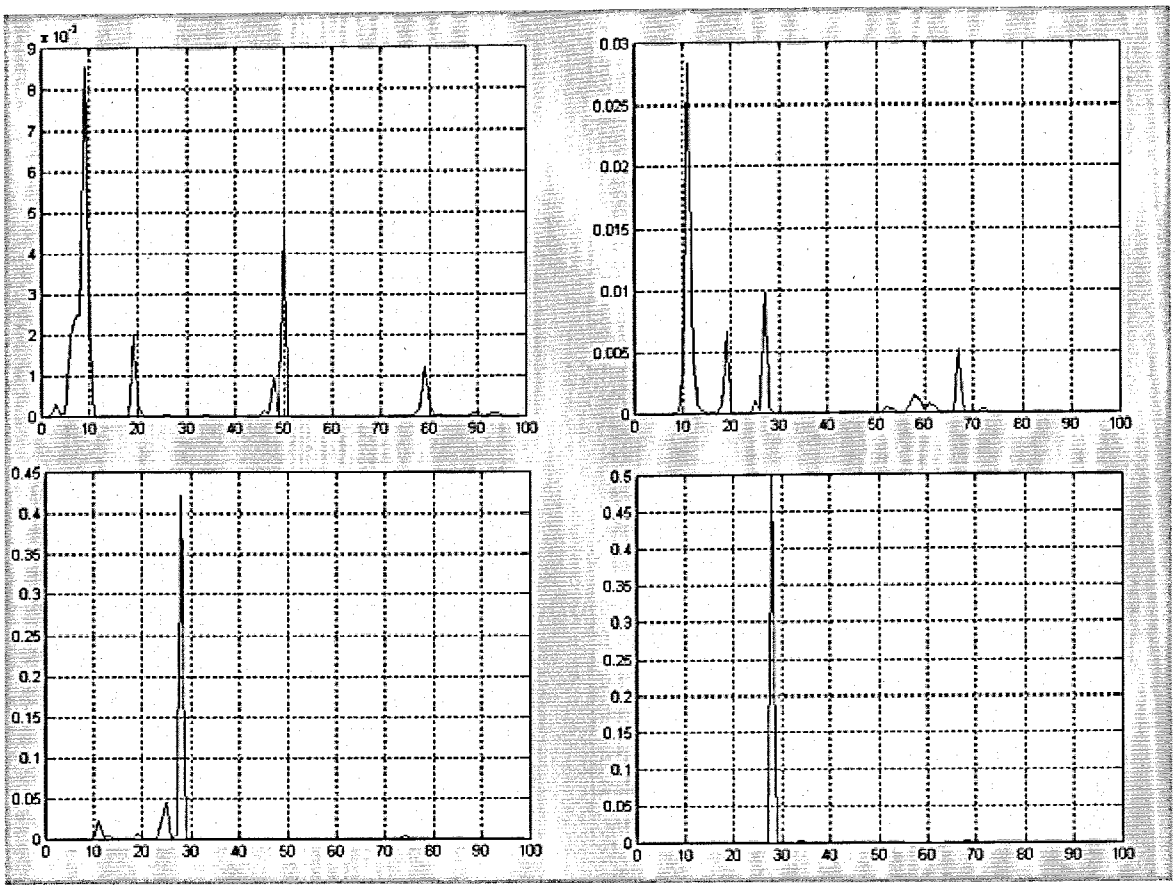


Figura 48 - Erro saídas: boca aberta, cara franzida / sobrolho e nariz f., cara inclinada dir.

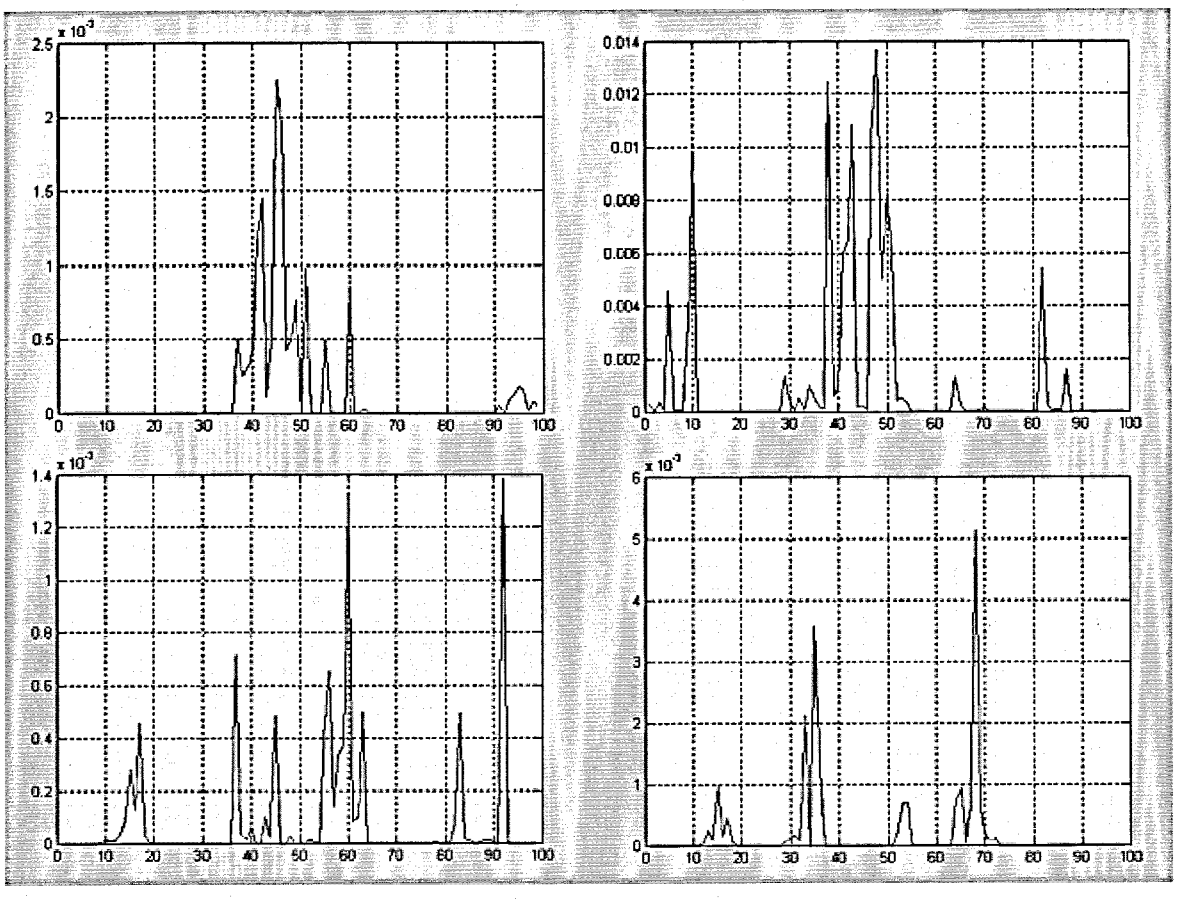


Figura 49 - Erro saídas: cara inclinada esq., normal / sobrancelha dir., sobrancelha esq.

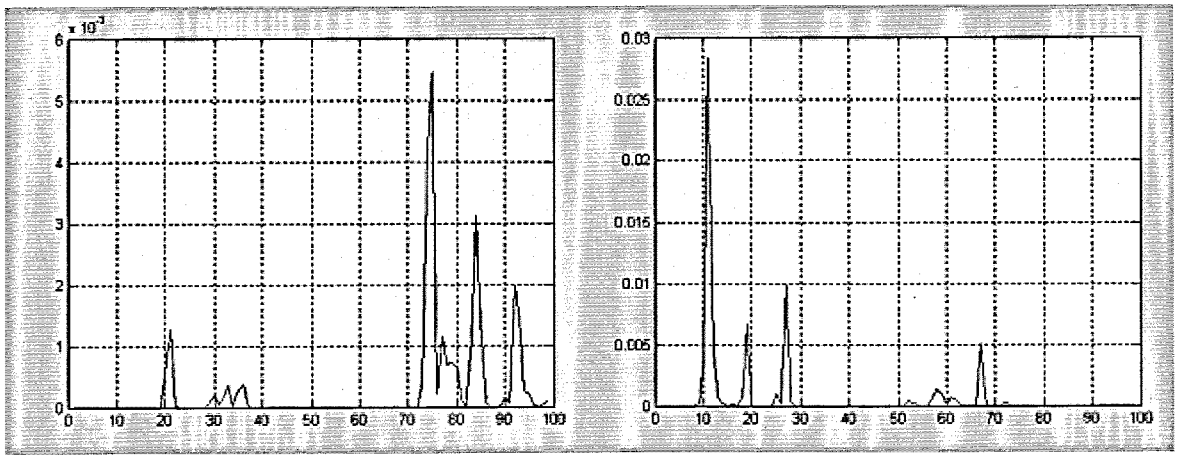


Figura 50 - Erro saídas: sobrancelhas levantadas, virada dir

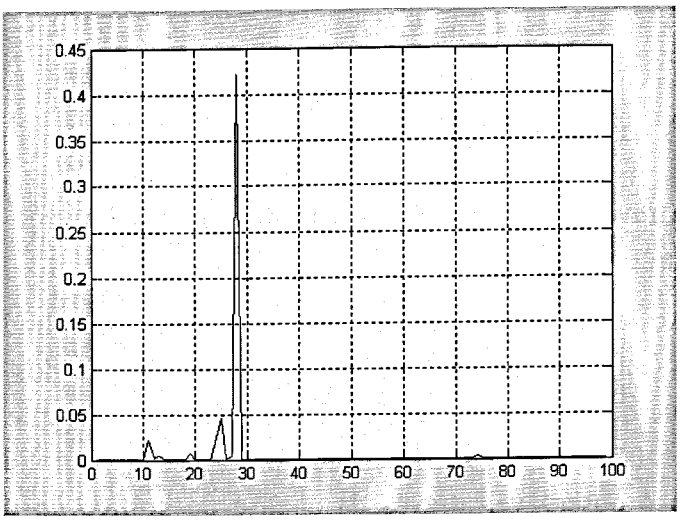


Figura 51 - Erro saídas: virada esq.

	Só segementação	Só contornos	Ambas
Cara com a boca aberta	0.0033	0.0029	2.8629e-004
Cara Franzida	0.0130	0.0092	6.7470e-004
Sobrolho e nariz franzidos	0.0144	0.0125	0.0055
Cara Inclinada para a direita	0.0052	1.8472e-004	0.0051
Cara inclinada para a esquerda	2.9394e-004	0.0152	1.3779e-004
Cara normal	0.0032	0.0024	0.0012
Sobrancelha direita leva	0.0025	0.0010	8.5787e-005
Sobrancelha esquerda levantada	1.9926e-004	8.6771e-005	1.9540e-004
Sobrancelhas levantadas	6.1088e-004	0.0053	2.7159e-004
Cara virada para a direita	4.7534e-004	0.0015	6.7470e-004
Cara virada para a esquerda	5.7220e-004	0.0152	0.0055

4.3.8 Análise de Resultados

Através da observação dos resultados dos 3 testes efectuados, é possível tirar várias conclusões.

Ao fazer o teste utilizando separadamente as entradas da segmentação de cor e as entradas resultantes das medidas derivadas dos contornos, é possível determinar quão boas são individualmente. Observou-se que as medidas resultantes da segmentação de cor são bastante melhores, que as obtidas através dos contornos. Isto é observável através da análise do gráfico do erro quadrático médio, Assim como o seu valor final. As medidas obtidas através da comparação de imagens tiveram dificuldade em fazer a rede convergir. Daqui conclui-se que este conjunto de medidas precisa de substituído. Os maus resultados apresentados no teste com contornos provém provavelmente de se estar a utilizar uma grelha muito "larga", o que torna as medidas demasiadamente imprecisas.

A análise do erro obtido individualmente para cada saída utilizando a rede com o conjunto de treino depois de treinada, permite saber se existe alguma expressão ou conjunto de expressões que são mais fáceis de identificar. Observou-se que a expressão "Cara Franzida com Nariz" apresentou os piores resultados nos 3 testes, pelo que as medidas utilizadas não devem ser as melhores para deduzir esta expressão. Quanto às outras saídas as variações de erro justificam-se sobretudo pelas diferenças entre os conjuntos de treino utilizados. Logicamente que deve ser dada preferência às expressões com menor erro na selecção da linguagem de comando.

Capítulo 5

5. Implementação

A interface está a ser desenvolvida em C++ e a ferramenta de desenvolvimento escolhida foi o Visual Studio 6.0. São conhecidas as características da linguagem de programação C++, mas relembra-se aqui a sua modularidade e popularidade. A utilização de uma linguagem de programação bastante conhecida facilita a análise e possível partilha de ideias no futuro, assim como a integração de algoritmos já desenvolvidos e optimizados, tornando o processo de desenvolvimento mais eficiente.

No desenvolvimento da interface com o utilizador recorre-se à "application programming interfaces" (API) do Windows. Isto significa que o seu funcionamento se destina principalmente ao sistema operativo Windows (poderá eventualmente funcionar noutros sistemas através de emuladores). As vantagens desta escolha situam-se na utilização generalizada deste sistema a nível do utilizador doméstico e à sua facilidade de interacção com o utilizador inexperiente. Será provável que o utilizador da cadeira de rodas, no caso de ter um computador portátil esteja a utilizar este sistema operativo e que não queira ter de se preocupar, ou preocupar outra pessoa, em configurar outro sistema operativo. De qualquer modo o sistema em si é passível de ser mudado para outra plataforma com algumas alterações no programa principal- são utilizadas algumas funções da MFC no programa principal e o acesso aos drivers da webcam depende dos SO também – e criando uma interface nova na totalidade.

API Win 32 e Microsoft Foundation Classes (MFC)

A API Win 32 é um conjunto de chamadas ao sistema, que permite programar a interface gráfica com o utilizador, nomeadamente a criação de janelas e passar as imagens da câmara, de forma simples. A MFC é uma biblioteca de classes que permite utilizar estas chamadas através de um programa em C++.

O facto de se recorrer ao sistema operativo para criar a interface tem várias vantagens. A primeira será a rapidez de execução, comparativamente com um sistema construído em cima de uma máquina virtual. No caso de haver necessidade de utilizar tempo real (não é utilizado mas pode vir a ser necessário) é também vantajoso porque permite garantir a consistência do thread de maior prioridade e do handler de interrupção de maior prioridade, assim como a alteração do tempo de processamento de cada thread.

O problema de recorrer à API Win32 é que o funcionamento do executável vai depender do sistema operativo e da CPU, o que não é muito importante para este trabalho. Conclui-se que é claramente vantajoso recorrer directamente às funções do sistema operativo neste caso.

O ideal num sistema deste género será separar tudo o que é processamento de imagem e identificação, de tudo o que é interface, em objectos diferentes. A ideia é ter sempre em vista a evolução do sistema, seja por quem iniciou o projecto, seja por outras pessoas. Seguindo esta organisação é possível trabalhar, individualmente em cada bloco, ou substituir um deles, e também tornar mais fácil a interpretação do código por parte de quem não está familiarizado com ele.

Isto foi conseguido em parte. A rede neuronal foi implementada directamente na mesma classe da interface com o utilizador, assim como a tabela de relação entre a expressão identificada e o comando correspondente. Este aspecto deve ser um dos primeiros a ser corrigido, numa implementação futura.

5.1 Interface com o Utilizador

A interface com o utilizador foi construída com vista em ser fácil de usar, fornecer toda a informação necessária à utilização e monitorização do sistema, e evitar ao máximo que o utilizador tenha de conhecer pormenores da implementação desnecessários. Tendo estes princípios em conta, considerou-se que uma interface gráfica seria preferível a uma interface do tipo linha de comandos.

Existem os seguintes modos de funcionamento: Executar, Configuração do Pré-Processamento, Configuração da Identificação, Configuração da Linguagem de Comando e Processamento de Imagem. Em qualquer destes modos é fornecida a imagem capturada directamente pela câmara sem processamento, e a imagem processada.

Execução: neste modo, uma vez carregadas as definições da identificação (através de simples cliques em botões), o programa encontra-se a interpretar as expressões faciais e a executar os comandos. É possível também observar os vários passos do processamento

http://paginas.fe.up.pt/~ee99041/IMCRI/PT

http://paginas.fe.up.pt/~ee99041/IMCRI/EN

de imagem: filtro de gauss, classificação e cor, operador de canny e o resultado final (inclui a grelha na zona da cara após a extracção de contornos).

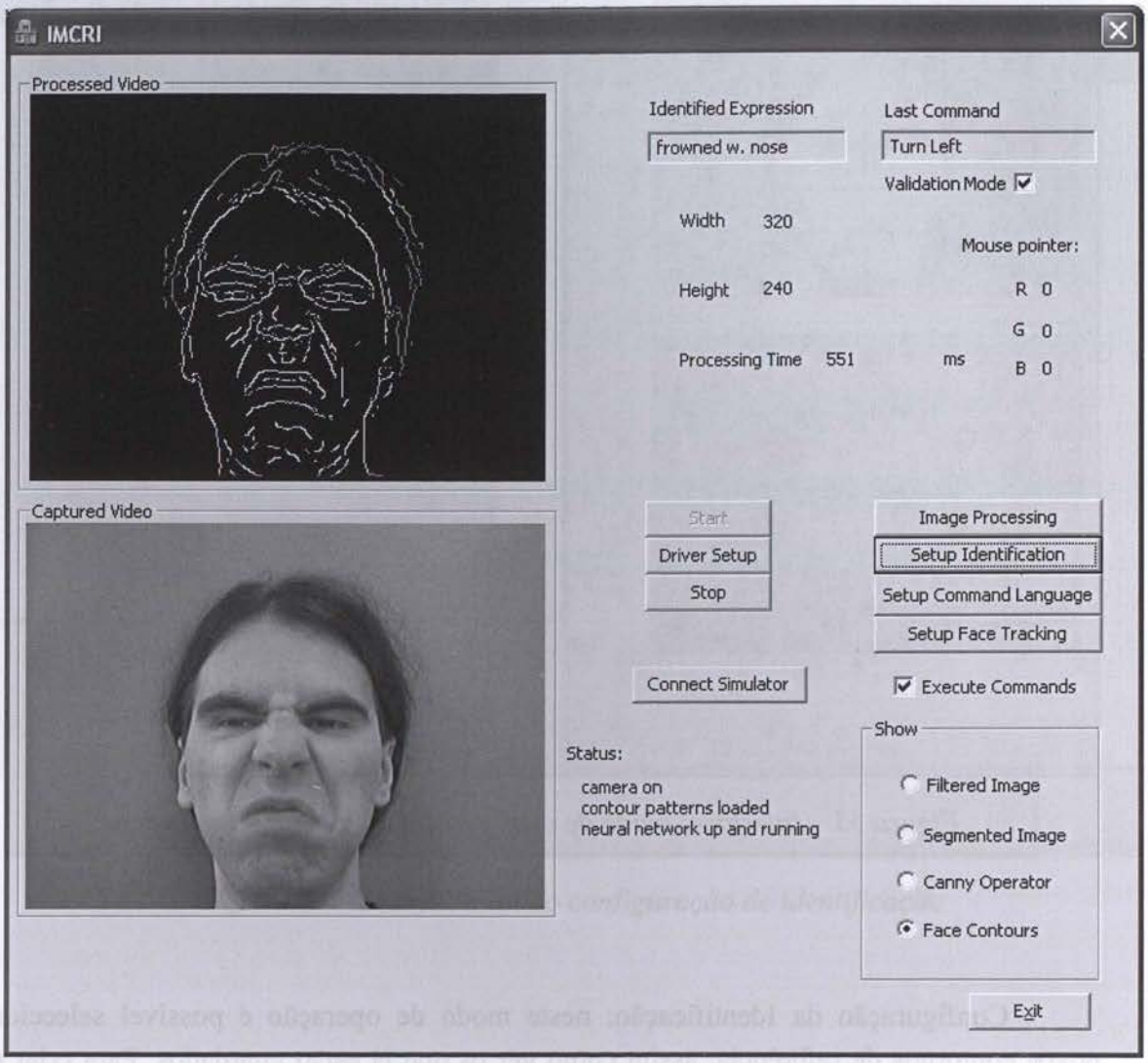


Figura 52 – Interface modo de execução

Configuração do Pré-Processamento: permite ao utilizador ajustar a segmentação de cor, o tamanho da janela de Gauss, e os níveis de comparação do operador de Canny. Idealmente não deveria ser necessário ao utilizador utilizar este modo de operação, mas de momento pode ainda ser necessário fazer algum ajuste na classificação de cor.

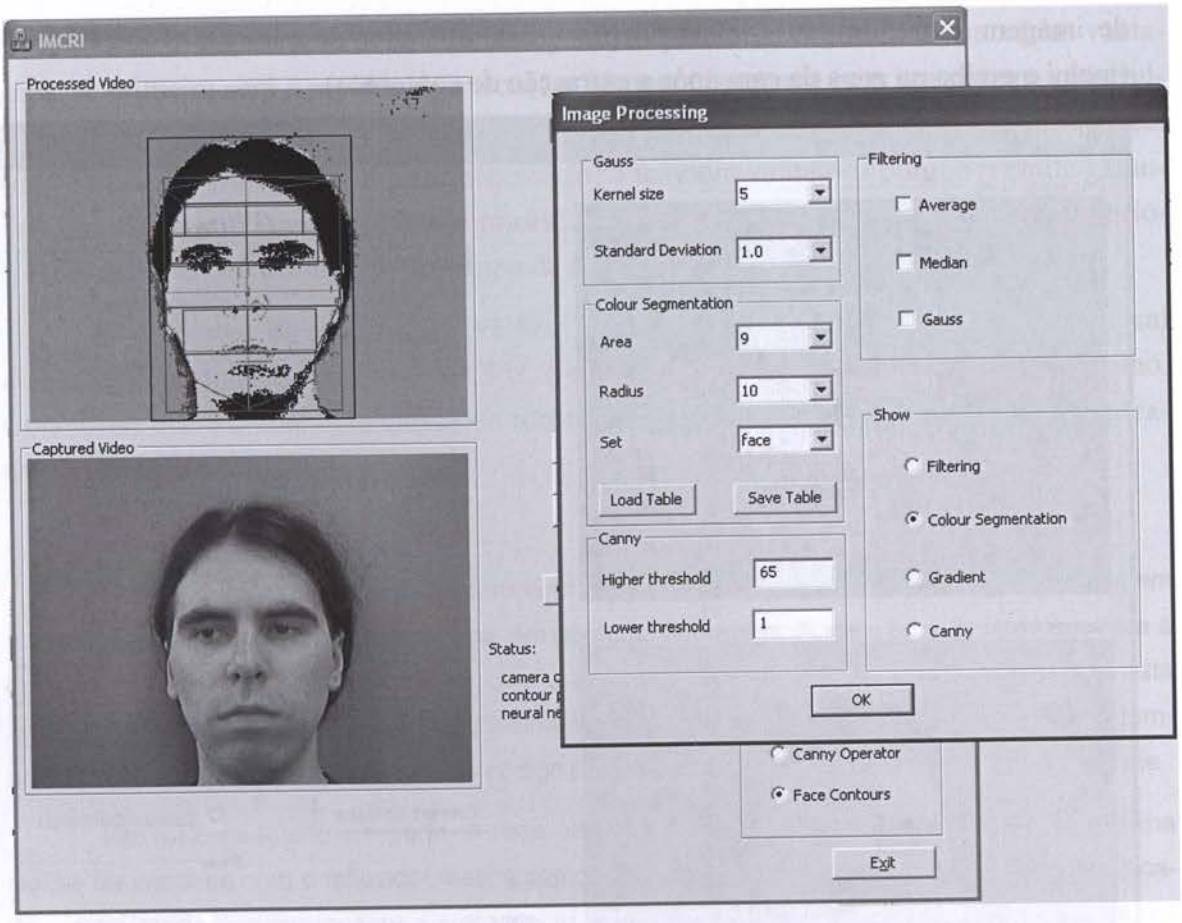


Figura 53 – Interface: modo de configuração do pré-processamento

Configuração da Identificação: neste modo de operação é possível seleccionar novos contornos de referência, assim como ver os que já estão guardados. Para criar um conjunto novo basta ir clicando no botão "Marcar", quando se entender que a expressão é a correcta. Uma vez carregados ou inicializados os contornos de referência é possível também criar um novo conjunto de treino da rede neuronal, treinar a rede ou carregar as definições anteriores. Para criar um novo conjunto de treino basta fazer a expressão de acordo com o que é mostrado automaticamente na caixa "Expressão" e clicar em "Gravar Entradas" quando a expressão é a correcta. No fim de treinar a rede é mostrado o erro quadrático médio final do treino. Tanto ao marcar os padrões de contornos como as entradas para rede neuronal existe um tempo morto durante o qual a imagem no momento do clique é mostrada.

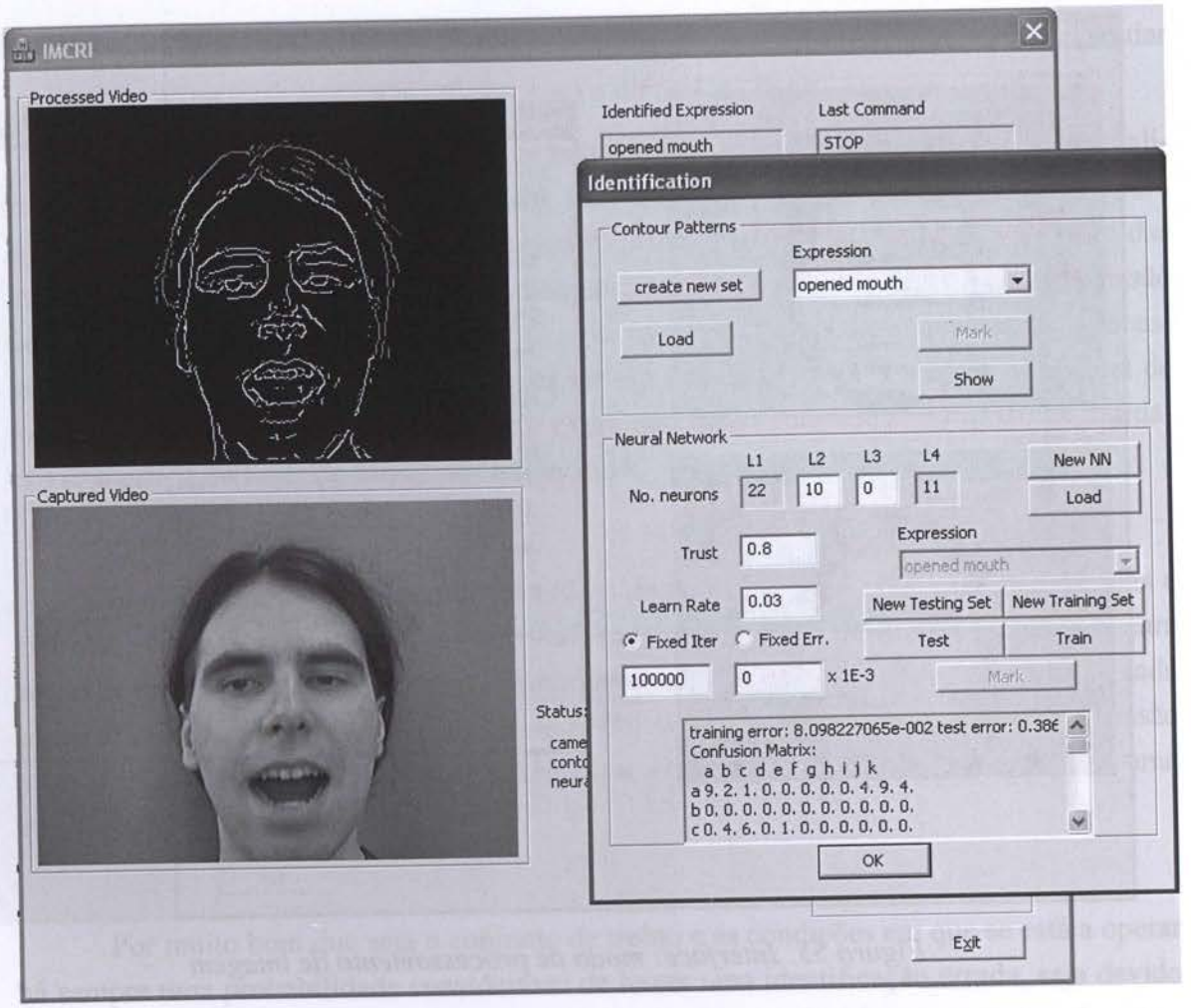


Figura 54 – Interface: modo configuração de identificação

Processamento de Imagem: aqui são apresentadas apenas algumas opções de processamento de imagem utilizados para o funcionamento da interface, a título de extra. Não tem influência no comando.

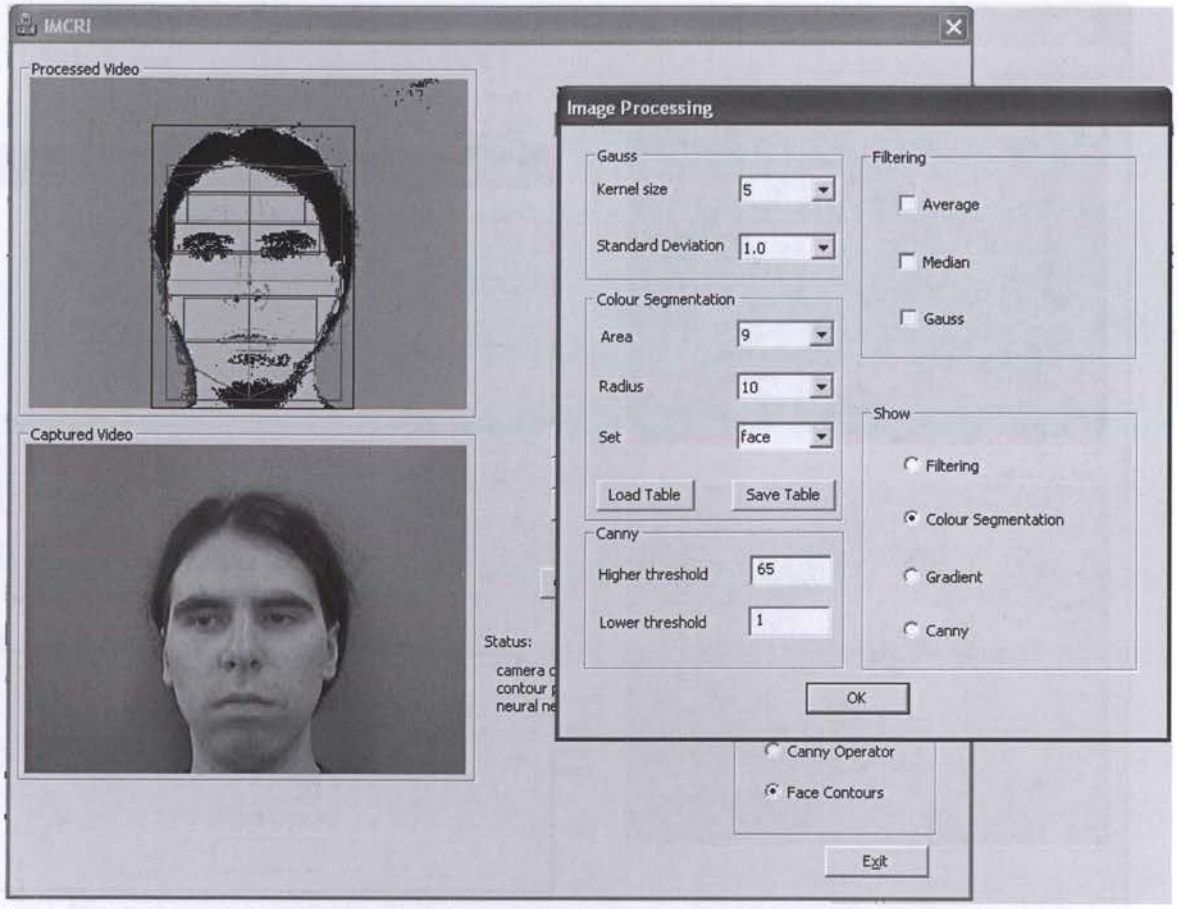


Figura 55: Interface: modo de processamento de imagem

Em qualquer momento é possível iniciar o sistema ou pará-lo e é disponibilizada também a hipótese de configurar o "driver" da câmara.

Os controlos do simulador também se encontram na interface, embora esta seja apenas um método de teste do sistema. O simulador pode ser ligado ou desligado completamente e pode funcionar no modo manual em que é possível introduzir a velocidade da cadeira e a posição, ou pode ser activado o modo "Simular" em que de facto simula o resultado dos comandos dados através da expressão facial.

5.2 Linguagem de Comando

A linguagem de comando é definida na interface pelo utilizador. Este pode associar uma expressão detectável, a um comando conforme lhe seja mais conveniente. Como não foram definidos os comandos a que a cadeira deve obedecer – isto depende sobretudo do

sistema de controlo – foram atribuídos 6 comandos possíveis: "Andar para frente", "Andar para trás", "Virar para esquerda", "Virar para a direita" e "Parar".

Existe um outro comando, que faz parte da interface em si, que inicia e pára a validação dos comandos: o comando "Recebe Instruções". Para dizer à cadeira de rodas para andar para frente, para trás, etc. , é necessário primeiro entrar no modo de validação das instruções. Uma vez dados os novos comandos, é conveniente sair repetindo a expressão facial associada a "Recebe Instruções". O objectivo deste protocolo é que não seja necessário uma preocupação constante com a expressão facial durante a condução da cadeira de rodas. Para isso basta salvaguardar que a expressão facial para entrar no modo de validação seja difícil de fazer de forma não intencional.

Implicitamente existe também a ordem de não fazer nada, que é dada por todas a expressões faciais não atribuídas. Foi verificado prático não utilizar a expressão "cara normal" para nenhum comando. É importante haver uma separação clara entre cada comando, de modo a evitar acções erradas. Garantir que na transição de uma expressão para outra não são feitas expressões intermédias associadas a comandos, proporciona uma utilização mais segura do sistema.

Por muito bom que seja o conjunto de treino e as condições em que se está a operar há sempre uma probabilidade considerável de haver uma identificação errada, seja devido a uma dedução errada (a rede neuronal não é infalível), seja devido a uma expressão intermédia não contemplada no conjunto de treino. Existe também a possibilidade do utilizador fazer um ou dois movimentos involuntários na cara sem querer quer sejam interpretados como comandos. De modo a minimizar estes erros definiu-se que é necessário para validar uma expressão, que esta esteja presente a 90% ou mais no registo de expressões identificadas no último segundo.

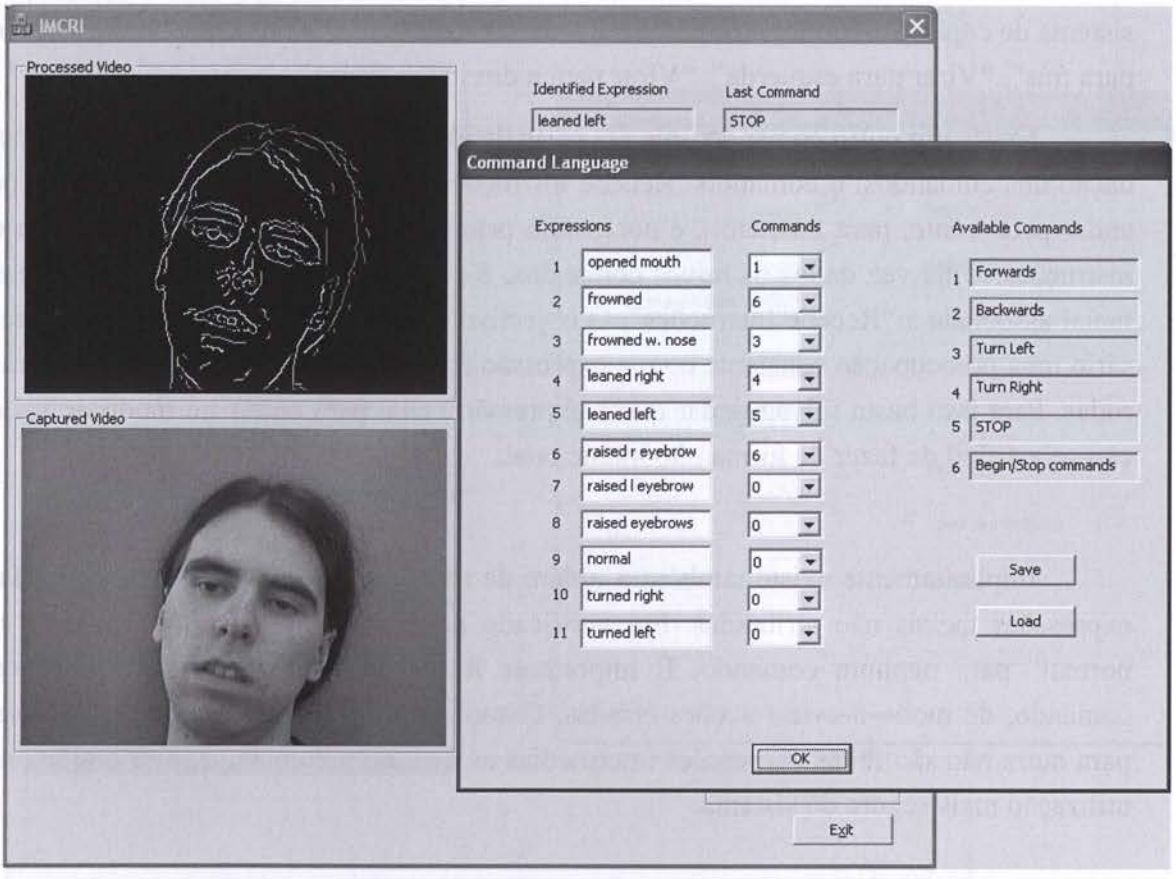


Figura 56 – Interface: modo de configuração da linguagem de comando

5.3 Simulador

O objectivo do simulador, é permitir verificar facilmente por simples observação, se o sistema está a funcionar ou não. Na prática não se espera que o utilizador esteja a olhar para a sua própria imagem, sempre que quer dar um comando. O natural é ir olhando em redor e ir fazendo as expressões abstraindo-se da existência da interface de comando. O melhor método de ensaio será portanto, ter uma representação do estado do mundo à medida que se comanda a cadeira virtual e ir observando o seu comportamento.

O simulador implementado consiste na representação tridimensional de uma cadeira de rodas num espaço aberto. A cadeira de rodas pode mover-se para a frente para trás e virar para a esquerda e para a direita. Foi introduzida alguma inércia, quer no movimento linear quer no angular de modo simular melhor o movimento real de uma cadeira de rodas.

A simulação introduz algum atraso no processamento, o que limita um pouco a avaliação dos resultados. Isto deve-se em parte a estar incluído no mesmo processo que a interface multimédia e o sistema operativo limitar processamento de cada processo individual.

http://paginas.fe.up.pt/~ee99041/IMCRI/PT

http://paginas.fe.up.pt/~ee99041/IMCRI/EN

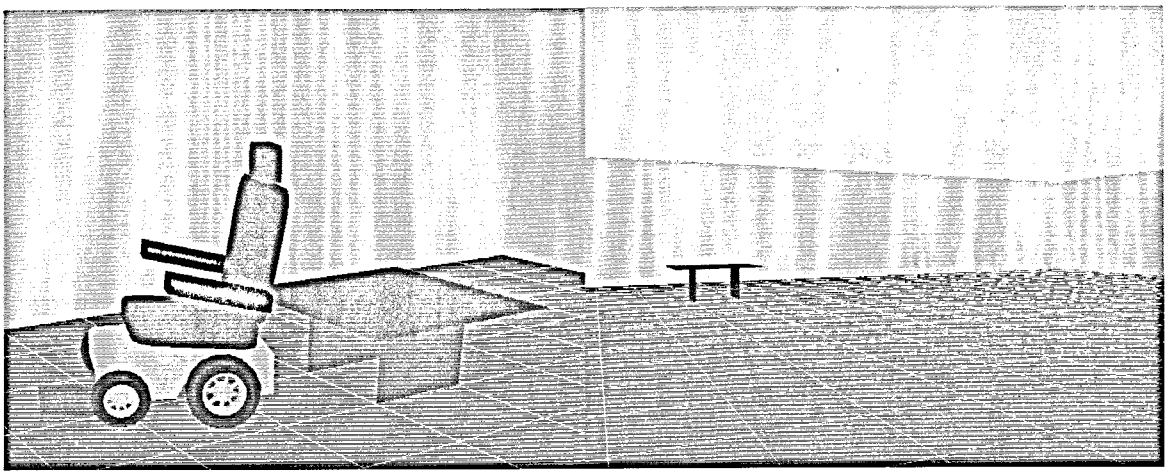


Figura 57 – Simulador gráfico(a-vista do centro;b-vista da cadeira)

Capítulo 6

6. Análise de Resultados

No geral os objectivos foram atingidos. Através dos resultados observados com o simulador, conclui-se que é possível comandar uma cadeira de rodas através da expressão facial. Contudo, os resultados não são perfeitos, sobram mesmo bastantes aspectos a melhorar.

O tempo de processamento deverá ser reduzido de modo a obter um melhor compromisso com a condição de tempo real.

A robustez não é a melhor, existem algumas más interpretações casuais durante a operação do sistema, que podem tornar o comando da cadeira de rodas algo impreciso, principalmente combinado com o elevado tempo de processamento. Basta uma expressão mal identificada, para a cadeira ficar 1 segundo sem um comando correcto. Alguns destes erros são devido à sensibilidade elevada à iluminação, que poderia ser compensada parcialmente se a informação a nível de contornos fosse melhor processada.

O simulador ainda não permite um teste rigoroso e sobretudo, objectivo ao sistema desenvolvido. Este ponto deve ser melhorado de maneira a minimizar as discrepâncias com o que se pode esperar na realidade.

Estima-se que os problemas encontrados possam ser minimizados ou eliminados com as soluções descritas no ponto 7.

Após 10 testes com cada expressão foi construída uma matriz de confusão que permite verificar a qualidade de identificação de cada expressão. Foram também feitos testes apenas com segmentação de cor e apenas com contornos de forma a ver o comportamento comparativamente ao funcionamento com segmentação de cor e contornos em simultâneo.

	boca aberta	franzida	franzida com nariz	Inclinada para esquerda	Inclinada para direita	normal	sobrancelha direita	sobrancelha esquerda	ambas sobrancelhas	virar para direita	virar para esquerda
boca aberta	9	0	0	0	0	0	0	0	0	0	0
franzida	0	3	0	0	0	0	0	0	0	0	0
franzida com nariz	0	0	8	0	0	0	0	0	0	0	0
inclinada para esquerda	0	0	0	10	0	0	0	0	0	0	0
inclinada para direita	0	0	0	0	10	0	0	0	0	0	0
normal	0	1	0	0	0	10	0	0	0	0	0
sobrancelha direita	0	0	0	0	0	0	3	0	0	0	0
sobrancelha esquerda	0	0	0	0	0	0	0	9	0	0	0
ambas sobrancelhas	0	0	0	0	0	0	1	0	10	0	0
virar para direita	0	0	0	0	0	0	0	0	0	10	0
virar para esquerda	0	0	0	0	0	0	0	0	0	0	8
não identificado	1	6	2	0	0	0	6	1	0	0	2

Figura 58 Matriz de confusão (ambos os métodos)

	a (nr de previsoes correctas de identificaçao negativa)	b (nr de previsoes incorrectas de identificaçao positiva)	c (nr de previsoes incorrectas de identificaçao negativa	d(nr de previsoes correctas de identificaçao positivas)
boca aberta	100	1	0	9
franzir	100	7	0	3
franzir com nariz	100	2	0	8
inclinar para direita	100	0	0	10
inclinar para esquerda	100	0	0	10
normal	99	0	1	10
sobrancelha direita	100	7	0	3
sobrancelha esquerda	100	1	0	9
ambas sobrancelhas	99	0	1	10
virar direita	100	0	0	10
virar esquerda	100	2	0	8

Figura 59 Medidas (ambos os métodos)

	Exactidão	Taxa de verdadeiros positivos	Taxa de falsos positivos	Taxa de verdadeiros negativos	Taxa de falsos negativos	Precisão
boca aberta	94,868	100,000	0,990	99,010	0,000	90,000
franzir	54,772	100,000	6,542	93,458	0,000	30,000
franzir com nariz	89,443	100,000	1,961	98,039	0,000	80,000
inclinar para direita	100,000	100,000	0,000	100,000	0,000	100,000
inclinar para esquerda	100,000	100,000	0,000	100,000	0,000	100,000
normal	95,346	90,909	0,000	100,000	9,091	100,000
sobrancelha direita	54,772	100,000	6,542	93,458	0,000	30,000
sobrencelha esquerda	94,868	100,000	0,990	99,010	0,000	90,000
ambas sobrancelhas	95,346	90,909	0,000	100,000	9,091	100,000
virar para direita	100,000	100,000	0,000	100,000	0,000	100,000
virar para esquerda	89,443	100,000	1,961	98,039	0,000	80,000
MEDIA	88,078	98,347	1,726	98,274	1,653	81,818

Figura 60 Resultados (ambos os métodos)

Como foi referido, foram também obtidas medidas apenas para os contornos, com resultados um pouco inferiores aos obtidos juntamente com segmentação de cor.

	boca aberta	franzida	franzida com nariz	Inclinada para esquerda	Inclinada para direita	normal	sobran-celha direita	sobrancel ha esquerda	ambas sobrancel has	virar para direita	virar para esquerda
boca aberta	**10**	0	0	0	0	0	0	0	0	0	0
franzida	0	**0**	0	0	0	0	0	0	0	0	0
franzida com nariz	0	**3**	**10**	0	0	0	**1**	0	0	**2**	0
inclinada para esquerda	0	0	0	**10**	0	0	0	0	0	0	0
inclinada para direita	0	0	0	0	**9**	0	0	0	0	0	0
normal	0	**2**	0	0	0	**8**	0	0	0	0	0
sobrancelha direita	0	0	0	0	0	0	**3**	0	0	0	0
sobrancelha esquerda	0	0	0	0	0	0	0	**10**	0	0	0
ambas sobrancelhas	0	0	0	0	0	0	0	0	**8**	0	0
virar para direita	0	0	0	0	0	0	0	0	0	**6**	0
virar para esquerda	0	0	0	0	0	0	0	0	0	0	**10**
não identificado	0	**5**	0	0	**1**	**2**	**6**	0	**2**	**2**	0

Figura 61 Matriz de confusão (contornos)

	a (nr de previsoes correctas de identificaçao negativa)	b (nr de previsoes incorrectas de identificaçao positiva)	c (nr de previsoes incorrectas de identificaçao negativa	d(nr de previsoes correctas de identificaçao positivas)
boca aberta	100	0	0	10
franzir	100	10	0	0
franzir com nariz	94	0	6	10
inclinar para direita	100	0	0	10
inclinar para esquerda	100	1	0	9
normal	98	2	2	8
sobrancelha direita	100	7	0	3
sobrancelha esquerda	100	0	0	10
ambas sobrancelhas	100	2	0	8
virar direita	100	4	0	6
virar esquerda	100	0	0	10

Figura 62 Medidas (contornos)

	Exactidão	Taxa de verdadeiros positivos	Taxa de falsos positivos	Taxa de verdadeiros negativos	Taxa de falsos negativos	Precisão
boca aberta	100,0	100,0	0,0	100,0	0,0	100,0
franzir	0,0	0,0	9,1	90,9	0,0	0,0
franzir com nariz	79,1	62,5	0,0	100,0	37,5	100,0
inclinar para direita	100,0	100,0	0,0	100,0	0,0	100,0
inclinar para esquerda	94,9	100,0	1,0	99,0	0,0	90,0
normal	80,0	80,0	2,0	98,0	20,0	80,0
sobrancelha direita	54,8	100,0	6,5	93,5	0,0	30,0
sobrencelha esquerda	100,0	100,0	0,0	100,0	0,0	100,0
ambas sobrancelhas	89,4	100,0	2,0	98,0	0,0	80,0
virar para direita	77,5	100,0	3,8	96,2	0,0	60,0
virar para esquerda	100,0	100,0	0,0	100,0	0,0	100,0
MEDIA	79,5999921	85,6818182	2,2209093	97,7790907	5,2272727	76,3636364

Figura 63 Resultados (contornos)

De igual forma, também foram feitos testes apenas para segmentação de cor, que à semelhança do que aconteceu com os testes feitos apenas com contornos, obtiveram também resultados inferiores quando comparados aos obtidos com ambos os métodos.

	boca aberta	franzida	franzida com nariz	Inclinada para esquerda	Inclinada para direita	normal	sobrance-lha direita	sobrance-lha esquerda	ambas sobrance-lhas	virar para direita	virar para esquerda
boca aberta	**9**	0	0	0	0	0	0	0	0	0	0
franzida	0	**3**	0	0	0	0	0	0	0	0	0
franzida com nariz	0	0	**8**	0	0	0	0	0	0	0	0
inclinada para esquerda	0	0	0	**10**	0	0	0	0	0	0	0
inclinada para direita	0	0	0	0	**10**	0	0	0	0	0	0
normal	0	**1**	0	0	0	**10**	0	0	0	0	0
sobrancelha direita	0	0	0	0	0	0	**3**	0	0	0	0
sobrancelha esquerda	0	0	0	0	0	0	0	**9**	0	0	0
ambas sobrancelhas	0	0	0	0	0	0	**1**	0	**10**	0	0
virar para direita	0	0	0	0	0	0	0	0	0	**10**	**0**
virar para esquerda	0	0	0	0	0	0	0	0	0	0	**8**
não identificado	**1**	**6**	**2**	0	0	0	**6**	**1**	0	**0**	**2**

Figura 64 Matriz de confusão (segmentação de cor)

	a (nr de previsoes correctas de identificaçao negativa)	b (nr de previsoes incorrectas de identificaçao positiva)	c (nr de previsoes incorrectas de identificaçao negativa	d(nr de previsoes correctas de identificaçao positivas)
boca aberta	100	1	0	9
franzir	100	7	0	3
franzir com nariz	100	2	0	8
inclinar para direita	100	0	0	10
inclinar para esquerda	100	0	0	10
normal	99	0	1	10
sobrancelha direita	100	7	0	3
sobrancelha esquerda	100	1	0	9
ambas sobrancelhas	99	0	1	10
virar direita	100	0	0	10
virar esquerda	100	2	0	8

Figura 65 Medidas (segmentação de cor)

	Exactidão	Taxa de verdadeiros positivos	Taxa de falsos positivos	Taxa de verdadeiros negativos	Taxa de falsos negativos	Precisão
boca aberta	94,9	100,0	1,0	99,0	0,0	90,0
franzir	64,5	83,3	4,8	95,2	16,7	50,0
franzir com nariz	83,7	100,0	2,9	97,1	0,0	70,0
inclinar para direita	100,0	100,0	0,0	100,0	0,0	100,0
inclinar para esquerda	100,0	100,0	0,0	100,0	0,0	100,0
normal	78,3	87,5	2,9	97,1	12,5	70,0
sobrancelha direita	54,8	100,0	6,5	93,5	0,0	30,0
sobrencelha esquerda	47,8	57,1	5,8	94,2	42,9	40,0
ambas sobrancelhas	84,5	71,4	0,0	100,0	28,6	100,0
virar para direita	100,0	100,0	0,0	100,0	0,0	100,0
virar para esquerda	100,0	100,0	0,0	100,0	0,0	100,0
MEDIA	82,6	90,9	2,2	97,8	9,1	77,3

Figura 66 Resultados (segmentação de cor)

Capítulo 7

7. Perspectivas de Desenvolvimento

7.1 Correcção de pontos de brilho/diferente iluminação

As diferenças nas condições de iluminação podem afectar a identificação de caras e também de características. Assim de forma a eliminar este efeito a nível local (zonas particulares de maior ou menor iluminação/brilho), poderá ser aplicado o conceito de *Run mínimo*, decorrente da compactação da imagem com RLE (*Run Length Encoding*), codificação que no entanto não vamos utilizar de facto. Resumidamente o RLE, transforma uma imagem segmentada em *Runs*, que são elementos seguidos pertencentes ao mesmo conjunto. Assim deixamos de ter os *pixels* representados e passamos a ter representado tamanhos de *runs* e categoria a qual pertencem os pontos de cada *run*.

Aplicando o conceito de *Run mínimo* temos que nenhuma *run* pode ter tamanho inferior ao definido como mínimo. Caso seja inferior, essa *run* será transformada/absorvida por outras *runs*, como se verifica nos exemplos seguintes (para *Run mínimo=5*):

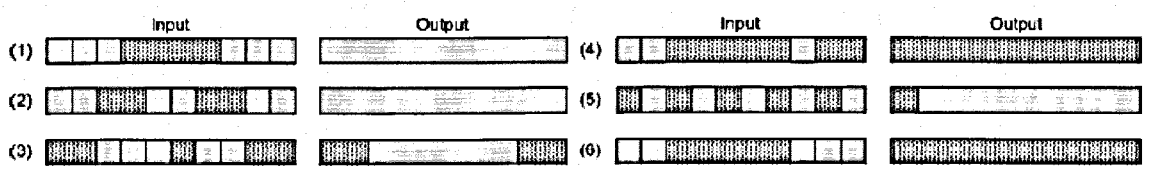


Figura 67 - Codificação RLE com run mínimo (5)

Desta forma esperamos eliminar efeitos localizados de diferença de luminosidade.

7.2 Actualização automática da tabela de *Look Up*

A tabela de *Look Up* é o elemento crítico da segmentação, muita da fiabilidade da segmentação de cor depende da boa calibração da tabela em relação ao indivíduo e ambiente. Uma tabela construída numa dada altura, para um dado utilizador num ambiente, dificilmente obterá os resultados desejados se alguma destas três componentes for variada.

Assim, uma boa maneira do sistema contornar este problema é, a partir de uma tabela pré-construida com valores típicos (o mais abrangentes possível) e da localização da cara dada pela segmentação, extrair os pontos que fiquem perto, ou entre pontos classificados como sendo da cara e se estiverem a menos de uma dada distancia do conjunto "cara" (no cubo RGB) passem a ser classificados como cara também. A partir desta nova classificação e usando o mesmo principio podemos actualizar dinamicamente os restantes conjuntos.

Numa fase mais avançada, poderiam até todos os subconjuntos serem criados pelo programa mediante uma referência, como a cor de fundo (teria de ser fixa) ou a cor genérica da cara.

7.3 Enquadramento automático da cara

Um dos objectivos do trabalho, passava por utilizar movimentos da camera para automaticamente enquadrar a cara do utilizador na imagem, de forma a melhor obter as imagens.

Este ponto foi deixado para o final do projecto, e não houve tempo para o concluir como desejado. Assim sendo a camera esta a utilizar o software do fabricante para seguir a cara. Este software não produz os resultados desejados, pelo que um ponto importante a desenvolver de seguida seria controlar os movimentos da camera.

O método utilizado para controlar os movimentos da camera seria estabelecer dois *treashholds* para cada direcção, de forma a que se o centro de massa passasse o treashhold máximo a camera seguiria essa direcção, e pararia se o centro de massa passasse o treashhold mínimo (a menos que fosse limitada pela sua excursão máxima).

Na figura seguinte exemplifica-se uma distribuição de treashholds.

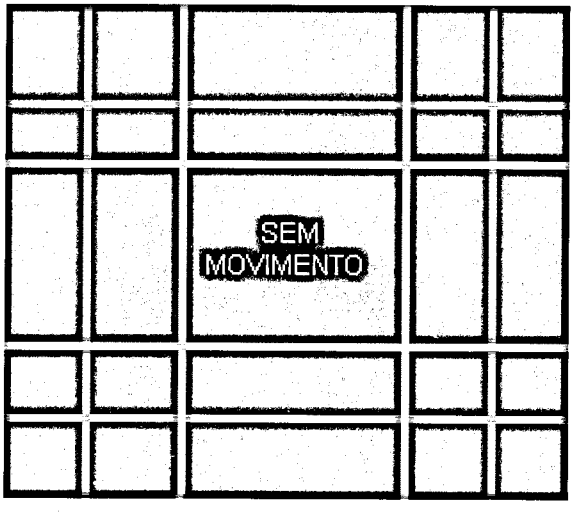


Figura 68 - Limite mínimo; Limite máximo

Este método permite evitar instabilidade no caso de o centro de massa estar a variar em torno de um limite (caso fosse único).

http://paginas.fe.up.pt/~ee99041/IMCRI/PT

http://paginas.fe.up.pt/~ee99041/IMCRI/EN

No caso da camera "Logitech QuickCam Sphere" disponibilizada, também é posivel controlar o zoom, até 3 vezes (digital). Assim seria também interessante controlar o zoom de forma automática, para tal usar-se-ia um coeficiente de ocupação de imagem por parte da cara, que seria calculado com a equação:

$$Coeficiente = \frac{n^{o}\ pixels_{ClasseCARA}}{n^{o}\ total\_pixels}$$

Seria de seguida estabelecido um valor típico para este coeficiente (resultante dum enquadramento considerado ideal). O programa tentaria alterar o zoom de forma a manter o coeficiente tão próximo do típico como possível. Caso o coeficiente fosse demasiado baixo aumentaria o zoom, caso o coeficiente fosse demasiado alto diminuiria o zoom. Tal como foi feito para os movimentos, seriam também estabelecidos treashholds mínimo e máximo, de forma a não haver instabilidade em torno do valor do coeficiente típico.

7.4 Extracção de Informação dos Contornos

Como já foi dito o processamento que reúne a informação da imagem de contornos e fornece as medidas ao bloco de identificação, ou é preciso e lento ou impreciso e rápido. Convém por isso solucionar este problema. No seguimento do trabalho desenvolvido, estima-se que uma boa solução passaria por substituir o cálculo de distâncias, por simples diferenças de imagens, mantendo o anterior processamento de redimensionamento e soma de imagens de referência.

A diferença de imagens aqui proposta trata-se de encontrar pontos não comuns. As medidas de saída seriam a contagem de pontos não comuns em vez da média de distâncias, e, na mesma, as coordenadas do centro de massa que indicariam onde se localiza a maior concentração de pontos não comuns. Tendo em conta que esta operação é efectuada apenas sobre os contornos da cara à mesma escala e que o padrão de referência contém vários exemplos diferentes da mesma expressão, é de prever que as medidas tiradas tenham uma forte relação com a semelhança entre eles. Poderá, no entanto, haver algum problema ao redimensionar imagens de tamanhos muito diferentes, mas isto não deve acontecer muitas vezes tendo em conta que a distância entre a câmara e a cara do utilizador não deve variar muito.

A solução proposta melhoraria o tempo de processamento dado que em vez de ser necessário pesquisar uma distância mínima para cada ponto da imagem, apenas é necessário percorrer cada imagem uma vez verificando o estado de cada ponto.

Quanto à precisão, prevê-se que ao melhorar o tempo de processamento, seja possível melhorar a resolução da grelha ou até dispensá-la e considerar todos os pontos dos contornos da cara. Dado que todos são relevantes para caracterizar uma expressão facial, a precisão na identificação deve aumentar.

7.5 Interface com o sistema de controlo

A nível da interface com o sistema de controlo da cadeira, é importante garantir ao máximo a modularidade de cada sistema. A implementação executada peca por ter os comandos pré-definidos: andar em frente, andar para trás, etc.. Para além da hipotese de escolher a expressão que acciona cada comando, seria conveniente que os comandos fossem também configuráveis. Desse modo seria possível ter um sistema ao sistema de controlo da cadeira de rodas evoluir sem que a interface de comando necessitasse de ser substituída.

Outro aspecto fundamental, será estudar o modo como os sistemas de controlo de cadeiras de rodas existentes no mercado, fazem a ligação ao sistema de comando, e implementar a ponte entre o sistema desenvolvido e os sistemas de controlo reais.

7.6 Simulador:

O melhoramento do sistema de teste sem recorrer a uma cadeira de rodas real, passa necessariamente por melhorar o simulador. Mais importante do que melhorar o aspecto gráfico, que é bastante simples, será fazer algum estudo do comportamento dinâmico de cadeiras eléctricas reais e transpor essas características para o simulador.

Uma vez melhorado o realismo do simulador seria útil introduzir uma prova de perícia do tipo: seguir uma pista sem ultrapassar certos limites e fazê-lo no mínimo tempo possível. Este tipo de prova simulada permitiria obter dados objectivos sobre a "performance" do sistema composto pela cadeira, condutor e sistema de controlo. Seria, também,

importante isolar apenas a prestação do sistema de comando, ou sistema de comando e utilizador. Isso pode ser obtido facilmente se se comparar cada simulação utilizando o sistema de comando multimédia com uma simulação utilizando, por exemplo, o teclado para comandar a cadeira virtual. Deste modo poder-se-ia obter estatísticas do tipo: tempo que demorou a fazer o percurso e número de vezes que saiu fora da pista; que reflectem directa e objectivamente o comportamento da interface multimédia.

7.7 Configuração

Depois de configurar o sistema para um determinado utilizador, essas configurações ficam registadas em vários ficheiros e é possível carregá-las em utilizações posteriores. Contudo, a actual versão dos sistema apenas guarda uma configuração. Seria útil ter a hipótese de salvar e carregar várias configurações distintas, sem ter de recorrer à cópia de ficheiros, tendo em vista que uma cadeira de rodas poderá ser utilizada por várias pessoas distintas (por exemplo em hospitais).

http://paginas.fe.up.pt/~ee99041/IMCRI/PT

http://paginas.fe.up.pt/~ee99041/IMCRI/EN

Capítulo 8

8. Referências Bibliográficas

1. Áudio-Visual Person Recognition: A Survey ; *Kjersti Aas,* 1996
2. Agent-Based Recognition of Facial Expressions; *Pablo Suau et al*, 2001
3. Facial expression recognition from video sequences: temporal and static modelling; *Ira Cohen et al*, 2003
4. Facial Expression Recognition- A Brief Tutorial Overview; *Claud C. et al*
5. Recognition of Facial Expressions in the presence of Occlusion; *Fabrice Beurel et al*
6. Real Time Face detection and facial expression recognition: Development and Applications to Human Computer Interaction; *Marian Stewart Bartlett et al*
7. Coding Facial Expressions with Gabor Wavelets; *Michael Lyons et al*
8. Real-time shape estimation for continuum robots using visin; *M.W. Haman et al*, 2004
9. Real-time machine Vision Perception and Prediction; *James Bruce et al*, 2000
10. Facial Expression Recognition Using Pseudo 3D Hidden Markov Models; *Stefan Muller et al*
11. Neuropsychosocial Factors in Emotion Recognition: Facial Expressions; www.neuropsychologycentral.com, 2002
12. Automatic Facial expression recognition using facial animation parameters and multi-stream HMMs; *Peter S. Alekic et al*, 2005
13. Semi- Supervised Learning for Facial Expression Recognition; *Ira Cohen et al*
14. Facial Expression Recognition using Support Vector Machines; *Phillipp Michel et al*
15. Real world, real-time, Automatic Recognition of facial expressions; *Ying-li Tian et al*, 2002
16. Facial Expression decomposition; *Hougcheng Wang et al*
17. Automatic Pixel Selection for Optimizing Facial Expression Recognition using Eigenfaces; *Carmen Frank et al*

18. A Neural Network Facial Expression Recognition System using Unsupervised Local Processing; *Leonardo Franco*, 2001

19. Facial Expression recognition using a Dynamic Model and Motion Energy; *Ifan Esa et al*

20. Facial expression recognition from video sequences; *Ira Cohen et al*, 2002

21. Digital Image Processing; *Rafael C. Gonzalez et al*

22. Fundamentals of Digital Image Processing; *Anil K. Jain*

12 Lip Corner Puller *Zygomaticus major* 	**13** Cheek Puffer *Levator anguli oris (a.k.a. Caninus)* 	**14** Dimpler *Buccinator* 
15 Lip Corner Depressor *Depressor anguli oris (a.k.a. Triangularis)* 	16 Lower Lip Depressor *Depressor labii inferioris* 	**17** Chin Raiser *Mentalis* 
18 Lip Puckerer *Incisivii labii superioris and Incisivii labii inferioris* 	**20** Lip stretcher *Risorius w/ platysma* 	**22** Lip Funneler *Orbicularis oris* 
23 Lip Tightener *Orbicularis oris* 	**24** Lip Pressor *Orbicularis oris* 	**26** Jaw Drop *Masseter, relaxed Temporalis and internal Pterygoid* 

http://paginas.fe.up.pt/~ee99041/IMCRI/PT

http://paginas.fe.up.pt/~ee99041/IMCRI/EN

27 Mouth Stretch *Pterygoids, Digastric* 	**28** Lip Suck *Orbicularis oris* 	**41** Lid droop *Relaxation of Levator palpebrae superioris* 
42 Slit*Orbicularis oculi* 	**44** Squint *Orbicularis oculi, pars palpebralis* 	**51** Head turn left 
52 Head turn right 	**53** Head up 	**54** Head down 
55 Head tilt left 	**56**Head tilt right 	**57** Head forward 
58 Head back 	**61** Eyes turn left 	**62** Eyes turn right 
63 Eyes up 	**64** Eyes down 	

9.2 Medidas.txt

-----Centroide-----

Zona-3

Massa-11398

Centro(x,y)-(165,108)

-----ContornoCara-----

Limite direito-226

Limite esquerdo-107

Limite inferior-10

Limite superior-186

-----ContaPontos-----

-----Sobrancelha Esquerda-----

Zona 0-0

Centro 0 (x,y)- (-1,-1)

Zona 1-0

Zona 3-913

Zona 4-0

Centro 4 (x,y)- (-1,-1)

Zona 5-27

Centro 5 (x,y)- (149,146)

-----Sobrancelha Direita-----

Zona 0-32

Centro 0 (x,y)- (189,145)

Zona 1-0

Zona 3-630

Zona 4-162

Centro 4 (x,y)- (194,147)

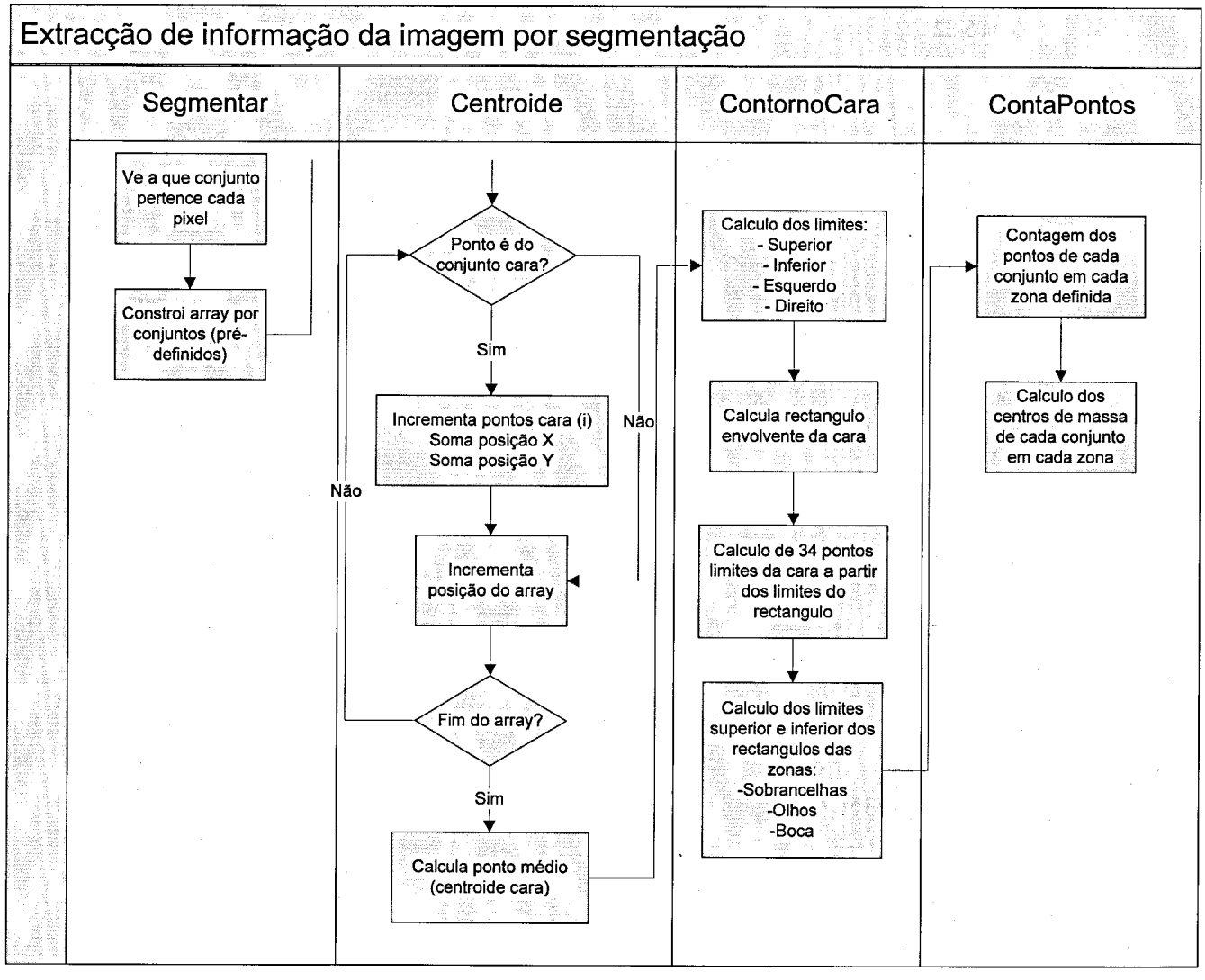
Zona 5-36

Centro 5 (x,y)- (175,147)

-----Olho Esquerda-----

Zona 0-127
Centro 0 (x,y)- (139,133)
Zona 1-0
Zona 3-446
Zona 4-405
Centro 4 (x,y)- (140,135)
Zona 5-22
Centro 5 (x,y)- (153,134)
-----Olho Direita-----
Zona 0-92
Centro 0 (x,y)- (188,134)
Zona 1-0
Zona 3-619
Zona 4-256
Centro 4 (x,y)- (186,135)
Zona 5-53
Centro 5 (x,y)- (179,133)
-----Boca Esquerda-----
Zona 0-204
Centro 0 (x,y)- (154,70)
Zona 1-0
Zona 3-702
Zona 4-138
Centro 4 (x,y)- (146,70)
Zona 5-251
Centro 5 (x,y)- (151,76)
-----Boca Direita-----
Zona 0-238
Centro 0 (x,y)- (175,71)
Zona 1-0
Zona 3-853
Zona 4-116
Centro 4 (x,y)- (188,75)
Zona 5-228
Centro 5 (x,y)- (179,72)

9.3 Fluxograma (Segmentação)



http://paginas.fe.up.pt/~ee99041/IMCRI/PT

http://paginas.fe.up.pt/~ee99041/IMCRI/EN

9.4 Retropopagação

O algoritmo divide-se em duas fases o cálculo da saída para um determinado padrão de entrada e o cálculo dos gradientes da superfície de erro em relação aos pesos na entrada de cada neurónio. Em pormenor o que é feito é:

Inicializar os pesos dos arcos da rede: são atribuídos os pesos de acordo com uma distribuição uniforme de média 0 (ou 0.5 conforme o ponto médio da função de activação).

Introduzir um padrão de treino nas entradas

Calcular a saída para o padrão de entrada apresentado e o erro: para cada neurónio $v_i[n] = w_0 + \sum_{i=o}^{m} w_i u_i$, em que i representa o índice do neurónio,w os pesos, u os valores no neurónio, ϕ a função de activação $y_i = \phi(v_i[n])$. O gradiente do erro é dado por:

para os neurónios de saída: $\delta_o = (c_o - u_o)u_o(1-u_o)$, com c a representar a resposta certa

para os neurónios escondidos: $\delta_i = \left(\sum_{i=o}^{m} w_{ji}\delta_o\right)u_i(1-u_i)$

Actualização dos pesos: $w^*_{i,j} = w_{i,j} + \rho\delta_i u_i$, em que ρ representa o índice de aprendizagem.

Isto é repetido até que o erro quadrático médio, $eqm = 0.5\sum\left(y_{esperado} - y\right)^2$, seja inferior a um determinado valor pretendido ou até que tenha atingido o mínimo. Nos ensaios feitos foram efectuadas 10000 iterações ao fim das quais o erro era em geral baixo: <0.02.

9.5 Estrutura do programa

Toda a interface foi implementada recorrendo à linguagem de programação C++. O ambiente de desenvolvimento utilizado foi o Visual Studio 6.0. Esta ferramenta inclui para além do compilador de C++, um biblioteca de classes já feitas: Microsoft Foundation Classes. A MFC permite utilizar as chamadas ao sistema (API Win 32) através de um programa em C++, num nível de abstracção mais elevado.

A API Win 32 é um conjunto de chamadas ao sistema, que permite programar a interface gráfica com o utilizador, nomeadamente a criação de janelas e passar as imagens da câmara, de forma simples.

O facto de se recorrer ao sistema operativo para criar a interface tem várias vantagens. A primeira será a rapidez de execução, comparativamente com um sistema construído em cima de uma máquina virtual. Quando há necessidade de utilizar tempo real é também vantajoso, porque permite garantir a consistência do thread de maior prioridade e do handler de interrupção de maior prioridade, assim como a alteração do tempo de processamento de cada thread. O problema de recorrer à API Win32 é que o funcionamento do executável vai depender do sistema operativo e da CPU, o que não é muito importante para este trabalho.

Conclui-se que é claramente vantajoso recorrer directamente às funções do sistema operativo neste caso.

Salienta-se ainda que as imagens devem ser guardadas em vectores unidimensionais (v[x*largura+y]) e não vectores encadeados do tipo v[x][y] quando se pretende tempos de processamento reduzidos. Isto é válido para o compilador utilizado pelo Visual Studio, é possível que com outras arquitecturas de processador(es) e compiladores diferentes, isto deixe de ser verdade. A vantagem de utilizar vectores bidimensionais, no que toca a tempo de execução, é apenas não ser necessário fazer as contas que permitem indexar a posição pretendida, no caso unidimensional.

Classes importantes.

http://paginas.fe.up.pt/~ee99041/IMCRI/PT

http://paginas.fe.up.pt/~ee99041/IMCRI/EN

Interface:

CImcriDlg: contém toda a interface com o utilizador assim como a captura de imagens provenientes da câmara digital. Contém também a rede neuronal e a tabela de transferência entre expressões e comandos, embora estas funções devessem estar em classes distintas.

CCameraDlg: contém a janela onde é mostrada a imagem processada.

CCanvDialog: contém a janela onde é mostrado o resultado da simulação.

Processamento de imagem e simulador

CMyBitmap: esta classe contém a imagem capturada (em RGB) sob a forma de um vector unidimensional, mas que pode ser endereçado em x,y facilmente: Imagem[y*largura-(largura+1-y)]. Todas as funções de processamento de imagem estão contidas nesta classe, ainda que algumas utilizem objectos de outras classes.

Por exemplo: Convolucao(CJanela), Segmentar(CTabela), Derivada(CJanela, CJanela)

CSimulador: aqui encontra-se implementado o simulador. Nesta classe são guardados o estado presente da cadeira de rodas virtual (x,y,□), a sua velocidade linear e angular e as suas velocidades no estado anterior. A actualização do estado da cadeira é feita através da função ActualizaEstado(double v_ref,double v_angular_ref), e é também nesta classe que está implementado o "motor gráfico" do simulador, através de DesenhaCadeira().

- foram também criadas outras classes destinadas a serem utilizadas pelas classes principais, por exemplo:

CTabela – destinada ao processamento da "look up table".

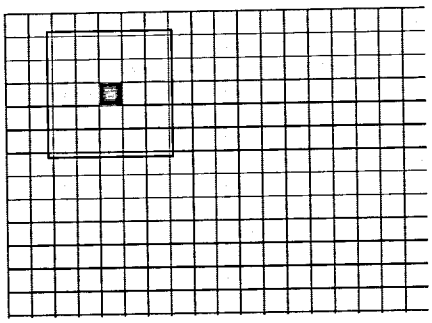
CHausdorffTransform – destinada ao cálculo de distâncias entre os pontos de dois padrões

CJanela – dedicada a criar e guardar a janela de convolução.

9.6 Convolução com uma Janela

Uma imagem é na verdade um conjunto bidimensional de pontos igualmente espaçados. Para fazer a convolução por uma janela de convolução selecciona-se um pixel, e por cima desse pixel é "encaixada" a janela de convolução. A janela de convolução não é mais do que a função pela qual se vai fazer a convolução limitada em x e em y, com o mesmo intervalo de amostragem.

Uma vez sobreposta a janela de convolução, multiplica-se o próprio peixel e os pixels envolventes pelo valor correspondente na janela. Somam-se todos os resultados da multiplicação e divide-se pelo somatório dos valores da janela. Por fim substitui-se o valor do pixel pelo resultado. Isto é feito para cada ponto da imagem.



$j_a(x,y)$ pode ser uma função qualquer linear ou não, com $-\frac{N}{2} \le \frac{x}{N}, \frac{y}{N} \le \frac{N}{2}$

N - dimensão da janela.

$(i,j) = \left(\frac{x}{N}, \frac{y}{N}\right)$, i e j são inteiros

$$imagem_{filtrada}(i_0 + i, j_0 + j) = \sum_{i=-N/2}^{N/2} \sum_{j=-N/2}^{N/2} \frac{j_a(i,j) \times imagem(i_0 + i, j_0 + j)}{\sum_{-N/2}^{N/2} \sum_{-N/2}^{N/2} j_a(i,j)}$$

(i_0, j_0) é o ponto central. Esta operação repete-se desde (i_0, j_0) até $(i_{largura}, j_{altura})$, passando por todos os pontos.

9.7 Janela de Gauss

$$j_a(x,y) = \sqrt{2\pi}\sigma \cdot A \cdot e^{\frac{2\pi^2 \cdot (x^2+y^2)}{N^2 \cdot \sigma^2}} \quad A = \left[\sqrt{2\pi}\sigma \cdot e^{-\frac{\pi}{\sigma^2}}\right]^{-1}, \text{ com } -\frac{N}{2} \leq \left(\frac{x}{N}, \frac{y}{N}\right) \leq \frac{N}{2}$$

N - dimensão da janela. $(i,j) = \left(\frac{x}{N}, \frac{y}{N}\right)$, i e j são inteiros

		6	8	10	8	6	
		8	16	19	16	8	
		10	19	23	19	10	
		8	16	19	16	8	
		6	8	10	8	6	

Aumentando o desvio padrão obtém-se uma imagem mais suave (menor frequência de corte), e diminuindo o desvio padrão obtém-se uma imagem mais próxima da original (maior frequência de corte).

Aumentar o tamanho à janela proporciona uma operação mais próxima do ideal.

9.8 Video

A captura de video é feita recorrendo, às funções e macros de captura de video disponíveis no Visual Studio 6.0.

A interface apresenta duas janelas com video. Uma delas apresenta sempre o video não processado e a outra o video com processamento de imagem. Foram utilizados dois métodos diferentes na obtenção destas imagens. O video não processado é obtido recorrendo somente às funções de captura de video:

```
m_hwndVideo=capCreateCaptureWindow(lpszWindowName,WS_CHILD|WS_VISIBLE,10,
290,320,240,GetSafeHwnd(),1);
capDriverConnect(m_hwndVideo, 0);
capDlgVideoSource(m_hwndVideo);
SetTimer(ID_CLOCK_TIMER, 50, NULL);
```

A cada 50ms mais o tempo de processamento da imagem, é mostrada e processada uma nova imagem.

A captura da imagem para processamento é feita utilizando o seguinte código na rotina de interrupção:

```
capGrabFrame(m_hwndVideo); // macro que amostra um frame da câmara
capEditCopy(m_hwndVideo);  // copia a imagem para o clipboard
OpenClipboard();    //like virtual memory.
 //m_hBmp é um "handle" de um Bitmap.
m_hFrmBmp = (HBITMAP)::GetClipboardData(CF_BITMAP);
CloseClipboard();
```

Este método permite obter um objecto do tipo CBitmap, que depois pode ser passado às classes de processamento de imagem. No entanto este método apresenta o inconveniente de utilizar o clipboard. Enquanto o programa está a ser executado é impossível fazer "copy, paste" ou "print screen". Convém melhorar este aspecto em implementações futuras.

É também possível chamar a janela de configuração do "driver" da câmara através de:

```
capDlgVideoSource(m_hwndVideo);
```

De referir ainda que as imagens utilizadas apresentam uma resolução de 320 por 240 a 24 bits de cor, ou seja 1 byte por cada canal R,G e B.