

Faculdade de Ciências e Faculdade de Engenharia da Universidade do Porto

COMANDO DE MOVIMENTO A PARTIR DA ANÁLISE DE GESTOS SIMBÓLICOS

Ricardo Sérgio Caetano Ferreira

Licenciado em Engenharia Electrotécnica e de Computadores, ramo de
Telecomunicações, pela Faculdade de Engenharia da Universidade do Porto

Dissertação provisória submetida para efeitos de atribuição do grau de Mestre em
Métodos Computacionais em Ciências e Engenharia

Sob a orientação científica do Doutor Armando Jorge Monteiro Neves Padilha,
Professor Associado da Faculdade de Engenharia da Universidade do Porto

Porto, Outubro de 2004

Resumo

O aumento das necessidades de comunicação entre o Homem e as máquinas tem revelado que nem sempre as interfaces Homem-Máquina existentes são as mais adequadas. Se os humanos no seu dia a dia comunicam entre si por fala e expressão corporal então toda a comunicação assente nestes princípios tenderá a ser natural, intuitiva e amigável para os humanos. Por isso tem havido, e continuará a haver, uma procura de novas interfaces que aproximem a comunicação Homem-Máquina da forma como os humanos comunicam entre si, de que este trabalho é um exemplo.

A presente dissertação relata então o trabalho desenvolvido na criação e análise de uma interface Homem-Máquina totalmente “mãos-livres” constituída por um sistema de reconhecimento de cinco gestos simbólicos, suficientes para comandar o movimento de um objecto num plano, por uma abordagem de visão por computador.

Foi também objectivo deste trabalho a verificação da qualidade dos resultados que a técnica da Análise de Componentes Independentes (ICA) consegue dar. Chegou-se à conclusão que a Arquitectura I do ICA, secundada por um classificador do vizinho mais próximo sustentado na distância cosseno, permite obter bons resultados, alcançando até taxas de acerto da ordem dos 100% de sucessos. Mostra-se também que a técnica ICA tem uma boa imunidade às condições de iluminação do cenário de captura.

De forma a verificar qual a eficácia da metodologia seguida construiu-se um demonstrador que é capaz de processar confortavelmente 30 imagens por segundo num computador comum actual.

Esta dissertação reporta-se ao trabalho desenvolvido no período que decorreu entre Junho e Outubro de 2004.

Abstract

The ever growing needs for communication between humans and machines have revealed that the existing interfaces are not the most appropriate for all applications. As humans interact with each other by speech and body expressions in a daily basis, every communication based on these principles will be natural, intuitive and friendly for humans. This is the main reason why there has been continuing research for new interfaces that minimise the differences between the manner how humans interact with each other and the requirements for Human-Machine interaction tools. The work herein presented is also an effort in that direction.

This thesis describes the creation and analysis of a Human-Machine hands-free interface based on a symbolic gestures recognition system capable of controlling planar movements of an object by identifying five gestures via a computer vision approach.

This work had another goal which was to assess the quality of the results that one can get from the Independent Component Analysis (ICA) technique. We concluded that ICA Architecture I performs very well when used in conjunction with a nearest neighbour classifier using a cosine distance as the similarity measure. We also show that ICA performs well against different lighting conditions.

To prove the effectiveness of the proposed method a prototype demonstrator was developed which is able to process comfortably 30 frames per second in an ordinary personal computer.

This thesis reports the work that has been done from June to October 2004.

Résumé

L'augmentation de la nécessité de communication de l'homme avec les machines révèle que les interfaces existantes Homme-Machine ne sont pas toujours les plus adaptées. Si les hommes, pendant sa vie, communiquent entre soi par la parole et expression corporelle, alors toute la communication soutenue par ces principes sera naturelle, intuitive et amicale pour les humains. Dû a cette constatation il y a eu, et il y aura toujours, une recherche de nouvelles interfaces qui puissent approcher la communication Homme-Machine des formes que les humains utilisent pour communiquer entre soi, dont ce travail est un exemple.

Cette dissertation présente le travail développé dans la création et analyse d'une interface Homme-Machine "mains-libres" constitué par un système de reconnaissance de cinq gestes symboliques, suffisants pour commander le mouvement d'un objet sur un plan, par une approche de vision par ordinateur.

Il fût aussi l'objet de ce travail la vérification de la qualité des résultats que la technique de l'Analyse en Composantes Indépendantes (du Anglais ICA – *Independent Component Analysis*) a donnée. Nous sommes arrivés à la conclusion que l'Architecture I de l'ICA, secondée par un classificateur du voisin le plus proche soutenu par la distance cosinus, permet d'obtenir des bons résultats, jusqu'à un taux de succès de l'ordre de 100%. Il est montré aussi que la technique ICA démontre une bonne immunité aux conditions d'illumination du scénario de capture.

De façon a vérifier quelle est l'efficacité de la méthodologie suivie nous avons construit un démonstrateur capable de traiter confortablement 30 images par seconde sur un ordinateur commun.

Cette dissertation se reporte au travail développé dans la période entre Juin et Octobre 2004.

Agradecimentos

Em primeiro lugar gostaria de agradecer especialmente ao Professor Armando Padilha que soube estar sempre presente dando todo apoio e aconselhamento necessário correspondendo às expectativas criadas por ter sido a pessoa que mais me marcou pela positiva durante o meu percurso académico e que me incutiu o gosto pela área de processamento de imagem e que por isso, se não nos tivéssemos cruzado eu não estaria com certeza a realizar este mestrado. O seu sentido prático sem nunca desprezar o rigor científico é de facto uma das suas virtudes aliado ao seu conhecimento abrangente, à forma clara como os sabe exprimir e à vontade que demonstra em transmiti-los.

Ao Professor Aurélio Campilho, outro docente importante na minha formação, que me conseguiu incutir o gosto por esta área e de quem tenho muito boas recordações, o meu obrigado pelo apoio dado na candidatura a este mestrado.

À Sónia Gouveia pelo seu sentido crítico, pelas suas ideias e pela disponibilidade que teve para as longas discussões sobre Estatística.

À comissão de mestrado que soube não restringir os conteúdos leccionados a matéria puramente de Engenharia e Ciências exactas.

Aos meus amigos, eles sabem quem são, por terem estado presentes e ausentes sempre que era preciso e por terem conseguido compreender a minha indisponibilidade.

Por fim agradeço à minha família pelo apoio importante que me deu e por terem conseguido compreender a minha ausência e indisponibilidade durante a realização deste mestrado.

Índice

1. INTRODUÇÃO.....	1
1.1 GESTOS.....	3
1.2 OBJECTIVOS	4
1.3 DESCRIÇÃO DO CASO DE ESTUDO.....	5
1.4 ORGANIZAÇÃO DO RELATÓRIO	7
2. APLICAÇÕES DO RECONHECIMENTO DE GESTOS.....	9
2.1 LINGUAGEM GESTUAL	9
2.2 TELECOMUNICAÇÕES.....	10
2.3 INTERACÇÃO 3D.....	10
2.4 TELEPRESENÇA E TELEROBÓTICA	11
2.5 LAZER E FINS SOCIAIS	11
3. REVISÃO DO ESTADO DA ARTE.....	13
3.1 RECONHECIMENTO DE GESTOS E INTERACÇÃO GESTUAL	13
3.1.1 <i>Luvas instrumentadas</i>	13
3.1.2 <i>Visão por computador</i>	16
3.1.2.1 Aquisição de dados e pré-processamento	17
3.1.2.2 Segmentação	17
3.1.2.3 Extração de características	18
3.1.2.4 Classificador.....	20
3.2 ICA - ANÁLISE DE COMPONENTES INDEPENDENTES	21
3.2.1 <i>Introdução</i>	21
3.2.2 <i>Formulação matemática do modelo ICA</i>	23
3.2.3 <i>Estimação</i>	24
3.2.4 <i>Redução da dimensionalidade dos dados</i>	28
3.2.5 <i>Aplicação de ICA em reconhecimento de gestos</i>	29
3.2.5.1 Arquitectura I – Base de imagens estatisticamente independentes.....	30
3.2.5.2 Arquitectura II – Coeficientes estatisticamente independentes	31

4.	ABORDAGEM ADOPTADA	33
4.1	SEGMENTAÇÃO	34
4.2	NORMALIZAÇÃO	39
4.3	ICA.....	42
4.3.1	<i>Modo de treino - bloco ICA.....</i>	<i>44</i>
4.3.2	<i>Modo de teste ou corrente – bloco Transformação ICA.....</i>	<i>45</i>
4.4	CLASSIFICADOR	45
5.	ANÁLISE DE RESULTADOS	49
5.1	SENSIBILIDADE À DIMENSÃO DAS IMAGENS NORMALIZADAS	54
5.2	SENSIBILIDADE AO NÚMERO DE COMPONENTES INDEPENDENTES	56
5.3	SENSIBILIDADE ÀS IMAGENS DE TREINO	58
5.4	SENSIBILIDADE ÀS IMAGENS DE REFERÊNCIA	59
5.5	SENSIBILIDADE ÀS CONDIÇÕES DE CAPTURA	61
5.6	SENSIBILIDADE AO UTILIZADOR	64
5.7	COMPARAÇÃO COM UM CLASSIFICADOR <i>ADHOC</i>	66
5.8	DEMONSTRADOR DE COMANDO DE MOVIMENTO	69
5.8.1	<i>Sistema de reconhecimento da contagem gestual chinesa.....</i>	<i>72</i>
6.	CONCLUSÕES E TRABALHO FUTURO.....	75
6.1	CONCLUSÕES	75
6.2	TRABALHO FUTURO	77
	REFERÊNCIAS	79

Lista de Figuras

Figura 1.1 – Campo de visão da câmara de vídeo	5
Figura 1.2 – Gestos a reconhecer (travar, acelerar, esquerda, direita, neutro)	7
Figura 3.1 - Digital Data Entry Glove	14
Figura 3.2 - VPL DataGlove.....	15
Figura 3.3 - Dexterous HandMaster	15
Figura 3.4 - Power Glove.....	15
Figura 3.5 - CyberGlove	16
Figura 3.6 – Diagrama de blocos de um sistema de visão por computador	17
Figura 3.7 – Observações como mistura das fontes.....	23
Figura 3.8 – Modelo do processo ICA (adaptado de [Draper, Baek et al., 2003])	24
Figura 3.9 – Imagens da base resultante da Arquitectura I.....	30
Figura 3.10 – Representação ICA do gesto x_i pela Arquitectura I ($a_{i1}, a_{i2}, \dots, a_{in}$)	31
Figura 3.11 – Imagens da base resultante da Arquitectura II	31
Figura 3.12 – Representação ICA do gesto x_i pela Arquitectura II ($u_1, u_2, \dots, u_n$)	31
Figura 4.1 – Diagrama de blocos do reconhecedor em modo de treino	34
Figura 4.2 – Diagrama de blocos do reconhecedor em modo corrente	34
Figura 4.3 – Pontos de interesse do cubo de cores RGB (sem e com condição de desvio padrão, respectivamente, esquerda e direita)	35
Figura 4.4 – Projecção vertical e horizontal de uma imagem binária.....	36
Figura 4.5 – Ocupação desalinhada de diferentes gestos.....	37
Figura 4.6 – Segmentação sem e com a informação da projecção horizontal	38
Figura 4.7 – Melhoria dada pela flexibilização do algoritmo de localização da zona de projecção horizontal mais larga	38
Figura 4.8 – Imagem original	39
Figura 4.9 – Imagem segmentada.....	39
Figura 4.10 – Binarização da imagem segmentada	40
Figura 4.11 – Direcção do eixo de menor inércia.....	40
Figura 4.12 – Direcções para onde são rodadas as imagens.....	41
Figura 4.13 – Imagem rodada.....	41
Figura 4.14 – Enchimento necessário após redimensionar as imagens	42
Figura 4.15 – Gestos normalizados.....	42

Figura 4.16 – Transformação da matriz imagem em vector imagem.....	43
Figura 5.1- Diferentes capacidades de discriminação	51
Figura 5.2 – Semelhança dos gestos de teste a cada classe - Arquitectura I.....	53
Figura 5.3 – Semelhança dos gestos de teste a cada classe - Arquitectura II.....	54
Figura 5.4 – Semelhança às duas classes mais próximas em função das dimensões das imagens.....	56
Figura 5.5 – Taxa de acerto em função do número de componentes independentes	57
Figura 5.6 – Semelhança às duas classes mais próximas em função do número de componentes independentes	58
Figura 5.7 – Semelhança às duas classes mais próximas em função do número de imagens de treino.....	59
Figura 5.8 – Taxa de acerto em função do número de imagens de referência	60
Figura 5.9 – Semelhança em função do número de imagens de referência	60
Figura 5.10 – Fluxograma do classificador <i>ad hoc</i>	68
Figura 5.11 – Demonstrador de comando de movimento por gestos simbólicos	70
Figura 5.12 – Contagem gestual chinesa.....	72

Lista de Tabelas

Tabela 4.1 – Semelhança cosseno entre imagens de gestos	47
Tabela 5.1 – Desempenho do reconhecedor	53
Tabela 5.2 – Taxa de acerto em função das dimensões das imagens	55
Tabela 5.3 – Classificações erradas na Architectura II	57
Tabela 5.4 – Classificações erradas na Architectura II no caso de 1 imagem de referência por gesto	61
Tabela 5.5 – Semelhança entre imagens	62
Tabela 5.6 – Desempenho para cenários de captura diferentes - Architectura I	63
Tabela 5.7 – Desempenho para cenários de captura diferentes - Architectura II	63
Tabela 5.8 – Desempenho para teste sem luva – Architectura I	64
Tabela 5.9 – Desempenho para teste sem luva – Architectura II	64
Tabela 5.10 – Desempenho com utilizadores diferentes - Architectura I	65
Tabela 5.11 – Desempenho com utilizadores diferentes - Architectura II	65
Tabela 5.12 – Desempenho com utilizadores diferentes - Architectura I	66
Tabela 5.13 – Desempenho com utilizadores diferentes - Architectura II	66
Tabela 5.14 – Características de diferentes gestos	67
Tabela 5.15 – Desempenho com 1 imagem de referência por gesto	72
Tabela 5.16 – Desempenho com 3 imagem de referência por gesto	72

CAPÍTULO 1

Introdução

No dia a dia os humanos têm necessidade de comunicar entre si com grande frequência. Quando o fazem presencialmente, esta comunicação assenta em duas formas distintas mas ao mesmo tempo complementares: fala e expressão corporal. Enquanto que a fala estimula sensores auditivos, a expressão corporal, cujos maiores expoentes são os gestos e as expressões faciais, assenta em estímulos visuais. Se por vezes uma palavra ou uma frase é suficiente para transmitir uma mensagem há outras em que basta um gesto ou uma expressão. Com a utilização simultânea destas duas formas de comunicar consegue-se dar maior expressividade à mensagem tornando-a mais clara e/ou convincente.

Por conseguinte, toda a comunicação assente em fala e expressão corporal será natural para os humanos. O problema surge quando é necessário comunicar com as máquinas. Inevitavelmente, foi o Homem que se teve de adaptar às máquinas e não o contrário. Inicialmente os dispositivos de interacção para com as máquinas eram muito rudimentares mas, com o evoluir da tecnologia, foram surgindo novos dispositivos, como é o caso do teclado, rato e *joystick*, sempre com o objectivo de tornar a interacção mais intuitiva. Mesmo assim ainda não se conseguiu satisfazer as necessidades elementares de toda a sociedade. Por exemplo, grande parte das pessoas que sofrem de paralisia cerebral não têm a coordenação suficiente para escrever à mão ou num teclado

mas têm a destreza suficiente para conseguir interagir com um sistema, adaptativo, de reconhecimento de gestos [Pausch e Williams, 1992].

Como se conclui, apesar da evolução dos dispositivos de interface Homem-Máquina, há ainda um longo caminho pela frente para as tornar mais naturais, assemelhando-as à forma como os humanos interagem entre si, e colocando-as ao alcance de todos.

Por outro lado, há uma crescente utilização dos computadores, quer como ferramenta de trabalho quer como instrumento de lazer. Actualmente utiliza-se o computador para executar tarefas tão diversas como a escrita de um documento, consulta de informação na Internet, realização de compras, pagamento de impostos, organização de álbum fotográfico digital e puro entretenimento, entre outros.

Por estes motivos, a interface de acesso ganha uma importância acrescida. Esta deve estar pensada de forma a tornar a interacção Homem-Máquina o mais amigável e intuitiva possível. Uma interface pouco natural e intuitiva dificulta o acesso à tecnologia a pessoas menos familiarizadas. Ao passo que uma interface simples, intuitiva, amigável e natural aumenta o número de possíveis utilizadores e melhora a experiência do utilizador (*user experience*).

Paralelamente a esta necessidade de novas interfaces, a evolução do poder computacional tem sido extraordinária e tudo indica que o continue a ser. O que abre portas à utilização do computador para realizar algumas tarefas que, no passado, eram computacionalmente inviáveis. E o mesmo sucederá no futuro! Tarefas com custo computacional demasiado elevado para os dias de hoje, no futuro poderão ser executadas por um computador.

É a conjugação destes 3 factores, banalização do uso do computador, aumento do poder computacional e necessidade da criação de interfaces Homem-Máquina alternativas às convencionais que motiva e possibilita trabalhos semelhantes ao que aqui se apresenta. “Porque não interagir com o computador/máquina através de gestos?”. Se repararmos, no dia a dia, utilizamos frequentemente as mãos para comunicar e que, nalguns casos, dispensam até o recurso à comunicação oral. O acenar “adeus”, “parar” e o “apontar” são exemplos disso mesmo. Veja-se o caso do polícia sinaleiro que apenas com gestos consegue instruir os condutores de como devem circular. É então plausível que se recorra também aos gestos para interagir com os computadores. Neste trabalho serão

exploradas metodologias que, no seu conjunto, visam reconhecer gestos simbólicos, para comando de movimento, com o intuito de controlar um objecto móvel.

No contexto de interacção Homem-Máquina, ao longo deste trabalho serão utilizados os termos computador e máquina indiferenciadamente.

1.1 Gestos

Na literatura não existe uma definição do termo “gesto” que seja unanimemente aceite. A sua definição é muito dependente da área em que está a ser abordado e do objectivo do trabalho que se está a realizar. No entanto, uma definição adequada é a que se pode encontrar em algumas enciclopédias, nomeadamente [Lello, 1974]:

Movimento do corpo, especialmente dos braços e mãos, para exprimir ideias.

No contexto deste trabalho, restringir-nos-emos aos gestos executados apenas pelas mãos, mais especificamente, por uma só mão.

Quanto ao contexto de aplicação, podemos considerar algumas classes de gestos (adaptada de [Wu e Huang, 1999]): gesticulação, pantomimas, linguagem gestual, emblemas, etc. A gesticulação representa os gestos que adornam o discurso. A sua principal função é trazer maior clareza, expressividade e emotividade ao discurso. Pantomimas são peças teatrais em que os actores se exprimem apenas por gestos. A linguagem gestual é a língua “falada” pelos surdos-mudos que tem um vocabulário próprio e uma gramática. Por emblemas entendem-se os gestos que representam determinadas mensagens dependentes do contexto, cultura e convenções. Por exemplo, o gesto representado pelo dedo indicador e médio esticados, em Portugal representa o número 2 enquanto que no Reino Unido é entendido como um gesto ofensivo.

De entre os diferentes tipos de gestos pode-se destacar os gestos simbólicos (que serão abordados neste trabalho). Os gestos simbólicos têm, quase sempre, o objectivo de controlar e manipular objectos sendo o seu significado predeterminado pelo sistema que os analisa. Pegar, rodar, pousar e apontar são exemplo de acções que eles podem representar.

Quanto à forma, os gestos podem ser distinguidos entre estáticos e dinâmicos. Por estáticos, tal como o nome indica, entende-se todo o tipo de gestos que não necessitam de recorrer a movimento para transmitir a mensagem pretendida. É o caso do gesto de “pedir boleia”. Por outro lado, os gestos dinâmicos são os que necessitam de movimento para exprimir a mensagem pretendida. Logo, a sua evolução temporal é importante [Birk e Moeslund, 1996]. O gesto de dizer “adeus” (executado por movimentos laterais com a mão aberta) é um bom exemplo de um gesto dinâmico visto que se não tiver movimento poderá ter outro significado completamente distinto, caso de “parar” (mão aberta estática).

Independentemente de se tratar de um gesto estático ou dinâmico, a sua realização é constituída por 3 fases [Kendon, 1986]: preparação, definição, retracção. A fase mais importante é a da definição (“*stroke*”) já que é a altura em que o gesto atinge o seu auge e que, por isso, contém a maior parte da informação útil ao reconhecimento do gesto. A preparação e a retracção são as fases, respectivamente, anterior e posterior e que também podem ser úteis para o reconhecimento de gestos, principalmente nos casos em que eles têm uma gramática associada e em que, por isso, a ocorrência depende do(s) anterior(es). O que pode indiciar existir informação importante na transição entre gestos (fases de preparação e retracção). Contudo neste trabalho apenas iremos utilizar a informação presente na fase de definição.

1.2 Objectivos

O trabalho descrito nesta dissertação tinha dois objectivos principais. O primeiro consistia em criar um sistema de reconhecimento de gestos simbólicos, utilizando uma metodologia baseada em visão por computador, que permita comandar o movimento de um objecto recorrendo apenas a gestos estáticos. Assim, definiram-se 5 gestos de comando (‘travar’, ‘acelerar’, ‘esquerda’, ‘direita’ e ‘neutro’) que devem ser executados com a mão direita e que permitem o controlo do movimento planar pretendido.

O segundo objectivo era a exploração da técnica ICA (*Independent Component Analysis*) por forma a verificar quais os resultados que se conseguem obter com a sua utilização num contexto de reconhecimento de gestos. Visto que, de acordo com o que

se conseguiu apurar, ainda ninguém publicou resultados da aplicação de ICA ao reconhecimento de gestos, este trabalho pretende também dar esse contributo.

Dada a limitação de tempo para a elaboração deste trabalho não é requerido que o reconhecedor de gestos simbólicos opere em tempo-real, contudo no desenvolvimento das rotinas necessárias à execução das tarefas pretendidas far-se-á esforços no sentido de minimizar os tempos de processamento.

Como a definição de tempo-real é um pouco vaga, neste trabalho entender-se-á que 10 a 15 imagens por segundo já é suficiente para se considerar como tempo-real mas o ideal será capacidades de processamento da ordem das 25-30 imagens por segundo.

1.3 Descrição do caso de estudo

Como já foi adiantado o caso de estudo será um sistema de reconhecimento visual de 5 gestos simbólicos por forma a comandar o movimento de um objecto num plano. Para este fim o cenário de captura de imagem será controlado a fim de não tornar demasiado complexa a localização da mão na imagem (tarefa denominada de segmentação que será explicada e devidamente enquadrada na secção 4.1). Assim, no campo de visão da câmara de captura de imagem/vídeo, responsável por adquirir as imagens do gesto executado, deverá apenas existir um objecto de cor clara, a que corresponderá a mão, e tudo o resto deverá ser preto, tal como se pode ver na figura seguinte.



Figura 1.1 – Campo de visão da câmara de vídeo

Para minimizar os problemas de variação de cor da mão e também para aumentar a diferença entre o fundo e a mão, esta terá de estar equipada com uma luva fina de cor branca. Caso não se fizessem estas restrições a operação de segmentação tornar-se-ia demasiado complexa e, por si só, seria assunto suficiente para justificar uma tese. A segmentação em qualquer tipo de fundos é um problema ainda em aberto e com muita margem de evolução. Um bom trabalho nessa área pode ser encontrado em [Cui e Weng, 1999], realizado por Cui e Weng.

Quanto à forma de abordagem do problema seguir-se-á um reconhecimento quadro a quadro em detrimento de uma abordagem puramente temporal por nos parecer mais adequada ao reconhecimento de gestos estáticos e por ter menor complexidade de implementação. Já Schlenszig, citado em [Birk e Moeslund, 1996], apontava estas vantagens no reconhecimento de gestos quadro a quadro.

Depois aplicar-se-á a técnica ICA que, na forma como vai ser aqui usada, se inclui nas técnicas de visão por computador baseadas na aparência (abordadas no capítulo 3) já que segundo Tan e Krieghan [Tan, Kriegman et al., 2002] são de mais fácil implementação e mais robustas do que as baseadas em características porque a extracção de algumas características, tais como contornos, pode não ser uma tarefa trivial. Apesar do desconhecimento dos resultados que poderemos alcançar com ICA há indícios que sejam satisfatórios visto que com PCA (*Principal Component Analysis*) já se conseguiram obter bons resultados, nomeadamente, por Birk e Moeslund [Birk e Moeslund, 1996]. E sabendo-se que PCA está ligado ao ICA por serem ambas técnicas de projecção em espaços próprios, residindo a principal diferença no facto do PCA gerar componentes não correlacionadas enquanto que o ICA gera componentes independentes, o que é uma condição mais forte como se irá ver na secção 3.2. Além desse facto, o PCA exige que os dados sejam gaussianos enquanto que o ICA é menos restritivo e apenas exige que não o sejam o que alarga a sua área de aplicação.

Relativamente aos gestos que se pretende reconhecer eles encontram-se na Figura 1.2 com a orientação aproximada de como se devem apresentar. É importante referir que as imagens desta figura estão como a câmara de captura as vê. Por isso, o gesto ‘direita’ resulta numa imagem que mostra uma mão a apontar para o lado esquerdo e o gesto ‘esquerda’ numa imagem com a mão a apontar para a direita.



Figura 1.2 – Gestos a reconhecer (travar, acelerar, esquerda, direita, neutro)

Um facto importante é que os gestos devem ser executados com a mão direita e devem ser realizados da seguinte forma:

- **travar** mão aberta e aproximadamente paralela ao plano imagem
- **acelerar** mão espalmada, aproximadamente perpendicular ao plano imagem, com os dedos encostados uns aos outros
- **esquerda** mão fechada com o polegar aberto e a apontar para o lado esquerdo (do ponto de vista do gesticulador)
- **direita** mão fechada com o indicador esticado apontando para o lado direito (do ponto de vista do gesticulador)
- **neutro** mão fechada

Com apenas 4 gestos já é possível controlar totalmente o movimento do objecto num plano. A existência do gesto ‘neutro’ servirá apenas para quando o utilizador não pretende dar nenhuma ordem, trata-se de um gesto de repouso. Desta forma o utilizador saberá sempre como se apresentar ao sistema mesmo em situações que não pretenda dar qualquer ordem.

1.4 Organização do relatório

Neste primeiro capítulo pretendeu-se dar o enquadramento do trabalho aqui apresentado assim como o seu âmbito e quais os objectivos que o regem.

No capítulo 2 dar-se-ão alguns exemplos de aplicações de sistemas de reconhecimento de gestos a fim de se tomar consciência da diversidade de utilizações possíveis que um sistema deste tipo pode ter.

O capítulo 3 pretende dar uma visão geral do estado da arte de sistemas de reconhecimento de gestos e interacção gestual, como surgiram, em que princípios se estabelecem e as técnicas a que habitualmente recorrem. Além disso também será

apresentada a teoria suficiente à compreensão da técnica ICA tal como é utilizada neste trabalho.

No capítulo 4 é feita a descrição mais pormenorizada do sistema de reconhecimento de gestos simbólicos que foi desenvolvido. Alguma das matérias aqui abordadas já mereceram a atenção em capítulos precedentes mas neste capítulo serão expostas sob uma índole mais prática por forma a propiciar uma melhor compreensão e também para permitir que terceiros possam reproduzir os resultados obtidos no decorrer deste trabalho.

A análise dos resultados do desempenho do reconhecedor desenvolvido em vários cenários diferentes é analisada por forma a averiguar como se comporta o ICA em função do número de componentes independentes, dimensão das imagens e diferentes utilizadores, entre outros. É também neste capítulo que vai ser apresentado o demonstrador implementado e quais os modelos de velocidade e da mudança de direcção com que é concebido e qual a taxa de processamento conseguida.

Por fim tiram-se conclusões sobre o trabalho realizado e apontam-se caminhos interessantes de exploração futura.

CAPÍTULO 2

Aplicações do reconhecimento de gestos

Os sistemas de reconhecimento de gestos têm inúmeras aplicações das quais se podem destacar as que se apresentam neste capítulo.

2.1 Linguagem gestual

A linguagem gestual é a forma pela qual os surdos-mudos se podem expressar. Para se poder comunicar com um surdo-mudo é, então, necessário saber “falar” esta linguagem gestual ou recorrer a uma pessoa capaz de fazer a tradução. Esta limitação que os surdos-mudos têm para comunicar causa algumas dificuldades no dia a dia, aumentando a sua dependência de terceiros que consigam fazer a ponte entre a linguagem gestual e a oral. A criação de sistemas reconhecedores de gestos pode atenuar este problema aumentando significativamente a qualidade de vida de uma pessoa surda-muda ao dar-lhe uma maior independência.

Starner, Weaver e Pentland [Starner e Pentland, 1995; Starner, Weaver et al., 1998] foram dos primeiros a explorar o reconhecimento automático da linguagem gestual americana (ASL – *American Sign Language*) mas outros os seguiram, como é o caso de [Vogler e Metaxas, 1998; Vogler e Metaxas, 1999; Vogler e Metaxas, 2001; Parashar, 2003].

Os trabalhos de reconhecimento de linguagem gestual não se restringem apenas à linguagem americana. Há também registo de trabalhos em linguagem gestual Chinesa ([Gao, Ma et al., 2000; Wang, Gao et al., 2002]), Alemã ([Bauer e Hienz, 2000]), Holandesa ([Assan e Grobel, 1998]), Tailandesa ([Liang e Ouhyoung, 1998]) e Inglesa ([Sweeney e Downton, 1997]). Imagawa, Lu e Igi [Imagawa, Lu et al., 1998b], em 1998, encontravam-se a desenvolver um sistema de tradução bidireccional entre a linguagem gestual Japonesa e a oral com o objectivo de facilitar a comunicação entre pessoas sem dificuldades de audição com as surdas-mudas mas cujo resultado não se teve conhecimento.

2.2 Telecomunicações

O reconhecimento de gestos em fotografia ou vídeo digital também pode ser aplicado em telecomunicações para efeitos de análise, armazenamento e transmissão de dados. Para reduzir a largura de banda necessária à transmissão de vídeo uma técnica que se usa, nomeadamente em MPEG-4, é modelizar a imagem, ou parte dela, transmitindo-se apenas os parâmetros do modelo em detrimento de toda a imagem.

Ainda em telecomunicações, para efeitos de anotações automáticas de vídeos e/ou imagens (em, por exemplo, MPEG-7 ou RDF), sistemas de reconhecimento automático para evitar que os humanos tenham de executar tarefas muito aborrecidas e repetitivas. Um exemplo de uma aplicação deste tipo pode ser encontrada em Ju *et al.* [Ju, Black et al., 1997].

2.3 Interacção 3D

Uma das maiores aplicações de sistemas de reconhecimento de gestos está na criação de interfaces Homem-Máquina alternativas, algumas das quais dedicadas a controlo multidimensional.

Segen e Kumar [Segen e Kumar, 1998] constroem um sistema de visão por computador para controlo 3D e aplicam-no para manipular objectos virtuais, comandar um avião virtual e para jogar *Doom*.

Outra utilização de manipulação 3D é feita por Zeller *et al.* [Zeller, Phillips et al., 1997] que apresentam uma aplicação de modelização bio-molecular que permite modelizar

bio-polímeros recorrendo a gráficos 3D de moléculas e a um programa de simulação dinâmica molecular.

2.4 Telepresença e telerobótica

Ambientes hostis (exemplo, manutenção de reactores nucleares) ou ambientes de difícil acesso físico (exemplo, missões espaciais e submarinas) criam necessidades de controlo remoto em que os gestos surgem, muitas vezes, como uma forma de interacção apropriada. Mesmo após formação dos operadores, os convencionais dispositivos de interacção (*joystick* e botões) não são muito intuitivos devido aos vários graus de liberdade existentes. Assim, o comando por gestos, e movimentação do braço, surge como a forma mais fácil e intuitiva visto que é algo que os humanos fazem com naturalidade.

Exemplos de aplicação de gestos, para estes fins, podem ser encontrados em [Cipolla e Hollinghurst, 1996; Kang e Ikeuchi, 1997; Matsumoto, Shimada et al., 2002].

2.5 Lazer e fins sociais

Outra das aplicações que o reconhecimento de gestos permite inclui-se nas áreas de lazer e fins sociais das quais destacamos duas.

Davis e Bobick [Davis e Bobick, 1998] fizeram um protótipo de um monitor de aeróbica virtual em que o utilizador pode personalizar e criar os treinos de acordo com os seus desejos e necessidades.

Já Becker e Pentland [Becker e Pentland, 1997] criaram uma aplicação/jogo para relaxamento dos pacientes de cancro em que os utilizadores comandam um glóbulo branco no fluxo sanguíneo. O objectivo é digerir as células malignas que aparecem pelo caminho sendo a interface totalmente mãos livres e as ordens dadas pela execução de gestos de T'ai Chi.

CAPÍTULO 3

Revisão do estado da arte

3.1 Reconhecimento de gestos e interacção gestual

Nos sistemas de reconhecimento de gestos, uma das partes mais importantes é a de aquisição de dados, porque é a responsável por fazer medidas ao gesto que se está a realizar por via de, por exemplo, sensores ópticos, acústicos e electromagnéticos. Medidas essas que servirão para comparar e distinguir os gestos admitidos.

Quanto aos dispositivos de aquisição de dados, verifica-se a existência de duas correntes importantes: a das luvas instrumentadas e a da visão por computador.

3.1.1 Luvas instrumentadas

Os primeiros sistemas a serem desenvolvidos para o reconhecimento de acções executadas pela mão baseavam-se na utilização de luvas instrumentadas.

A primeira a ser inventada, por DeFanti e Sandin da Universidade de Illinois, data de 1977 e tem o nome de *Sayre' Glove* visto ter sido baseada numa ideia de Rich Sayre da Universidade de Chicago. Esta luva é constituída por pequenos tubos flexíveis que têm uma fonte de luz numa extremidade e uma célula fotossensível na outra. A luz recebida

pela célula fotossensível varia consoante a flexão dos dedos o que permite obter uma medida do grau de flexão dos dedos, essencial para se reconhecer o gesto executado.

Em 1983, Gary Grimes, afecto aos Bell Labs, patenteia a sua *Digital Data Entry Glove* (Figura 3.1) que alia aos sensores ópticos, para medir a flexão dos dedos, sensores tácteis nas pontas dos dedos e um sensor 3SPACE que permite determinar a orientação e a posição do pulso. O posicionamento destes sensores é tal que a combinação das medidas permite distinguir entre 80 e 96 caracteres ASCII, o suficiente para cobrir um conjunto significativo do alfabeto manual dos americanos surdos [Sturnman, 1992].

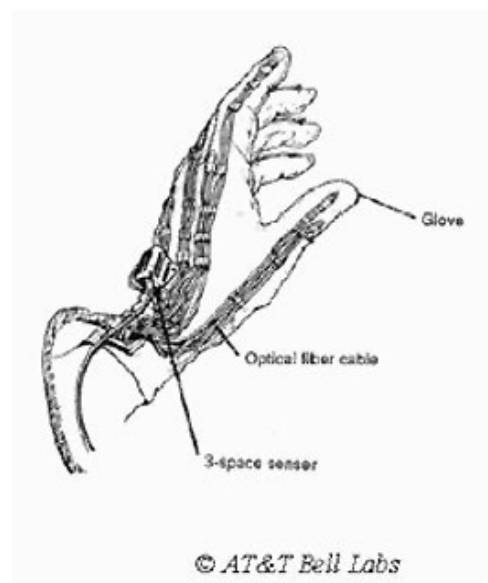
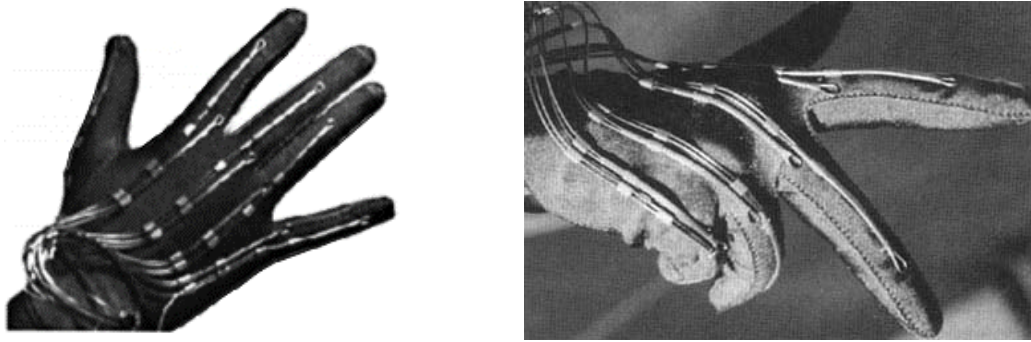


Figura 3.1 - Digital Data Entry Glove

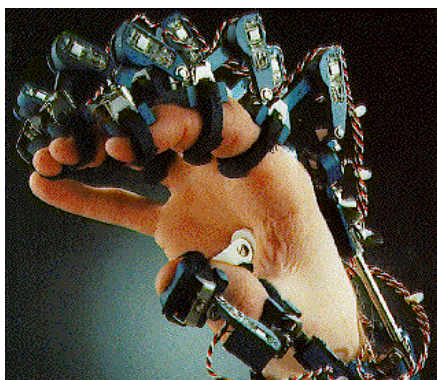
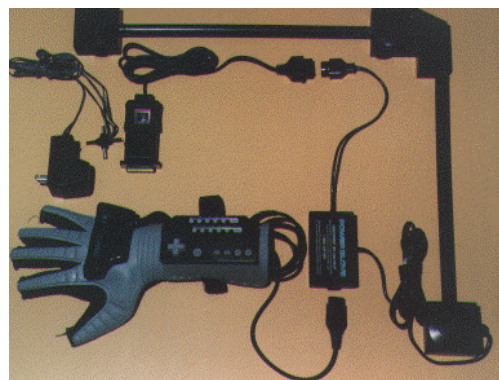
Mas é em 1987 que surge a luva que teve o maior sucesso comercial, a *VPL DataGlove* (Figura 3.2), criada por Thomas Zimmerman. Esta luva recorre a 2 fibras ópticas, por cada dedo, em que os sinais por si gerados são enviados para um processador que, baseado na obrigatória calibração inicial, determina os ângulos das junções dos dedos [Watson, 1993]. Juntamente com a luva é incluído software capaz de reconhecer gestos através do mapeamento das junções em comandos. A orientação e a posição da mão, tal como na *Digital Data Entry Glove*, são determinadas pela utilização de um sensor Polhemus 3SPACE.

O Polhemus é um dispositivo muito utilizado nesta área que consiste num sensor capaz de determinar os 6 graus de liberdade da mão (posição e orientação) através da monitorização dos impulsos de campos magnéticos de baixa frequência emitidos por uma estação transmissora.

**Figura 3.2 - VPL DataGlove**

Embora esta luva tenha trazido grandes avanços às luvas instrumentadas, ela tinha um inconveniente importante. Era necessária uma calibração inicial por utilizador e também após algumas horas de uso. Além deste inconveniente, a luva não era muito precisa quando o utilizador tinha mãos muito pequenas ou muito grandes.

Assim surgiu a *Dexterous HandMaster* (Figura 3.3), desenvolvida pela empresa *Exos*, cuja principal vantagem era uma maior precisão mas, no entanto, um pouco mais desconfortável visto ter uma estrutura de alumínio na parte posterior da mão. A medição da flexão de cada um dos dedos é realizada por sensores do efeito de Hall (efeito que surge quando um material condutor, percorrido por corrente eléctrica, é sujeito a um campo magnético).

**Figura 3.3 - Dexterous HandMaster****Figura 3.4 - Power Glove**

Em 1989, inspirados pelo sucesso da *VPL DataGlove*, a empresa de brinquedos *Mattel*, lançou a *Power Glove* (Figura 3.4) para ser usada como um controlador de baixo custo para jogos da consola *Nintendo* [Watson, 1993]. A precisão desta luva é pior do que a

da *VPL DataGlove* e da *Dexterous HandMaster* porque a flexão de cada dedo é medida apenas por um sensor piezo-resistivo. Apesar disso, é capaz de determinar os 6 graus de liberdade, da posição e orientação, através de ondas ultra-sónicas emitidas por transmissores colocados em ambos os lados da luva. Os receptores são colocados junto ao monitor de televisão e, devido ao conhecimento prévio da localização relativa dos transmissores, consegue-se realizar a triangulação pretendida.

Também em 1989 surge uma outra luva, a *Virtex CyberGlove*, que surgiu de um trabalho de James Kramer na Stanford University. Existem duas versões da luva, uma com 18 e outra com 22 sensores resistivos de flexão que permitem obter dados do estado dos ângulos das junções dos dedos.



Figura 3.5 - CyberGlove

Para uma análise mais exaustiva das luvas instrumentadas aconselha-se a consulta de [Sturnman, 1992; Watson, 1993; Sturman e Zeltzer, 1994].

3.1.2 Visão por computador

As luvas instrumentadas são capazes de dar informação referente ao estado das junções dos dedos, que auxilia no reconhecimento do gesto. Contudo, obter esta mesma informação, sem necessitar da luva e dos cabos que a ligam ao sistema de análise de dados, seria bastante mais confortável e além disso permitiria que os utilizadores se movimentassem com maior liberdade [O'Hagan, Zelinsky et al., 2000]. Assim, surge a outra corrente: a utilização de visão por computador em sistemas de reconhecimento de gestos.

Um sistema de visão por computador, no contexto de reconhecimento de gestos, tem o diagrama de blocos da Figura 3.6, comum a muitas outras aplicações.

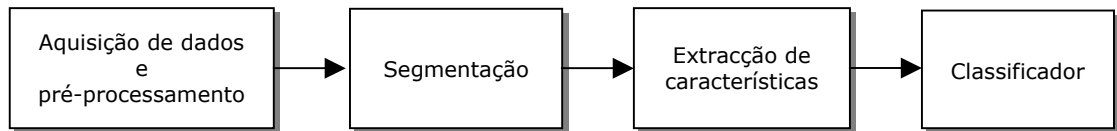


Figura 3.6 – Diagrama de blocos de um sistema de visão por computador

3.1.2.1 Aquisição de dados e pré-processamento

A captura de imagem por uma câmara de vídeo (noutros casos, apenas câmara fotográfica) constitui a primeira fase de um sistema de visão por computador. É aqui que se tomam algumas decisões importantes que influenciarão o desempenho do sistema, nomeadamente, estrutura de iluminação, restrições de movimentos, posicionamento relativo da câmara ao objecto a capturar, frequência de amostragem e gestos admitidos.

Uma das restrições mais frequentes, dos sistemas de reconhecimento de gestos, é a utilização de fundos de cor predeterminada, luvas coloridas ou marcas nas pontas dos dedos [Ahmad, 1994; Davis e Shah, 1995; Starner e Pentland, 1995; Imagawa, Lu et al., 1998a; Starner, Weaver et al., 1998; O'Hagan, Zelinsky et al., 2000] a fim de facilitar a fase seguinte, a segmentação da mão.

Pode também existir pré-processamento, por exemplo, equalização do histograma de brilhos ou operações pontuais de manipulação de brilho e contraste para minimizar problemas de iluminação e realçar certos objectos.

Por “objectos”, no decorrer deste relatório e no contexto de visão por computador, entende-se o que se pretende detectar em oposição a “fundo” que é o que não interessa, o que é acessório. Por isso, iremos utilizar a palavra “objecto” para referir, por exemplo, a mão que se pretende localizar na imagem capturada.

3.1.2.2 Segmentação

As técnicas utilizadas para a segmentação podem ser divididas em dois grupos: cor ou movimento.

Segmentação através da cor assume que o objecto que se pretende detectar tem uma cor diferente dos restantes elementos presentes na cena capturada, o que nem sempre é verdade. O modelo de cor que muitas vezes se usa é HSV (*Hue-Saturation-Value*), ou também YUV (Luminância, croma vermelho, croma azul) [Michael Stark, 1995] visto que, como separam a componente de luminância das de crominância, conseguem ser um

pouco menos sensíveis à variação da iluminação da cena do que o modelo RGB (*Red-Green-Blue*) [Yang e Waibel, 1996].

A utilização do movimento para detectar a localização da mão, tem o pressuposto de que o único objecto em movimento na cena é a mão. Através de diferença de quadros (*frames*) consecutivos detectam-se zonas onde houve variações de tonalidade e que, por isso, têm grande probabilidade de serem zonas onde está, ou por onde passou, a mão.

A segmentação ainda é um problema em aberto visto que a utilização da cor ou do movimento, ou a conjugação das duas, não consegue resolver totalmente o problema devido a, entre outras dificuldades, variações de cor da pele, deficiente iluminação, sombras e presença de fundos complexos (onde existem objectos com cor semelhante à dos procurados e/ou onde o fundo não é uniforme).

Cui e Weng [Cui e Weng, 1999] propõem uma abordagem um pouco diferente através da utilização de “*attention images from multiple fixations*” e que basicamente consiste em subdividir sucessivamente uma imagem, noutras mais pequenas, com o intuito de reduzir o número de *pixels* pertencentes ao fundo da cena. Assim, ao progredir nesta pirâmide hierárquica vai-se encontrando imagens com um número de pontos pertencente ao fundo cada vez menor. Os resultados obtidos são bastante satisfatórios porque, numa cena com fundo complexo, conseguiram obter 95% de segmentações correctas.

3.1.2.3 Extracção de características

Num sistema de reconhecimento de gestos de visão por computador, para extracção de características, tipicamente têm-se duas abordagens relativas à natureza das técnicas utilizadas:

- baseada na aparência (*appearance-based*) ou vista (*view-based*)
- baseada nas características (*feature-based*)

Apesar de existir alguma ambiguidade nas técnicas pertencentes a cada uma destas abordagens e das fronteiras não estarem bem definidas, ao longo deste trabalho seguir-se-á a classificação que se entendeu mais coerente e que é utilizada, por exemplo, em [Tan, Kriegman et al., 2002].

Nas técnicas baseadas na aparência, a imagem é encarada como um ponto num espaço N dimensional, em que N representa o número de *pixels* pertencentes à imagem. Ao contrário, nas técnicas baseadas em características, tal como o nome indica, são

extraídas características intrínsecas dos objectos presentes na imagem, tais como momentos, rectângulos envolventes, áreas, contornos, cantos, etc.

Abordagem baseada na aparência

A maior parte das técnicas utilizadas nesta abordagem inserem-se no grupo das transformações para os chamados espaços próprios, caso de: PCA (*Principal Component Analysis*), e ICA (*Independent Component Analysis*).

Birk e Moeslund [Birk e Moeslund, 1996] utilizaram extensivamente PCA, tanto para determinar a orientação, num plano 2D, dos gestos capturados, como para transformar os dados (*pixels* da zona da mão) para um espaço que realce as diferenças (MEF – *Most Expressive Features*, segundo [Swets e Weng, 1996]). Para reduzir os dados com que o classificador tem de lidar, experimentaram várias técnicas para a escolha das características mais discriminantes (MDF – *Most Discriminat Features*, segundo [Swets e Weng, 1996]) e acabaram por escolher um método, chamado de *m-method*, que se baseia no valor relativo da variância das componentes desprezadas, porque dá resultados semelhantes aos outros métodos analisados e é de menor complexidade. A taxa de sucesso foi da ordem dos 99% num conjunto de 25 gestos e 20 imagens de treino para cada gesto. De referir que estes resultados não foram obtidos em tempo-real e utilizaram as imagens de treino também para teste. Uma das conclusões que tiram é que o PCA tem uma performance fraca na fase de treino. Necessita de muitas imagens de treino e é extremamente lento. No entanto em funcionamento corrente consegue desempenhos de tempo-real a 14Hz (desconhecem-se as características do computador).

Histogramas de orientação é outra das técnicas baseadas na aparência e que foi a escolhida, para obtenção de vectores de características por Freeman e Roth [Freeman e Roth, 1995] para o seu trabalho de reconhecimento de gestos a fim de controlar um guindaste virtual ou jogar um jogo. Um dos factores que realçam é a capacidade desta técnica poder ser aplicada em tempo-real e ter alguma imunidade à iluminação.

Outro grupo das técnicas baseadas em aparência são as que têm na base o princípio de *template matching*. A imagem capturada é comparada com imagens padrão de gestos e é classificada na classe do padrão a que mais se assemelha.

Darrell, Essa e Pentland [Darrell, Essa et al., 1996] usam um conjunto de vistas representativas que servem para se confrontar com a imagem capturada, a fim de se

determinar a combinação linear das imagens padrão que melhor representa a imagem capturada. Neste mesmo trabalho, Darrell *et al.* chamam atenção para o facto da correlação normalizada ser menos precisa que metodologias baseadas em espaços próprios, apesar de ser mais adequada para fins de tempo-real.

Abordagem baseada nas características

Uma classe de características que se podem tirar são, por exemplo, os momentos geométricos de imagem, momentos de Hu e momentos de Zernike. Teh e Chin [Teh e Chin, 1988] estudaram momentos geométricos de imagem, momentos de Legendre, momentos de Zernike, pseudo-momentos de Zernike, momentos rotacionais e momentos complexos quanto à capacidade de representação da imagem, sensibilidade ao ruído e redundância de informação. De entre os momentos estudados os melhores resultados foram obtidos pelos momentos de Zernike (definidos por [Teague, 1980]). Uma das características principais dos momentos de Zernike é a sua invariância à rotação. Os momentos de Zernike de duas imagens idênticas, mas com diferentes rotações no plano imagem, apenas diferem num factor de fase, a sua amplitude é semelhante.

Uma versão adaptada dos momentos de Zernike foi utilizada por Schlenzig, Hunter e Jain [Schlenzig, Hunter et al., 1994] no seu trabalho de reconhecimento de gestos para extracção de características da imagem após segmentação.

Além dos momentos existem muitas outras características que se podem extrair de uma imagem, tais como: contorno, descritores de forma, área, perímetro, rectângulo envolvente, elipse ajustada e centróide.

3.1.2.4 Classificador

Os classificadores mais adoptados são os que recorrem a Modelos Escondidos de Markov (HMM – *Hidden Markov Models*). A sua vantagem deve-se ao facto de conseguirem modelizar não só a parte dinâmica de baixo-nível do gesto mas também, em certos gestos, a sua semântica [Wu e Huang, 1999].

Starner e Pentland [Starner, 1995; Starner e Pentland, 1995] foram dos primeiros a utilizar modelos de Markov em linguagem gestual americana (ASL). O seu modelo era constituído por 4 estados, com um de salto de transição para abranger o caso de abreviaturas, e as observações assumidas como gaussianas, independentes e

multidimensionais. Os resultados obtidos foram da ordem de 97% para o caso de frases gramaticalmente correctas.

Pavlovic, Sharma e Huang [Pavlovic, Sharma et al., 1997] indicam que um dos problemas da utilização de HMM prende-se com a assumção, para maior eficiência na parte de treino, que a função de distribuição de probabilidade das observações pode ser modelizada por uma mistura de gaussianas, o que nem sempre é verdade.

Quando é relevante analisar a semântica dos gestos pode-se utilizar uma máquina de estados finita (FSM – *Finite State Machine*). E é isso que adoptam Davis e Shah [Davis e Shah, 1995] para o reconhecimento de um conjunto de gestos simples (semelhantes aos presentes nesta tese) conseguindo uma taxa de sucesso muito próxima de 100% com o sistema a correr a 10Hz numa estação de trabalho SPARC-1.

Bobick e Wilson [Bobick e Wilson, 1997] recorrem a uma abordagem também baseada em estados e utilizam *Dynamic Time Warping* para comparar a sequência de estados capturada com as obtidas na fase de treino.

3.2 ICA - Análise de Componentes Independentes

Nesta secção será apresentada uma descrição sucinta da Análise de Componentes Independentes, dando-se maior enfoque às partes relevantes para o presente trabalho. Pretendendo-se uma descrição mais alargada remete-se o leitor para a consulta dos artigos [Hyvärinen, 1999; Hyvärinen e Oja, 1999b] ou até mesmo do livro [Hyvärinen, Karhunen et al., 2001] que além da descrição da Análise de Componentes Independentes, também explica toda a teoria estatística em que se fundamenta o ICA. Excepção feita às duas formas de aplicar o ICA em reconhecimento de gestos (apresentadas na secção 3.2.4) cuja consulta de [Draper, Baek et al., 2003] e [Bartlett, Lades et al., 1998] se aconselha para o caso da explicação aqui presente não ser suficiente.

3.2.1 Introdução

A Análise de Componentes Independentes (*Independent Component Analysis* – ICA) é uma técnica estatística recente que foi introduzida, no âmbito da modelização de redes neuronais, no início dos anos 80 e que, nos últimos anos, tem merecido a atenção de diversos investigadores um pouco por todo o mundo.

Esta técnica tem como objectivo procurar uma representação de dados que, de uma forma mais evidente, permita desvendar os factores subjacentes a determinados dados estatísticos multivariados. O espaço de dados resultante caracteriza-se por maximizar a independência estatística entre as diversas componentes da nova representação dos dados [Hyvärinen, 1999], o que é uma característica importante do ICA quando comparada com outros métodos como, por exemplo, o PCA (*Principal Component Analysis*) onde as componentes obtidas são apenas descorrelacionadas. A independência estatística é uma condição mais forte do que descorrelação já que, como se irá ver, esta última pode ser vista como um caso particular da primeira. A definição de descorrelação diz que, dadas as variáveis aleatórias $x_1, x_2, \dots, x_n$ estas são descorrelacionadas se verificarem:

$$E\{x_i x_j\} = E\{x_i\}E\{x_j\} \quad \forall_{i \neq j} \quad (3.1)$$

enquanto que independência estatística diz que, para quaisquer funções não lineares, absolutamente integráveis, g e h :

$$E\{g(x_i)h(x_j)\} = E\{g(x_i)\}E\{h(x_j)\} \quad \forall_{i \neq j} \quad (3.2)$$

Atendendo às propriedades da linearidade do valor esperado:

$$\begin{aligned} E\{cx\} &= cE\{x\} \\ E\{x + c\} &= E\{x\} + c, \text{ para } x \text{ e } y \text{ variáveis aleatórias e } c \text{ constante} \\ E\{x + y\} &= E\{x\} + E\{y\} \end{aligned} \quad (3.3)$$

pode-se então dizer que a descorrelação é o caso particular da independência estatística quando g e h são funções lineares. Por conseguinte, a independência estatística, por vezes, também é designada de descorrelação não linear.

A única restrição que o ICA impõe aos dados é que estes sejam não gaussianos, caso contrário inviabilizarão a aplicação do ICA (explicação em 3.2.3). À primeira vista poder-se-á pensar que a exigência da não gaussianidade é uma restrição muito limitativa à aplicação do ICA, mas isso não é verdade. Exigir que os dados sejam gaussianos até é mais restritivo porque em muitos casos não se conhece a distribuição dos mesmos. Assumir-se que os dados seguem uma determinada distribuição, de um conjunto possível delas, é mais restritivo do que assumir que não seguem uma determinada distribuição desse conjunto. E, se a distribuição fosse conhecida, então o problema ficaria bastante mais simples [Hyvärinen, Karhunen et al., 2001]. A não gaussianidade dos dados é um factor de muita importância no ICA porque, além de ser uma exigência, reduzirá a estimação dos parâmetros do modelo a um problema de minimização de uma

função de custo (que estima a gaussianidade dos dados), tal como se irá constatar mais à frente.

3.2.2 Formulação matemática do modelo ICA

O princípio em que assenta o modelo ICA é que os dados observados provêm da mistura dos dados que explicam mais facilmente o problema em análise, a que se dá o nome de fontes (Figura 3.7).

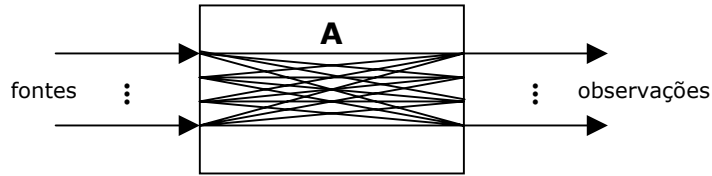


Figura 3.7 – Observações como mistura das fontes

A correspondente representação matricial do modelo ICA ilustrado na Figura 3.7 é expressa pela equação:

$$\mathbf{x} = \mathbf{A}\mathbf{s} \quad (3.4)$$

em que o vector $\mathbf{x}$ representa as observações, o vector $\mathbf{s}$ as fontes e a matriz $\mathbf{A}$ a transformação linear representativa da mistura. Além de fontes, o vector $\mathbf{s}$ também é frequentemente associado a sinais originais, designação que provém da aplicação de ICA em problemas do tipo *cocktail-party*. Neste tipo de problemas tem-se várias fontes sonoras e vários dispositivos de captura de som (vulgo microfones) que registam as observações $\mathbf{x}$, a partir das quais se pretende determinar os sinais originais emitidos pelas fontes sonoras, $\mathbf{s}$.

No modelo expresso pela equação (3.4) não se entra em linha de conta com possível ruído que possa existir. Nos casos em que seja importante incluir a influência do ruído utiliza-se (3.5) em que o vector $\mathbf{n}$ representa o ruído existente.

$$\mathbf{x} = \mathbf{A}\mathbf{s} + \mathbf{n} \quad (3.5)$$

Contudo, neste trabalho, decidiu-se não utilizar o modelo que inclui a influência do ruído porque, além de ser de mais difícil interpretação, iria aumentar significativamente a complexidade do problema, sem grandes garantias de trazer um valor acrescentado relevante, tal como Hyvärinen e Oja antevêem em [Hyvärinen e Oja, 1999b].

O propósito do ICA passa então por determinar a matriz de mistura, $\mathbf{A}$, para se poder estimar os sinais fonte, $\mathbf{s}$, pela equação:

$$\mathbf{s} = \mathbf{W}\mathbf{x} \quad (3.6)$$

onde $\mathbf{W}$ é a matriz inversa de $\mathbf{A}$ (ou pseudo-inversa no caso de esta última não ser quadrada), a que se dá o nome de matriz de separação. Todavia, na prática o que se obtém não é $\mathbf{s}$ mas sim uma sua estimativa, que comumente se designa por $\mathbf{u}$:

$$\mathbf{u} = \mathbf{W}\mathbf{x} \quad (3.7)$$

A Figura 3.8 pretende ilustrar todo o processo descrito: as observações ($\mathbf{x}$) resultam de uma mistura ($\mathbf{A}$) das fontes ($\mathbf{s}$) que são estimadas a partir da matriz de separação ($\mathbf{W}$).

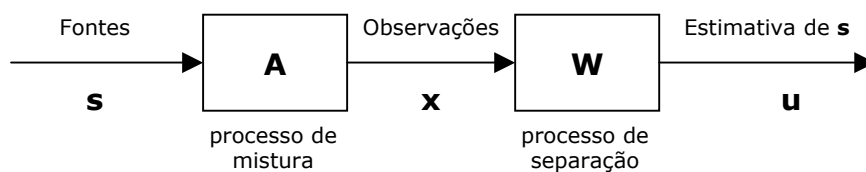


Figura 3.8 – Modelo do processo ICA (adaptado de [Draper, Baek et al., 2003])

3.2.3 Estimação

Na equação (3.4) a única variável conhecida é a das observações, $\mathbf{x}$, e através do ICA pretende-se obter as componentes independentes, $\mathbf{s}$, depois de determinada $\mathbf{A}$. Assim temos 2 incógnitas e apenas uma equação mas curiosamente a assumption das componentes do vector $\mathbf{s}$, s_i , serem estatisticamente independentes, e não serem gaussianas, possibilita a estimação das componentes independentes [Hyvärinen, Karhunen et al., 2001] a menos de um sinal. A estimação dos parâmetros do modelo ICA assenta em dois princípios em que o primeiro provém da independência estatística poder ser vista como descorrelação não linear.

➤ 1º princípio de estimação ICA - **Descorrelação não linear**

Encontrar a matriz $\mathbf{W}$ tal que quaisquer duas diferentes componentes sejam não correlacionadas e as suas transformações, por funções não lineares, também o sejam.

Porém, o ponto essencial da estimação ICA é a assumption/exigência dos dados não serem gaussianos. Caso os dados sejam gaussianos é impossível estimar as componentes independentes devido ao facto de transformações ortogonais de dados gaussianos manterem a mesma distribuição. O que origina que a informação da

independência estatística se torne irrelevante inviabilizando a estimação das componentes independentes [Hyvärinen, Karhunen et al., 2001], porque em dados gaussianos componentes não correlacionadas são sempre estatisticamente independentes.

Com base no Teorema do Limite Central, que diz que, sob certas condições, a distribuição da soma de variáveis aleatórias independentes tende para uma distribuição gaussiana, a determinação das componentes independentes transforma-se num problema de maximização de uma função que traduz o grau de não gaussianidade. Apesar de também existir uma fundamentação teórica rigorosa, isto é explicado heurísticamente da seguinte forma: sendo a componente s_i uma combinação linear de $\mathbf{x}$ (com os pesos dados por uma linha da matriz de separação $\mathbf{W}$ que se pretende determinar) então, pelo Teorema do Limite Central, será mais gaussiana do que o próprio x_i , visto que a soma (combinação linear) de variáveis aleatórias independentes tem uma distribuição mais próxima da gaussiana do que as respectivas distribuições originais. Então determinando-se essa linha de $\mathbf{W}$ de forma a maximizar a não gaussianidade de s_i obtém-se uma componente independente, pelo facto de, sendo “pouco” gaussiana então estar-se próximo da independência! De seguida, estimam-se as outras componentes através da localização do máximo local destas funções de custo, restringindo-se a procura ao espaço que dá estimativas não correlacionadas das componentes já determinadas, o que é expresso pelo 2º princípio de estimação:

➤ 2º princípio de estimação ICA – **Maximização da não gaussianidade**

Encontrar o máximo local da não gaussianidade de uma combinação linear das observações com variância constante.

Resta então definir a função, medida de não gaussianidade, que se pretende maximizar. A curtose (3.8), ou cumulante de 4ª ordem, é desde logo uma candidata pois é nula para variáveis aleatórias gaussianas e diferente de zero para grande parte das não gaussianas.

$$\text{kurt}(y) = E\{y^4\} - 3(E\{y^2\})^2 \quad (3.8)$$

No entanto, a curtose é uma medida de não gaussianidade pouco robusta porque é bastante sensível a observações díspares (*outliers*). Assim, surgem outras medidas de não gaussianidade, como é o caso da neguentropia (*negentropy*) que se fundamenta na entropia.

A entropia de uma variável aleatória é definida como uma medida da quantidade de informação que ela contém. Assim, uma variável pouco previsível conterá muita informação, visto não ser de fácil predição, logo a sua entropia é alta. Enquanto que uma variável aleatória previsível terá baixa entropia, visto ser fácil prever o seu valor. Como a entropia de uma variável aleatória gaussiana é mais elevada do que qualquer outra variável aleatória com a mesma variância (provado em [Cover e Thomas, 1991]), a entropia pode ser usada como medida de não gaussianidade. Contudo, em vez de se aplicar directamente a entropia usa-se a neguentropia (3.9), J , de forma a ter-se uma função que toma valores positivos para variáveis não gaussianas e zero para gaussianas:

$$J(y) = H(y_{\text{gauss}}) - H(y) \quad (3.9)$$

onde $H(y)$ representa a entropia de y e $H(y_{\text{gauss}})$ a entropia de uma variável aleatória gaussiana com a mesma covariância de y , sendo a entropia de uma variável aleatória contínua dada por:

$$H(y) = -\int f(y) \log[f(y)] dy \quad (3.10)$$

Apesar da neguentropia ser um bom estimador da não gaussianidade, tem o inconveniente de ser computacionalmente difícil de obter. Por esta razão, existem algumas aproximações à neguentropia para torná-la computacionalmente mais viável, das quais se destaca (3.11) que foi proposta em [Hyvärinen, 1997].

$$J(y) \approx \sum_{i=1}^p k_i [E\{G_i(y)\} - E\{G_i(v)\}]^2 \quad (3.11)$$

onde k_i são constantes positivas, v uma variável aleatória gaussiana de média zero e variância 1 e G_i funções de ordem superior à quadrática. No caso de se usar apenas uma função de ordem superior à quadrática (3.11) vem

$$J(y) \propto [E\{G(y)\} - E\{G(v)\}]^2 \quad (3.12)$$

Fazendo-se uma boa escolha da função G , de modo a que não cresça rapidamente, obtém-se um bom estimador da neguentropia. Neste trabalho a escolha recaiu na função G expressa em (3.13), o que significa que a função de custo g , obtida da derivação de G , é dada por (3.14). Esta escolha é justificada pela afirmação de Hyvärinen, Karhunen e Oja em [Hyvärinen, Karhunen et al., 2001] de que a tangente hiperbólica ($\tanh$) é adequada na utilização de FastICA.

$$G(u) = \log \cosh(u) \quad (3.13)$$

$$g(u) = \tanh(u) \quad (3.14)$$

Definida a função de custo a utilizar resta agora escolher qual o algoritmo de maximização a utilizar. Neste trabalho optou-se por utilizar o FastICA que é apresentado em [Hyvärinen e Oja, 1997; Hyvärinen e Oja, 1999a]. A escolha deve-se à disponibilidade do FastICA (<http://www.cis.hut.fi/projects/ica/fastica/>) e à sua velocidade de convergência. Contudo, a utilização de outros algoritmos, como por exemplo o Infomax, não garante melhores resultados e, no caso de o serem, não serão muito melhores e também dependem da arquitectura utilizada (apresentada em 3.2.4), tal como Draper et al. concluíram em [Draper, Baek et al., 2003].

O FastICA é um algoritmo de ponto fixo que procura a direcção que maximiza a não gaussianidade da componente a estimar, s_i , e caracteriza-se por [Hyvärinen e Oja, 1999b]:

- apresentar convergência cúbica, ou na pior das hipóteses quadrática, ao contrário dos métodos da descida do gradiente (*gradient descent*), como é o Infomax [Bell e Sejnowski, 1995], que têm uma convergência apenas linear, logo mais lenta.
- não necessitar da especificação do passo de iteração (isto é, o incremento que há ao passar para uma nova iteração) o que o torna mais fácil de usar quando comparado com os algoritmos de gradiente.
- conseguir estimar as componentes independentes de praticamente qualquer distribuição não gaussiana, independentemente da não linearidade g (utilizada indirectamente em (3.12)), ao contrário de outros algoritmos (caso do Infomax) que precisam de escolher uma não linearidade de acordo com a distribuição em causa.
- permitir determinar as componentes independentes uma a uma, o que pode ser útil nos casos em que não se pretende obter todas as componentes.

Um pré-processamento dos dados que é comum fazer-se antes de se lhe aplicar o FastICA é centralizar e de seguida branquear porque, apesar de dispensável, melhora o desempenho do FastICA.

A centralização dos dados corresponde a subtrair-lhes o seu vector médio $\mathbf{m}$, equação (3.15), por forma a tornar $\mathbf{x}$ numa variável de média nula.

$$\mathbf{m} = E\{\mathbf{x}\} \quad (3.15)$$

Quanto ao branqueamento, também conhecido por esferização quando a variável aleatória tem dimensão superior a 2, a vantagem é tornar a matriz $\mathbf{A}$ ortogonal por forma

a reduzir o número de parâmetros a estimar. Assumindo-se $\mathbf{A}$ como uma matriz quadrada de dimensão n , em vez de se ter n^2 parâmetros a estimar tem-se apenas $n(n-1)/2$ [Hyvärinen e Oja, 1999b].

Uma forma de realizar o branqueamento dos dados consiste em aplicar a equação

$$\mathbf{z} = \mathbf{E}\mathbf{D}^{-\frac{1}{2}}\mathbf{E}^T\mathbf{x} \quad (3.16)$$

em que $\mathbf{z}$ representa o vector $\mathbf{x}$ branqueado, $\mathbf{E}$ a matriz ortogonal constituída pelos vectores próprios da matriz de covariância, $\mathbf{C}_x$, e $\mathbf{D}$ a matriz diagonal com os valores próprios correspondentes. A obtenção da matriz de covariância de dados, no caso de estarem centralizados, é dada pela equação (3.17).

$$\mathbf{C}_x = E\{\mathbf{x}\mathbf{x}^T\} \quad (3.17)$$

O branqueamento tem o efeito de tornar a matriz de covariância igual à matriz identidade, (3.18), o que significa que as diferentes componentes além de descorrelacionadas têm variância unitária.

$$E\{\mathbf{z}\mathbf{z}^T\} = \mathbf{I} \quad (3.18)$$

3.2.4 Redução da dimensionalidade dos dados

O número máximo de componentes independentes que se consegue estimar por ICA corresponde ao número de observações existentes (*i.e.* número de variáveis aleatórias, logo a dimensão do vector $\mathbf{x}$). Podem-se estimar menos componentes independentes do que as observações existentes mas mais componentes do que observações não é possível.

Decidindo-se obter menos componentes independentes do que o número de observações surge a questão de quais estimar, definindo quais as componentes mais importantes. O que desvenda uma das características da Análise de Componentes Independentes: o ICA não é uma técnica vocacionada para redução de dimensionalidade de dados porque não tem implícita nenhuma ordenação das componentes independentes, ao contrário do PCA (*Principal Component Analysis*) cujas componentes são ordenadas segundo a variância. Em PCA, quanto maior for a variância de uma componente mais importante ela é, assim, componentes de baixa variância podem ser descartadas por forma a reduzir a dimensão dos dados. Isto está assente no pressuposto de que a variância de uma componente é proporcional à contribuição que teve no valor da observação, logo eliminando componentes de baixa variância não se está a suprimir informação relevante

e até pode ocorrer o caso de se estar a remover ruído indesejável que possa existir nos dados.

Contudo vários autores têm-se dedicado a procurar técnicas que permitam a ordenação/selecção das componentes independentes das quais se destacam, a ordenação com base na norma da correspondente coluna da matriz de mistura e a ordenação de acordo com o valor da medida de não gaussianidade dessa componente.

Mas, pelo que se conhece, das técnicas exploradas para ordenação das componentes, nenhuma tem um fundamento teórico preciso e experimentalmente ainda não foi provado que dão bons resultados independentemente da aplicação em causa. A ordenação das componentes independentes é assim um problema ainda em aberto!

Há no entanto situações em que na utilização do ICA é importante reduzir o número de componentes independentes estimadas quer por motivos de rapidez de processamento quer até por insuficiência de memória disponível. Assim, é comum aplicar-se uma redução da dimensionalidade dos dados por PCA antes de aplicar o ICA, o que terá o efeito de diminuir o número de componentes independentes. Um facto importante é que a aplicação do PCA não inviabiliza a aplicação do ICA. Como o PCA consiste numa combinação linear dos dados (segundo a variância), as relações de ordem superior a 2 continuam a existir nos dados, apenas as de 2ª ordem são removidas.

Uma vantagem de aplicar PCA antes de ICA prende-se com o facto de melhorar a performance do ICA porque põe de parte variáveis aleatórias de baixa variância antes do branqueamento e também porque ao remover as dependências de 2ª ordem reduz a complexidade computacional.

3.2.5 Aplicação de ICA em reconhecimento de gestos

Como na aplicação de ICA se assume que as observações são constituídas por um vector, logo 1D, e as imagens de gestos capturadas são 2D, é necessário fazer uma concatenação coluna a coluna por forma a transformá-las num vector unidimensional, de dimensão igual ao número de *pixels* total, a fim de se empregar o ICA. A escolha foi de executar a concatenação coluna a coluna mas também se poderia ter feito linha a linha. Assim designar-se-á por matriz imagem a imagem capturada 2D e por vector imagem o correspondente vector 1D.

Uma vez obtido o vector imagem há duas formas distintas de se aplicar o ICA:

- encarar as imagens como variáveis aleatórias e os *pixels* como observações

- encarar os *pixels* como variáveis aleatórias e as imagens como observações

Dependendo da alternativa escolhida ter-se-á as imagens como vectores linha da matriz $\mathbf{A}$, e consequentemente da $\mathbf{W}$, ou como vectores coluna dessas matrizes consoante se use a primeira alternativa ou a segunda, respectivamente.

A estas alternativas é comum chamar-se de “Arquitectura I” e “Arquitectura II”, designações que vêm de aplicação de ICA em áreas de reconhecimento visual de faces e expressões faciais.

Nas duas secções seguintes vai-se utilizar as letras maiúsculas $\mathbf{X}$, $\mathbf{U}$ e $\mathbf{S}$ para representar $\mathbf{x}$, $\mathbf{u}$, e $\mathbf{s}$ que foram expressos em letras minúsculas na secção 3.2.2. Isto deve-se ao facto de agora $\mathbf{X}$, e consequentemente $\mathbf{U}$ e $\mathbf{S}$, serem matrizes cujas linhas ou colunas são constituídas pelos vectores imagem referenciados em 3.2.2.

3.2.5.1 Arquitectura I – Base de imagens estatisticamente independentes

De acordo com esta arquitectura, as imagens são encaradas como variáveis aleatórias e os *pixels* como observações o que origina que, através do ICA se obtenha um conjunto de imagens base estatisticamente independentes, isto é, as imagens resultantes das observações, $\mathbf{X}$, são interpretadas como a mistura linear, $\mathbf{A}$, de um conjunto de imagens estatisticamente independentes, $\mathbf{S}$, que constituem a base de imagens. Por imagens estatisticamente independentes entende-se que não é possível prever os valores dos *pixels* de uma imagem da base a partir dos *pixels* das outras imagens constituintes da base.

Na Figura 3.9 mostra-se 8 imagens de uma base de imagens obtida pela Arquitectura I.

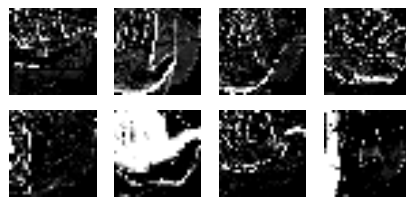


Figura 3.9 – Imagens da base resultante da Arquitectura I

Nesta arquitectura as imagens observações são dispostas nas linhas de $\mathbf{X}$ e as imagens da base encontram-se nas linhas de $\mathbf{U}$, que é obtida por $\mathbf{U}=\mathbf{W}\mathbf{X}$, sendo a representação ICA de uma dada imagem, $\mathbf{x}_i$, dada pelo conjunto de coeficientes da mistura (Figura 3.10). Isto é, os valores das componentes que definem o gesto $\mathbf{x}_i$ no espaço ICA estão no vector $\mathbf{b}_i=(a_{i1}, a_{i2}, \dots, a_{in})^T$ em que os a_{i*} são os coeficientes retirados da linha i da matriz de mistura $\mathbf{A}$.

$$\mathbf{x}_i = a_{i1} \times \begin{matrix} \mathbf{u}_1 \\ \text{img} \end{matrix} + a_{i2} \times \begin{matrix} \mathbf{u}_2 \\ \text{img} \end{matrix} + \dots + a_{in} \times \begin{matrix} \mathbf{u}_n \\ \text{img} \end{matrix}$$

Figura 3.10 – Representação ICA do gesto $\mathbf{x}_i$ pela Arquitectura I ($a_{i1}, a_{i2}, \dots, a_{in}$)

3.2.5.2 Arquitectura II – Coeficientes estatisticamente independentes

Na Arquitectura II, as imagens são entendidas como observações e os *pixels* como variáveis aleatórias, o que resulta na obtenção de um conjunto de coeficientes de mistura estatisticamente independentes, ao passo que, na Arquitectura I se obtinham imagens estatisticamente independentes. Pela Arquitectura I obtém-se um conjunto de imagens base estatisticamente independentes mas os coeficientes da mistura não o são. Na Arquitectura II verifica-se o contrário, os coeficientes são estatisticamente independentes mas as imagens da base não. Na Figura 3.11 tem-se uma amostra de imagens da base de imagens do espaço ICA.



Figura 3.11 – Imagens da base resultante da Arquitectura II

De um ponto de vista algébrico, a diferença da Arquitectura I e II traduz-se pela transposição da matriz $\mathbf{X}$, que contém os vectores imagem; assim, na Arquitectura II as imagens são dispostas nas colunas de $\mathbf{X}$ e, portanto, as colunas de $\mathbf{U}$ dadas por $\mathbf{U}=\mathbf{W}\mathbf{X}$, contêm a representação ICA das imagens observações, presentes em $\mathbf{X}$. Na Figura 3.12 ilustra-se a mistura de imagens que gera o gesto $\mathbf{x}_i$.

$$\mathbf{x}_i = \begin{matrix} \mathbf{a}_1 \\ \text{img} \end{matrix} \times u_1 + \begin{matrix} \mathbf{a}_2 \\ \text{img} \end{matrix} \times u_2 + \dots + \begin{matrix} \mathbf{a}_n \\ \text{img} \end{matrix} \times u_n$$

Figura 3.12 – Representação ICA do gesto $\mathbf{x}_i$ pela Arquitectura II ($u_1, u_2, \dots, u_n$)

As imagens $\mathbf{a}^*$ da Figura 3.12 são oriundas das colunas da matriz $\mathbf{A}$, que representam a base de imagens (estatisticamente dependente) do espaço ICA. E a representação ICA do gesto é dada pelos coeficientes de uma coluna de $\mathbf{U}$, que se designou de $\mathbf{u}=(u_1, u_2, \dots, u_n)$.

Como as imagens que geram o espaço ICA (linhas de U na Architectura I e colunas de A na Architectura II) podem ser interpretadas como imagens características (por serem um vector de características), pela Architectura I obtêm-se características locais enquanto que na Architectura II características globais, na medida em que pela Architectura I as componentes ICA de uma imagem são influenciadas por pequenas porções da imagem enquanto que na Architectura II a influência já é de toda a imagem, ou seja, cada componente é influenciada por todos os *pixels* [Draper, Baek et al., 2003]. Isto também pode ser confirmado pela comparação das imagens de base das duas architectures (Figura 3.9 e Figura 3.11); pela Architectura II tem-se imagens globalmente bem definidas onde se vêem os gestos, enquanto que as da Architectura I realçam apenas pequenos pormenores da imagem o que, por vezes, visualmente não tem grande significado aparente.

Assim, dado que os gestos têm configurações bem distintas, de acordo com a classe que representam, a análise de pequenas partes do mesmo pode revelar a sua classe. Por isso, uma metodologia que se foque principalmente nas características locais tenderá a dar melhores resultados em reconhecimento de gestos. Por esta razão, a Architectura I é mais vocacionada para reconhecimento de gestos do que a Architectura II [Draper, Baek et al., 2003].

CAPÍTULO 4

Abordagem adoptada

No desenvolvimento do pretendido sistema reconhecedor de gestos simbólicos, baseado em visão por computador, foi necessário executar várias tarefas distintas e complementares que se vão agora descrever com algum pormenor. A descrição terá uma índole prática, em oposição aos capítulos precedentes, por forma a tornar mais claro como se passou da teoria para a implementação computacional e dessa forma auxiliar quem pretender reproduzir os resultados obtidos no decorrer do trabalho aqui descrito. No entanto, sempre que se entender adequado, será explanada também a teoria de base dos métodos que ainda não tenham sido descritos.

A primeira noção importante a reter é que o sistema terá dois modos distintos de funcionamento, a que iremos chamar: treino e corrente (ou teste).

O treino consiste numa fase de aprendizagem com o objectivo de encontrar a matriz de transformação que permite levar do espaço de dados original para o espaço ICA. Na Figura 4.1 é apresentado o diagrama de blocos deste modo de funcionamento. As imagens são capturadas por uma câmara a cores apontada sobre um fundo preto estando a mão equipada com uma luva de cor branca. Posteriormente as imagens capturadas são submetidas a uma segmentação para se extrair a parte da imagem que contém a mão que, depois de normalizada, servirá para treinar o modelo ICA com o fim de se obter as imagens de base que definem o espaço ICA.

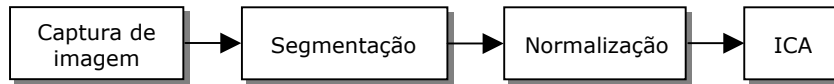


Figura 4.1 – Diagrama de blocos do reconhecedor em modo de treino

Em funcionamento corrente, Figura 4.2, os 3 primeiros blocos são exactamente iguais aos de funcionamento em modo de treino, a diferença reside nos dois últimos blocos. Enquanto que o bloco ICA da Figura 4.1 tem a responsabilidade de encontrar a matriz de transformação de base (pseudo-inversa de $\mathbf{U}$ ou $\mathbf{W}$ no caso de se utilizar a Arquitectura I ou a Arquitectura II, respectivamente), o bloco Transformação ICA da Figura 4.2 realiza essa transformação para que a classificação das imagens capturadas se faça no espaço ICA que se presume ter uma maior capacidade de discriminação.

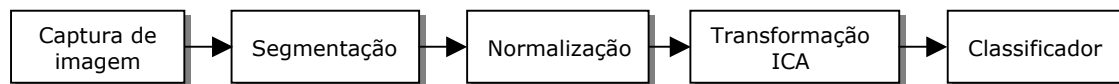


Figura 4.2 – Diagrama de blocos do reconhecedor em modo corrente

Como os diagramas de blocos em modo de treino e modo corrente são semelhantes e atendendo ao facto de até 3 deles serem iguais, a descrição de cada um dos blocos será feita ao mesmo nível independentemente do modo de funcionamento, com excepção do bloco correspondente à utilização da técnica ICA, cuja funcionalidade difere consoante se esteja em treino ou teste.

4.1 Segmentação

Ao bloco de Segmentação cabem as responsabilidades de, dada uma determinada imagem capturada, localizar a mão e detectar a sua ausência. A importância da capacidade de detecção da ausência da mão prende-se com o rastreio inicial que, além de poder evitar possíveis erros de classificação, impede o desperdício de recursos computacionais a processar algo que já se sabe não ter interesse.

Para esse fim é aplicada então uma operação local de *thresholding* à imagem colorida cuja representação de cor segue o modelo RGB (*Red, Green, Blue*) digitalizado com 8 bits por componente. Assim, todos os *pixels* cujas componentes RGB sejam em média superiores a um determinado limiar de *threshold* e tenham um desvio padrão inferior a 50 serão considerados *pixels* de interesse, ou dito de outra forma, os *pixels* de interesse serão os *pixels* claros e aproximadamente brancos. “*Pixels* de interesse”, ou “pontos de

interesse” é a denominação que vai ser seguida neste relatório para referir os *pixels* que se julga pertencerem à mão.

Quanto ao valor do limiar de *thresholding* este deve ser determinado numa fase inicial de calibração do sistema por forma a fazer uma escolha adequada de acordo com as condições de iluminação do cenário de captura. A utilização da condição de desvio padrão inferior a 50 serve para restringir os *pixels* de interesse aos próximos de branco (cor da luva que equipa a mão) e não apenas aos *pixels* que têm componentes de cor elevadas. O efeito da aplicação desta condição pode ser visto nos cubos de cores RGB apresentados na Figura 4.3 onde se representa os pontos cujas componentes RGB satisfazem as condições de *thresholding*. Na imagem da esquerda apenas se aplica o limiar de *threshold* e, por isso, os pontos de interesse descrevem uma pirâmide triangular no cubo RGB. Ao passo que na imagem da direita se mostra o caso em que, além do limiar de *threshold*, também se tem em conta a condição do desvio padrão e, por isso, os pontos de interesse são em menor número e agora descrevem aproximadamente um cilindro dentro do cubo RGB.

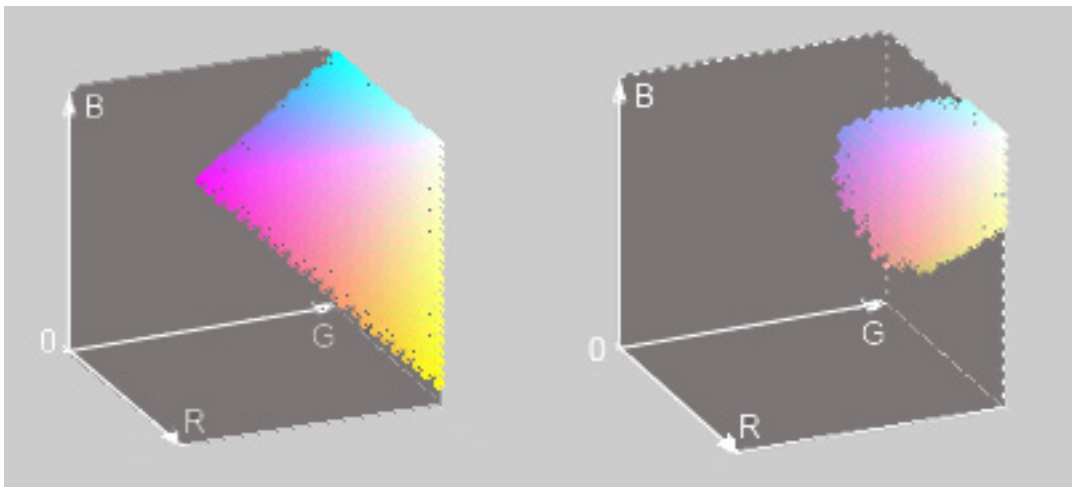


Figura 4.3 – Pontos de interesse do cubo de cores RGB (sem e com condição de desvio padrão, respectivamente, esquerda e direita)

Por vezes, a segmentação baseada apenas em *thresholding* não é suficiente porque muito frequentemente a pessoa que está a executar o gesto, embora de uma forma parcial, aparece no ângulo de visão da câmara de captura. Há também outros casos em que por deficiências de iluminação ou imperfeições do fundo da cena capturada (relembra-se que tal como se especificou na secção 1.3, o fundo deve ser preto) certos *pixels* são considerados de interesse pelo *thresholding* embora não pertençam à mão.

Assim, para aumentar a robustez neste cenário de ambiente controlado, decidiu-se recorrer à informação dada pelas projecções vertical e horizontal da imagem após o *thresholding*. A projecção horizontal e a projecção vertical consistem em realizar a contagem dos pontos brancos em, respectivamente, cada coluna e linha da imagem binária (a imagem resultante de *thresholding*). Na Figura 4.4 pode-se ver um exemplo da projecção vertical e da projecção horizontal de uma imagem binária de um gesto.

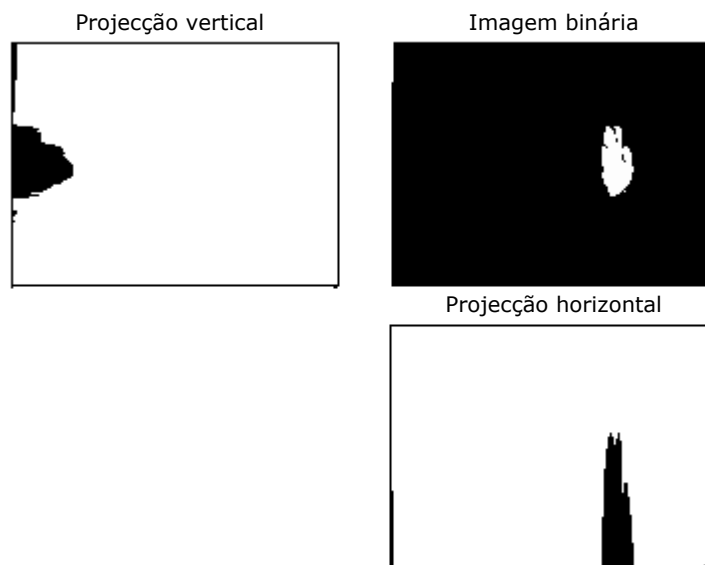


Figura 4.4 – Projecção vertical e horizontal de uma imagem binária

Neste trabalho optou-se por usar apenas a informação dada pela projecção horizontal (explicada mais à frente) visto ter sido suficiente para todas as imagens de teste; no entanto, em casos mais difíceis, poder-se-á também usar a projecção vertical. A opção por apenas uma das projecções tem o propósito de minimizar o processamento necessário a fim de facilitar a utilização desta metodologia em tempo-real. Por isso, sempre que há margem para não executar algum processamento específico, ele não é executado. Por esta mesma razão, a projecção horizontal só é usada nos casos em que se suspeita que nem todos os *pixels* considerados de interesse pertencem à mão. A detecção da necessidade de se utilizar a projecção horizontal é dada pela ocupação

espacial desalinhada¹, $O_{\text{desalinhada}}$ (4.1), que consiste no quociente do número de pontos de interesse pela área do seu rectângulo envolvente.

$$O_{\text{desalinhada}} = \frac{\text{número de pontos de interesse}}{\text{área do rectângulo envolvente}} \quad (4.1)$$

Na Figura 4.5 exemplifica-se o valor deste quociente para o caso de mãos bem segmentadas.



Figura 4.5 – Ocupação desalinhada de diferentes gestos

Se este quociente for inferior a um determinado limiar então é melhor confirmar se todos os *pixels* pertencem à mão, recorrendo-se então à análise da projecção horizontal. O valor desse limiar deve ser escolhido de tal forma que, independentemente da orientação com que é capturada, em mãos bem segmentadas não haja a necessidade de se utilizar a projecção, e também que consiga afirmar com grande certeza que o que foi segmentado ou não é gesto ou tem algo mais que o gesto. Assim, neste trabalho utilizou-se o valor 0,25 para esse limiar. Nos casos em que $O_{\text{desalinhada}}$ continua inferior a 0,25 depois de se utilizar a informação da projecção, a imagem é descartada, para impedir que chegue aos blocos de processamento seguintes, já que se entende não ser possível encontrar lá um gesto.

Na Figura 4.6 pode-se ver o rectângulo envolvente de duas segmentações distintas da mesma imagem. Do lado esquerdo está o caso em que o *thresholding* é insuficiente para segmentar correctamente. No caso do lado direito, foi utilizada também a informação dada pela projecção horizontal, o que permitiu uma segmentação correcta.

¹ A denominação “desalinhada” é para salientar o facto de não se tratar do rectângulo envolvente alinhado mas sim do menor rectângulo, de lados paralelos às margens da imagem, que consegue englobar todos os pontos de interesse.

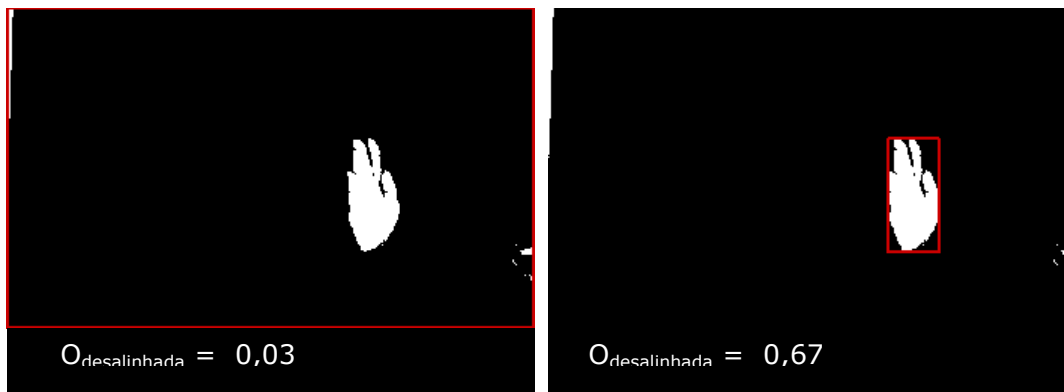


Figura 4.6 – Segmentação sem e com a informação da projecção horizontal

A metodologia utilizada para retirar informação da projecção horizontal consiste em percorrê-la a fim de encontrar a zona mais larga com contagens superiores a um determinado limiar (não se usou o valor zero para não ficar sensível a ruído do *thresholding*). Uma vez determinada essa zona fica definida uma faixa vertical que restringe os *pixels* mão apenas aos pontos de interesse (resultado do *thresholding*) que se encontram aqui localizados.

De referir que o algoritmo de localização da zona mais larga não pode procurar apenas por contagens sucessivas superiores a um determinado limiar. É necessário incluir alguma flexibilidade/tolerância para aumentar a qualidade do seu resultado. Assim, este algoritmo foi implementado com dois critérios de tolerância, um exigente e um brando, por forma a aumentar a robustez. Na imagem central da Figura 4.7 tem-se um caso em que a sombra causada pelo dedo (imagem da esquerda) origina uma má segmentação (imagem central) mas que, através da flexibilização do algoritmo de localização da zona mais larga, já se consegue uma segmentação correcta (imagem da direita).



Figura 4.7 – Melhoria dada pela flexibilização do algoritmo de localização da zona de projecção horizontal mais larga

Há que referir que neste caso específico poder-se-ia tentar resolver o problema por diminuição do limiar de *thresholding*; contudo, essa solução poderá não ser a mais indicada visto poder causar outros problemas.

4.2 Normalização

A saída do bloco de Segmentação é uma imagem em tons de cinzento constituída apenas pelo excerto da imagem capturada onde se encontra localizada a mão (Figura 4.9). A conversão de cor para níveis de cinza é efectuada pela média aritmética das componentes RGB da imagem e, como a imagem é codificada com 8 bits/*pixel*, os *pixels* assumem valores entre 0 e 255, em que 0 representa preto e 255 branco.



Figura 4.8 – Imagem original



Figura 4.9 – Imagem segmentada

A função do bloco de Normalização é analisar e processar a imagem segmentada para gerar uma outra com dimensões e orientação pré-estabelecidas por forma a que diferentes imagens capturadas sejam apresentadas ao bloco seguinte, o ICA (quer se esteja em modo de treino ou corrente), aproximadamente nas mesmas condições. A obrigatoriedade de todas as imagens serem entregues ao ICA com as mesmas dimensões deve-se ao facto dos dados multivariados (representados pela imagem) não poderem mudar de dimensão ao longo do tempo. As observações têm de ser sempre das mesmas variáveis aleatórias. Relativamente à correcção da orientação, isto deve-se a ser praticamente impossível as pessoas realizarem os gestos sempre com a orientação com que foram definidos (secção 1.3). É então importante fazer uma pequena correcção angular. Caso o gesto executado tenha uma orientação bastante diferente da especificada, ele não é considerado válido e, por isso, será provavelmente classificado como desconhecido.

A determinação da orientação com que o gesto se apresenta ao sistema é definida pela direcção do eixo de menor inércia da mão que é obtido através dos momentos centrais (momentos de 2ª ordem) da imagem segmentada. Os momentos centrais de 2ª ordem, de uma imagem binária, são definidos por:

$$\mu_{pq} = \frac{1}{N} \sum_x \sum_y (x - \bar{x})^p (y - \bar{y})^q \quad (4.2)$$

onde o somatório é efectuado apenas nos pontos brancos, (x,y) são as coordenadas espaciais desses pontos, N representa o número total de pontos brancos e $\bar{x}$ e $\bar{y}$ as coordenadas do centróide.

$$\bar{x} = \frac{1}{N} \sum_x x \quad (4.3)$$

$$\bar{y} = \frac{1}{N} \sum_y y \quad (4.4)$$

Assim, o ângulo formado entre a horizontal e o eixo de menor inércia da mão será dado por (4.5) se $\mu_{02} > \mu_{20}$ ou em caso contrário por (4.6). A definição detalhada de como se chega a estas fórmulas pode ser encontrada em [Haralick e Shapiro, 1992].

$$\theta = \arctg \left(\frac{\mu_{02} - \mu_{20} + \sqrt{(\mu_{02} - \mu_{20})^2 + 4\mu_{11}^2}}{2\mu_{11}} \right) \quad (4.5)$$

$$\theta = \arctg \left(\frac{2\mu_{11}}{\mu_{20} - \mu_{02} + \sqrt{(\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2}} \right) \quad (4.6)$$

Para se saber qual a orientação do gesto capturado utiliza-se não a imagem segmentada (em tons de cinzento) mas sim uma sua versão binarizada (só pontos pretos e brancos) que, por razões de eficiência computacional, é a imagem binária após o *thresholding* do bloco de Segmentação (Figura 4.10). Uma vez encontrado o ângulo que o eixo de menor inércia faz com a horizontal (Figura 4.11), θ , executa-se uma rotação em sentido contrário de forma a alinhar o gesto.



Figura 4.10 – Binarização da imagem segmentada



Figura 4.11 – Direcção do eixo de menor inércia

Como apenas se pretende dar um pequeno ajuste ao gesto, visto que a orientação com que ele é executado é importante para a sua interpretação, a rotação aplicada é a menor possível para alinhar o gesto com o eixo vertical ou horizontal (ver Figura 4.12).

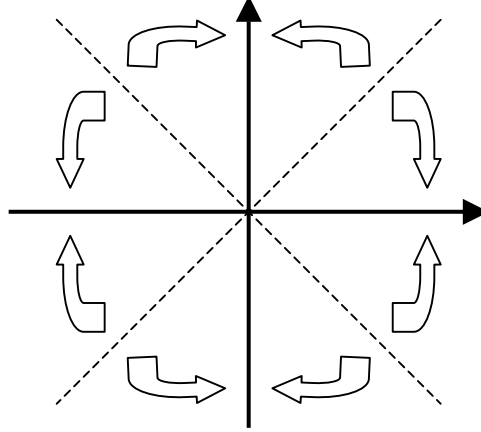


Figura 4.12 – Direcções para onde são rodadas as imagens

A aplicação da rotação traduz-se numa transformação geométrica dos *pixels* de acordo com a equação (4.7) onde θ representa o ângulo de rotação e (x',y') as coordenadas destino do *pixel* que originalmente tinha coordenadas (x,y) .

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \quad (4.7)$$

Na prática não se utiliza a transformação directa, equação (4.7), mas sim a transformação inversa, o que significa que se percorrem os *pixels* da imagem destino e por (4.8) obtém-se cada um dos *pixels* correspondente da imagem original.

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}^{-1} \begin{bmatrix} x' \\ y' \end{bmatrix} \quad (4.8)$$

Como as coordenadas dos *pixels* são números inteiros e as coordenadas origem dadas por (4.8) muito provavelmente resultarão em números de parte fraccionária não nula, é necessário efectuar uma interpolação de brilhos. Por motivos de eficiência computacional, e porque para o problema em questão é suficiente, decidiu-se utilizar a interpolação do vizinho mais próximo que se traduz por um arredondamento das coordenadas ao inteiro mais próximo. Deste modo, um pixel de coordenadas (x',y') herdará o brilho do *pixel* de coordenadas (x_a,y_a) da imagem original, onde os índices “a” referem tratar-se de valores arredondados.

Feita a rotação há que obter o novo rectângulo envolvente da mão (Figura 4.13) por forma a proceder-se à fase seguinte de redimensionamento para dimensões predefinidas.



Figura 4.13 – Imagem rodada

O redimensionamento será tal que aumente (ou diminua), na mesma proporção, as dimensões laterais do rectângulo envolvente da mão até alcançar ou o limite de largura ou altura predefinidos. Após este redimensionamento será necessário realizar um enchimento (*padding*), ou na parte de baixo ou na da direita da imagem, a fim de se ter uma imagem não deformada da mão com as dimensões predefinidas. Na Figura 4.14, apenas para fins ilustrativos, apresenta-se a cinza o enchimento que foi necessário fazer a imagens de alguns gestos; porém, na realidade o enchimento é realizado com *pixels* pretos de forma a simular o fundo da cena.

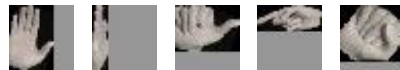


Figura 4.14 – Enchimento necessário após redimensionar as imagens

Na Figura 4.15 apresentam-se imagens normalizadas dos 5 gestos que se pretende reconhecer e é possível verificar que nem todos os gestos normalizados de uma mesma classe ficam com a mesma orientação. Isto deve-se à diferença da forma das mãos entre gesticuladores e/ou ao facto da existência de sombras entre os dedos, que causa diferentes imagens binárias e consequentemente variações na direcção do eixo de menor inércia.



Figura 4.15 – Gestos normalizados

Portanto, o sistema reconhecedor terá de prever estes casos a fim de lhes dar o melhor tratamento. Uma via poderá até passar por utilizar uma imagem de referência por cada orientação possível do gesto.

4.3 ICA

Após o bloco de Normalização seguir-se-á um bloco ICA que tem a finalidade de determinar a matriz de transformação ou transformar os dados para o espaço gerado pelas componentes independentes, consoante se esteja em treino ou teste, respectivamente. Como a técnica ICA exige que os dados multivariados estejam representados por um vector, é necessário transformar a matriz imagem (isto é, a matriz que contém os níveis de cinzento de cada *pixel*) num vector. Assim, faz-se um

varrimento coluna a coluna, concatenando-as umas às outras a fim de se transformar a matriz num vector (Figura 4.16). Em detrimento da concatenação de colunas poder-se-ia ter escolhido antes realizar uma concatenação de linhas. A opção pela concatenação de colunas deve-se ao facto de se utilizar o Matlab na obtenção dos resultados apresentados no capítulo seguinte. Devido à forma como o Matlab armazena os dados em memória, neste particular ambiente de desenvolvimento, a concatenação das colunas é mais eficiente que a das linhas.

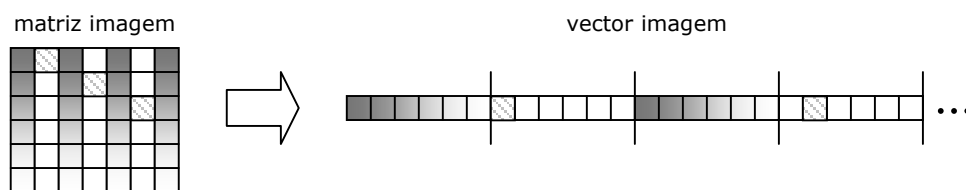


Figura 4.16 – Transformação da matriz imagem em vector imagem

Tendo-se o vector imagem pode-se aplicar o ICA segundo duas arquitecturas distintas, tal como já foi abordado em 3.2.4. Todavia, dependendo dos dados de entrada poderemos não ter a oportunidade de escolher a arquitectura devido ao seu impacto computacional. Vejamos o caso de ter n imagens de treino de dimensão m (correspondente ao produto da largura pela altura da imagem). Segundo a Arquitectura I as imagens são dispostas nas linhas de $\mathbf{X}$ o que significa que a matriz de covariância terá dimensão n^2 enquanto que pela Arquitectura II, como as imagens são dispostas nas colunas, a matriz de covariância terá dimensão m^2 . Durante este trabalho, como se verá no capítulo 5, valores comuns de n e m são: $n=125$ e $m=32 \times 32=1024$. Assim, a dimensão da matriz de covariância será 125^2 na Arquitectura I e 1024^2 na Arquitectura II, o que pode inviabilizar a utilização da Arquitectura II por motivos de escassez de memória, a não ser que se faça uma redução de dimensionalidade prévia à aplicação do ICA. O que pode ser feito por PCA (secção 3.2.4) ou até mesmo reduzindo-se o tamanho as imagens normalizadas.

Quanto ao número de componentes independentes estimadas, ele está majorado pelo número de variáveis aleatórias, ou seja, pela dimensão dos dados apresentados ao bloco ICA. Podem-se determinar as componentes independentes que se pretender mas nunca poderão ser em número superior à quantidade de variáveis aleatórias. Portanto, a escolha da arquitectura irá definir o número máximo de componentes independentes que se podem estimar. Assim, pela Arquitectura I, como as imagens são encaradas como

variáveis aleatórias e os *pixels* observações, só se poderá estimar n componentes, o que no exemplo dado, corresponde a 125. Enquanto que, pela Arquitectura II, como os *pixels* são variáveis aleatórias e as imagens observações, o número máximo de componentes será de m , ou seja, 1024.

Vejamos agora como é aplicado o ICA consoante se esteja em modo de treino ou corrente.

4.3.1 Modo de treino - bloco ICA

Em modo de treino é realizada uma aprendizagem não supervisionada pela técnica ICA a fim de se obter a base de um novo espaço de dados, caracterizado por minimizar a dependência estatística entre as componentes de dados multivariados, que se designará de espaço ICA.

O algoritmo de estimação das componentes independentes utilizado, conforme dito em 3.2.3, foi o FastICA com função de custo não linear dada pela tangente hiperbólica (equação (3.14)). Quanto à forma de determinação das componentes independentes seguiu-se a abordagem paralela uma vez que se pretendia obter todas as componentes e também porque, além de ser mais rápida, há uma menor propagação dos erros numéricos do que se fosse por uma determinação das componentes em série, isto é, uma após a outra.

Seguindo-se a Arquitectura I, a aprendizagem tem a finalidade de determinar a matriz $\mathbf{U}$ após estimação da matriz de separação $\mathbf{W}$, que posteriormente permitirá transformar as imagens do espaço de dados original para o espaço ICA. Por outro lado, na Arquitectura II o objectivo é encontrar a matriz de separação $\mathbf{W}$.

Representando-se por $\mathbf{X}_{\text{treino}}$ a matriz que contém as imagens de treino, a fase de treino ICA consistirá em:

➤ Arquitectura I

1. Dispor todas as imagens de treino nas linhas de $\mathbf{X}_{\text{treino}}$
2. Determinar a matriz de separação, $\mathbf{W}$, pelo FastICA
3. Obter as imagens da base do espaço ICA, $\mathbf{U}$, por $\mathbf{U}=\mathbf{W}\mathbf{X}_{\text{treino}}$

➤ Arquitectura II

1. Dispor todas as imagens de treino nas colunas de $\mathbf{X}_{\text{treino}}$
2. Aplicar o FastICA para obter as imagens da base do espaço ICA, dadas pela matriz de separação, $\mathbf{W}$

4.3.2 Modo de teste ou corrente – bloco Transformação ICA

No sistema em modo corrente, também designado como modo de teste, o objectivo é simplesmente obter a representação ICA das imagens, ou seja, as componentes que definem a imagem no espaço ICA.

Seguindo-se a Arquitectura I, a representação ICA, $\mathbf{b}_{\text{teste}}$, de uma imagem obtém-se por

$$\mathbf{b}_{\text{teste}} = \mathbf{x}_{\text{teste}} \mathbf{U}^T (\mathbf{U} \mathbf{U}^T)^{-1} \quad (4.9)$$

onde $\mathbf{x}_{\text{teste}}$ é o vector linha que representa a imagem no espaço original e $\mathbf{U}^T (\mathbf{U} \mathbf{U}^T)^{-1}$ a forma de Moore-Penrose de obtenção da matriz pseudo-inversa.

De forma semelhante, representando a imagem por $\mathbf{x}_{\text{teste}}$, agora vector coluna, e a sua representação ICA por $\mathbf{b}_{\text{teste}}$, a mudança de representação segundo a Arquitectura II é dada por

$$\mathbf{b}_{\text{teste}} = \mathbf{W} \mathbf{x}_{\text{teste}} \quad (4.10)$$

4.4 Classificador

O último bloco na cadeia de processamento em modo corrente (Figura 4.2) corresponde ao Classificador, o bloco que terá a responsabilidade de tomar a decisão na classificação do gesto numa das classes de gesto existentes ou como gesto desconhecido. Para executar esta tarefa decidiu-se utilizar o classificador do vizinho mais próximo devido à sua baixa complexidade e à boa qualidade dos resultados conseguidos. Assim, escolhe-se um conjunto de imagens, a que chamaremos imagens de referência, às quais se faz todo o processamento necessário (Figura 4.2) até se aplicar a equação (4.9) ou a (4.10) a fim de se obter as representações ICA das imagens de referência, caso se esteja a utilizar, respectivamente, a Arquitectura I ou a Arquitectura II. Por outro lado, cada nova imagem capturada pelo sistema seguirá na cadeia de processamento (Figura 4.2) até que ao bloco Classificador chega a sua representação ICA, $\mathbf{b}_{\text{teste}}$. Por conseguinte, o classificador do vizinho mais próximo, operando no espaço ICA, irá classificar a

imagem gesto como pertencente à classe da imagem de referência da qual mais se aproxima.

A medida de proximidade utilizada neste trabalho foi a medida de semelhança do cosseno dada por (4.11), onde d_{12} representa a semelhança cosseno entre os vectores imagem $\mathbf{b}_1$ e $\mathbf{b}_2$. A medida de semelhança do cosseno representa o cosseno do ângulo entre dois vectores num espaço m dimensional mas também pode ser interpretada como uma distância Euclideana normalizada.

$$d_{12}(\mathbf{b}_1, \mathbf{b}_2) = \frac{\mathbf{b}_1 \cdot \mathbf{b}_2}{\|\mathbf{b}_1\| \|\mathbf{b}_2\|} \quad (4.11)$$

A preferência da escolha da medida de semelhança do cosseno, face à distância Euclideana, deve-se a ter dado resultados significativamente melhores nas experiências realizadas até à presente data [Bartlett, Movellan et al., 2002]. O que pode ter a explicação no facto dos vectores da base ICA não serem mutuamente ortogonais [Draper, Baek et al., 2003].

Para prever os casos de gestos desconhecidos, ou pouco semelhantes, se a medida de semelhança não for superior a um determinado valor então o gesto é classificado como desconhecido. O valor desse limiar deve ser determinado após se analisarem as semelhanças que é comum obter e de acordo com o risco que se pode ter em classificações erradas. Um valor elevado vai garantir poucos erros mas terá o inconveniente de classificar muitos gestos como desconhecidos, ao contrário de um valor baixo que causará poucos desconhecidos mas que aumenta a probabilidade de classificações erradas.

Na Tabela 4.1 mostra-se a semelhança cosseno de um conjunto de imagens de teste às imagens referência em ambas as arquitecturas.

Arquitectura I Arquitectura II						
         		0,868 0,904	0,358 0,678	0,236 0,706	-0,248 0,619	0,163 0,684
		0,881 0,932	0,329 0,599	0,227 0,656	-0,250 0,599	0,156 0,703
		0,418 0,612	0,829 0,801	0,074 0,762	-0,218 0,624	-0,096 0,718
		0,381 0,628	0,884 0,910	0,053 0,789	-0,199 0,708	-0,110 0,722
		0,284 0,541	0,037 0,717	0,983 0,986	0,307 0,669	0,393 0,786
		0,304 0,619	0,005 0,743	0,914 0,943	0,283 0,739	0,506 0,821
		-0,233 0,384	-0,176 0,531	0,271 0,683	0,898 0,889	0,149 0,659
		-0,213 0,269	-0,171 0,433	0,274 0,494	0,850 0,708	0,163 0,495
		0,293 0,755	-0,070 0,667	0,355 0,690	0,154 0,650	0,909 0,925
		0,313 0,787	-0,015 0,661	0,422 0,787	0,222 0,653	0,908 0,879

Tabela 4.1 – Semelhança cosseno entre imagens de gestos

CAPÍTULO 5

Análise de resultados

Uma vez seguida a metodologia apresentada no capítulo 4 executaram-se uma série de testes por forma a analisar a eficácia da abordagem adoptada e a averiguar qual a sua sensibilidade à dimensão das imagens normalizadas, ao número de componentes independentes, à quantidade de imagens de referência, à dimensão do conjunto de treino, ao utilizador e às condições de captura de imagem. É o resultado desses testes que aqui se analisa.

Antes de prosseguir é importante esclarecer quais as denominações que vão ser utilizadas para os diferentes conjuntos de imagens. Definiram-se então 3 grupos principais de imagens: conjunto de treino, conjunto de referência e conjunto de teste.

Por conjunto de treino entende-se as imagens utilizadas para treino do modelo ICA, ou seja, as imagens utilizadas no modo de treino a fim de se obter a matriz que transforma os dados (imagens) do espaço original para o espaço ICA. Atendendo a que o ICA é uma técnica de aprendizagem não supervisionada, houve o cuidado de construir cada um dos conjuntos de treino com um mesmo número de imagens de cada classe de gesto. As imagens que constituem o conjunto de referência são as que representam as diferentes classes e que servirão de termo de comparação no classificador do vizinho mais próximo a fim de determinar a que classe é que a imagem de teste mais se assemelha. De referir que estas imagens não resultam de qualquer combinação de

imagens normalizadas, trata-se de um grupo de imagens normalizadas escolhidas como representativas de uma classe. Regra geral, no conjunto de referência tem-se igual número de imagens por classe de gesto, mas isso não é obrigatório. Em casos em que imagens pertencentes a uma mesma classe possam ter representações bem distintas (tal como foi apontado na secção 4.2) poder-se-á justificar ter diferente número de imagens por classe de gesto.

Por conjunto de teste entende-se as imagens utilizadas para, tal como o nome diz, testar o sistema reconhecedor de gestos a fim de obter medidas de desempenho. Trata-se das imagens utilizadas com o sistema em modo corrente/teste.

Apesar de se poder utilizar imagens do conjunto de referência que também pertencem ao conjunto de treino, no decorrer deste trabalho optou-se preferencialmente por não o fazer. Por isso, as imagens do conjunto de treino serão distintas das imagens de referência salvo indicação em contrário. Relativamente ao conjunto de teste, houve sempre o cuidado de escolher imagens de teste que não pertencessem nem ao conjunto de referência nem ao conjunto de treino.

A forma mais objectiva de avaliar a qualidade do reconhecedor de gestos é através da comparação da taxa de acerto dada pela equação (4.12), onde $N_{\text{correctos}}$ representa o número de imagens correctamente classificadas e N_{total} o número total de imagens submetidas a teste.

$$T_{\text{acerto}} = \frac{N_{\text{correctos}}}{N_{\text{total}}} 100\% \quad (4.12)$$

Todavia, a mera comparação da taxa de acerto entre testes executados com premissas diferentes pode não ser suficiente, principalmente nos casos de diferentes testes originarem taxas de acerto semelhantes ou até mesmo iguais. Como comparar o desempenho do reconhecedor nestes casos? É possível dizer que um é melhor que o outro? Vejamos os dois casos presentes na Figura 5.1 que ilustram a localização espacial de variáveis aleatórias num espaço de 2 dimensões.

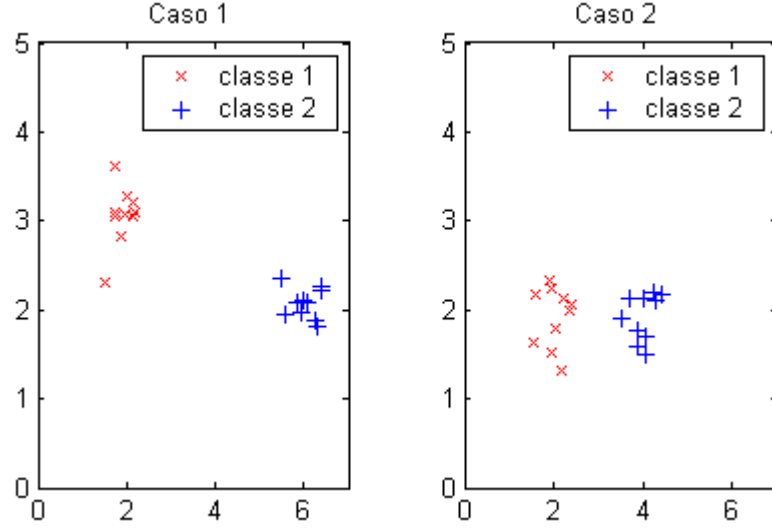


Figura 5.1- Diferentes capacidades de discriminação

Com uma escolha adequada da referência de cada uma das classes é natural esperar ter-se $T_{\text{acerto}}=100\%$, no entanto consegue-se verificar visualmente que como as nuvens de pontos das duas classes estão mais distantes no “Caso 1”, o classificador inerente a este caso será menos sensível a factores externos do que o do “Caso 2”.

Por forma a poder comparar o poder de discriminação das diferentes configurações do reconhecedor, ao longo deste trabalho analisou-se a diferença entre as médias das medidas de semelhança às imagens de referência da classe mais próxima e as da segunda classe mais próxima como factor indicativo da capacidade de discriminação. A notação escolhida para representar estas duas médias das medidas de semelhança foi:

S_{1^a} média do valor da semelhança à classe mais próxima

S_{2^a} média do valor da semelhança à segunda classe mais próxima

Além destes dois factores de desempenho, por vezes também se irá utilizar S_r , que designaremos de factor de confiança, que conjuga S_{1^a} e S_{2^a} , de acordo com (4.13), dando uma grandeza que quanto maior for, mais discriminante será o classificador associado.

$$S_r = \frac{S_{1^a} - S_{2^a}}{S_{1^a}} \quad (4.13)$$

Portanto, um classificador com boa capacidade de discriminação deverá ter S_{1^a} elevada e o mais distante possível de S_{2^a} , o que significará S_r elevado.

Apresentadas as medidas de desempenho que se vai usar passemos então aos testes.

Apesar de já haver indícios do melhor comportamento da Arquitectura I face à Arquitectura II no reconhecimento de gestos (3.2.5.2) decidiu-se testar também a Arquitectura II de forma a poder confirmar este facto.

Todavia é importante fazer uma salvaguarda relativamente aos resultados apresentados para a Arquitectura II visto ter surgido uma adversidade na implementação dos *scripts* Matlab de obtenção de resultados. A certa altura da execução dos testes verificou-se que a transformação matriz imagem em vector imagem estava a ser executada de forma diferente consoante se estava em treino ou teste. Numa das situações estava-se a fazer concatenação de colunas e na outra concatenação de linhas. Após se ter corrigido isto, a Arquitectura II passou a dar taxas de acerto bastante fracas e os resultados obtidos nos diferentes testes não faziam sentido. Por isso, deverá haver uma gralha no código fonte que inicialmente compensaria a concatenação errada e que actualmente impede a obtenção dos resultados devidos para essa arquitectura. Por esta razão, visto não ter sido possível detectar essa gralha em tempo útil, os resultados aqui apresentados para a Arquitectura II são os que resultaram dos testes iniciais pois é nossa convicção serem os correctos, já que fazem sentido.

O primeiro teste a fazer consiste em verificar qual a taxa de acerto esperada e qual a sua sensibilidade ao conjunto de teste. Assim, seleccionaram-se 200 imagens do mesmo gesticulador, 40 de cada gesto, de entre as quais foi retirada uma imagem para teste e as restantes usadas para treino, repetindo-se o teste 200 vezes por forma a testar todas as imagens. Quanto às imagens de referência apenas se utilizou uma por cada classe de gesto, escolhida ao acaso mas assegurando que o gesto que representa se encontra bem definido.

Resumindo, cada repetição deste teste é executada nas seguintes condições:

- | | |
|---------------------------------------|-------|
| ❑ Imagens de treino (por gesto) | 39 |
| ❑ Imagens de referência (por gesto) | 1 |
| ❑ Imagens de teste (por gesto) | 1 |
| ❑ Dimensões das imagens normalizadas | 32x32 |
| ❑ Número de componentes independentes | 60 |

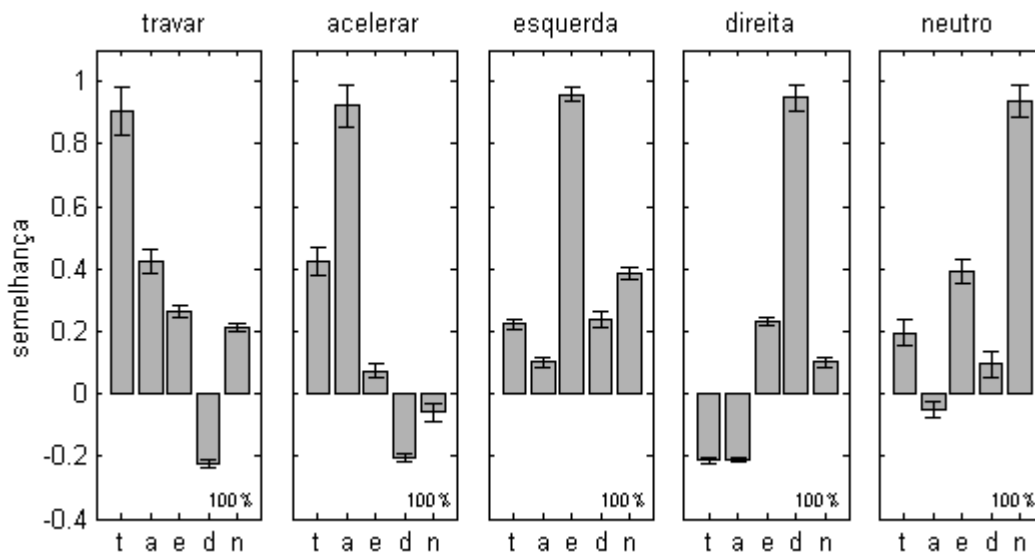
A média dos resultados das 200 simulações é apresentada na Tabela 5.1 onde juntamente com as médias da semelhança à classe mais próxima e à 2ª mais próxima está indicado o respectivo desvio padrão.

	T_{acerto}	S_{1^a}	S_{2^a}	S_r
Arquitectura I	100%	$0,932 \pm 0,053$	$0,372 \pm 0,031$	0,601
Arquitectura II	100%	$0,954 \pm 0,034$	$0,669 \pm 0,021$	0,299

Tabela 5.1 – Desempenho do reconhecedor

Em ambas as arquitecturas a taxa de acerto é muito boa chegando mesmo aos 100%, porém a Arquitectura I é mais discriminante do que a Arquitectura II já que apresenta maior diferença entre S_{1^a} e S_{2^a} , o que se manifesta num factor de confiança praticamente duplo. O que pode indiciar que a Arquitectura I seja mais vocacionada para reconhecimento de gestos do que a Arquitectura II.

Nas Figura 5.2 e Figura 5.3 apresentam-se gráficos de barras das médias e desvio padrão (representado pelos bigodes) da medida de semelhança de cada um dos gestos de teste às diferentes classes existentes. No eixo das abcissas listam-se as diferentes classes de gesto de teste onde “t”, “a”, “e”, “d” e “n” significam, respectivamente, ‘travar’, ‘acelerar’, ‘esquerda’, ‘direita’ e ‘neutro’. A taxa de acerto para cada uma das 5 classes de gesto testadas encontra-se apresentada no canto inferior direito do respectivo gráfico.

**Figura 5.2 – Semelhança dos gestos de teste a cada classe - Arquitectura I**

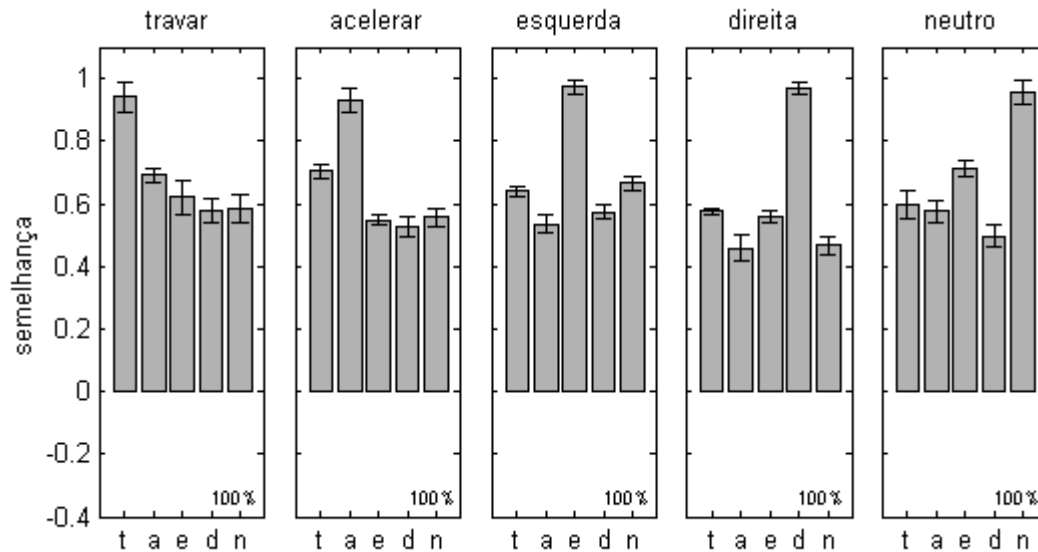


Figura 5.3 – Semelhança dos gestos de teste a cada classe - Arquitectura II

Da análise destas duas figuras confirma-se que a Arquitectura I é mais discriminante do que a Arquitectura II porque, apesar da diferença de semelhanças entre as duas classes mais próximas ser parecida, a diferença à semelhança das outras classes é maior na Arquitectura I, o que também pode ser concluído pelo factor de confiança, S_r , presente na Tabela 5.1. Outro facto que se pode constatar é que a barra correspondente à classe à qual o gesto de teste pertence, é a mais elevada em todos os casos deixando mesmo uma distância considerável à segunda barra mais elevada, logo todos os gestos são bem discriminados.

5.1 Sensibilidade à dimensão das imagens normalizadas

De forma a averiguar qual a sensibilidade do reconhecedor à dimensão das imagens normalizadas testou-se o reconhecedor para imagens normalizadas quadradas com 8, 16, 32 e 64 *pixels* de lado, nas seguintes condições:

- ❑ Imagens de treino (por gesto) 25
- ❑ Imagens de referência (por gesto) 1
- ❑ Imagens de teste (por gesto) 10
- ❑ Número de componentes independentes 60

A escolha de apenas uma imagem de referência por gesto foi feita com o objectivo de se estar na pior situação e dessa forma aumentar a sensibilidade à variação de outros parâmetros (facto que será demonstrado na secção 5.4). Com o objectivo de obter

resultados independentes das imagens de referência escolhidas, repetiu-se este teste 8 vezes com diferentes conjuntos de referência, por isso, os valores apresentados são a média dessas repetições.

Pela Architectura I não foi possível ir até 128 *pixels* de lado por motivos de insuficiência de memória no cálculo da matriz pseudo-inversa de U . No caso da Architectura II teve-se de ficar por 32x32, também por problemas de memória mas, neste caso, para o cálculo da matriz de covariância.

Como na Architectura II o número de componentes independentes está majorado pela dimensão das observações, isto é, a quantidade de *pixels*, foi necessário fazer uma redução de dimensionalidade, por PCA, por forma a assegurar que nos diferentes casos de teste o número de componentes independentes estimadas fosse o mesmo. No caso mais desfavorável, imagens 8x8, só se conseguem estimar no máximo 64 componentes independentes, por isso, escolheu-se usar 60 componentes por forma a salvaguardar também possíveis casos de existência de singularidades na matriz de covariância que causam uma diminuição do número de componentes independentes estimadas.

Analisando a Tabela 5.2 não se consegue tirar conclusões da influência das dimensões das imagens normalizadas porque a taxa de acerto mantém-se inalterável nos 100% de sucessos.

	8x8	16x16	32x32	64x64
Arquitectura I	100%	100%	100%	100%
Arquitectura II	100%	100%	100%	

Tabela 5.2 – Taxa de acerto em função das dimensões das imagens

Com o aumento da dimensão das imagens aumenta a informação relativa a cada amostra (*i.e.* imagem capturada), logo é de esperar que o reconhecedor discrimine melhor as diferentes classes de gesto. Se na Tabela 5.2 isso não era possível concluir, na Figura 5.4 já se pode ver que à medida que aumenta a dimensão das imagens, a semelhança da imagem de teste à imagem de referência da classe a que pertence vai aumentando. Por outro lado, a semelhança à segunda classe mais próxima também aumenta mas como é em menor valor absoluto, a capacidade de discriminação aumenta.

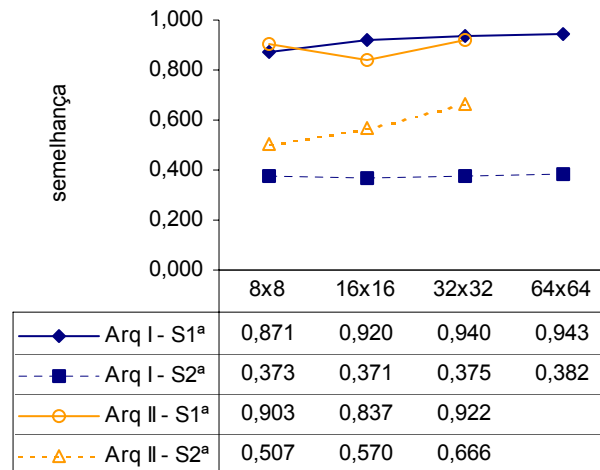


Figura 5.4 – Semelhança às duas classes mais próximas em função das dimensões das imagens

Analisando os valores da semelhança à classe mais próxima para o caso da Arquitectura II verifica-se que em imagens 8x8 a semelhança à classe mais próxima é superior à do caso de imagens 32x32, o que pode ter explicação na organização dos dados da Arquitectura II (secção 3.2.5.2) já que origina que, em imagens 32x32, se tenha mais variáveis aleatórias do que observações e que, por isso, a estimação do modelo ICA seja realizada com pouca informação. Como na Arquitectura II as imagens são tratadas como observações e os *pixels* como variáveis aleatórias então tem-se 64, 256 e 1024 variáveis aleatórias para, respectivamente, 8x8, 16x16 e 32x32, enquanto que observações tem-se sempre 125 (5 gestos com 25 imagens). Logo apenas no caso 8x8 é que existem mais observações que variáveis aleatórias o que permite um treino mais rigoroso e por isso melhores resultados. No entanto é importante voltar a referir que esta análise tem de ser vista com alguma prudência já que não se tem a certeza absoluta que os resultados obtidos são os correctos.

Para as classes de gesto utilizadas neste trabalho talvez fosse suficiente utilizar imagens de 16 *pixels* de lado, ou até mesmo de 8, no entanto decidiu-se utilizar imagens de 32 *pixels* de lado para ter alguma folga e também porque se entende que esta é uma dimensão razoável.

5.2 Sensibilidade ao número de componentes independentes

No teste anterior foram utilizadas 60 componentes independentes mas a pergunta que fica é: qual a influência do número de componentes independentes na qualidade do

reconhecedor? Para responder a esta pergunta foi testado o reconhecedor com diferente número de componentes independentes. Como não há forma de ordenar as componentes independentes por ordem de importância, ou mais correctamente, as ordenações existentes são essencialmente heurísticas (secção 3.2.4), o número de componentes é definido por uma prévia redução de dimensionalidade efectuada por PCA.

As condições em que foi realizado este teste são as seguintes:

- ❑ Imagens de treino (por gesto) 25
- ❑ Imagens de referência (por gesto) 1
- ❑ Imagens de teste (por gesto) 10
- ❑ Dimensão das imagens normalizadas 32x32
- ❑ Variação do número de componentes independentes desde 3 a 125

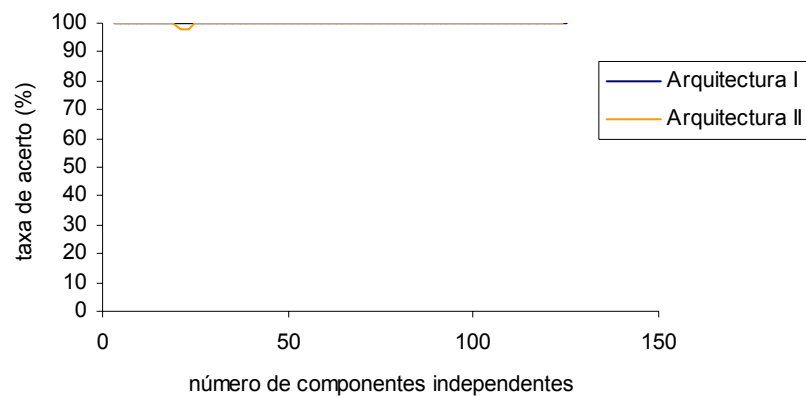


Figura 5.5 – Taxa de acerto em função do número de componentes independentes

Mais uma vez a taxa de acerto é de 100% (ver Figura 5.5), excepção feita a dois casos referentes à Arquitetura II em que existe 1 insucesso quando o número de componentes independentes é de 21 e 23. Na Tabela 5.3 pode-se ver os dois casos de insucesso em que um gesto ‘neutro’ se assemelhava mais ao gesto de referência ‘esquerda’.

semelhanças às imagens de referência					
21 componentes independentes	0,749	0,656	0,934	0,767	0,930
23 componentes independentes	0,749	0,659	0,945	0,721	0,943

Tabela 5.3 – Classificações erradas na Arquitetura II

Um facto curioso deste teste é que a semelhança às duas classes mais próximas baixa ligeiramente à medida que se aumenta o número de componentes independentes (Figura 5.6), o que pode ter a explicação no facto de existir uma remoção, não premeditada, de ruído quando se usam poucas componentes independentes. Embora seja uma redução quase desprezável, está de acordo com o que se referiu na secção 3.2.4 quanto à melhoria do desempenho do ICA quando há uma redução de dimensionalidade por PCA.

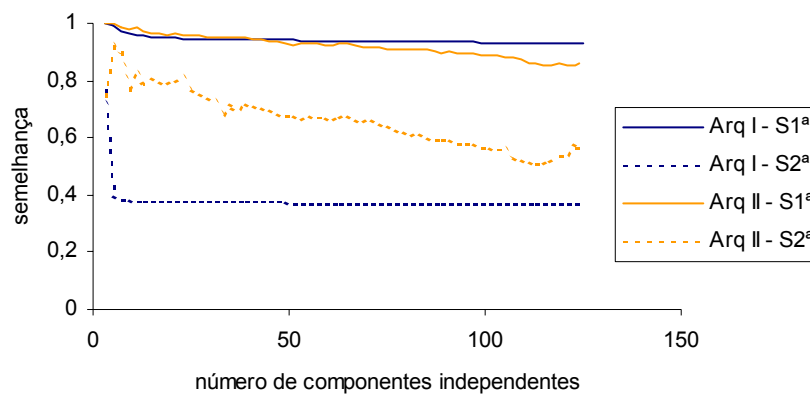


Figura 5.6 – Semelhança às duas classes mais próximas em função do número de componentes independentes

5.3 Sensibilidade às imagens de treino

De forma a averiguar qual a necessidade de treino que o sistema reconhecedor precisa fez-se variar o número de imagens de treino por gesto desde 1 a 25, fixando as imagens de referência, nas seguintes condições:

- | | |
|--|-------|
| ❑ Imagens de referência (por gesto) | 1 |
| ❑ Imagens de teste (por gesto) | 10 |
| ❑ Dimensão das imagens normalizadas | 32x32 |
| ❑ Componentes independentes utilizadas | todas |

A taxa de acerto foi de 100% para todos os casos de ambas as arquiteturas e apenas é possível ver variações nas semelhanças às duas classes mais próximas.

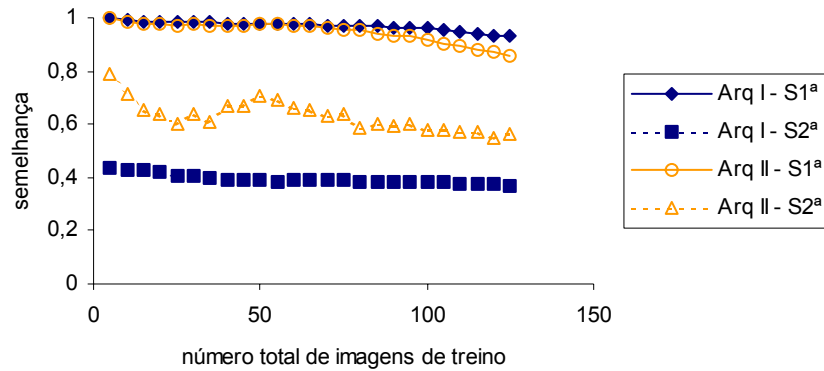


Figura 5.7 – Semelhança às duas classes mais próximas em função do número de imagens de treino

Contudo é importante notar que o número de componentes independentes utilizadas na Arquitectura I, como está limitado pelo número de imagens de treino, não foi constante ao longo de todas as simulações. O que levou a que este teste fosse um pouco inconclusivo porque nas simulações com pequena quantidade de imagens de treino as componentes independentes também escasseavam tendo por isso um impacto significativo no resultado final.

Para se poder tirar conclusões mais fundamentadas seria necessário ter bastante mais imagens de treino do que as 125 que se usaram, o que não foi possível fazer por não se dispor dessa quantidade de imagens de um só utilizador e por não haver tempo suficiente para as adquirir. Além de permitir chegar a melhores conclusões, a existência de mais imagens de treino permitiria também averiguar a ocorrência de *overlearning* do ICA, problema algumas vezes reportado na bibliografia.

5.4 Sensibilidade às imagens de referência

O teste que se seguiu tinha o objectivo de averiguar qual a influência das imagens de referência no desempenho do reconhecedor. Fez-se então variar o número de imagens de referência por gesto desde 1 a 25 escolhendo-as de uma forma aleatória de entre o conjunto de treino.

Nas Figura 5.8 e Figura 5.9 apresenta-se o resultado deste teste que foi executado nas seguintes condições:

- ❑ Imagens de treino (por gesto) 25
- ❑ Imagens de teste (por gesto) 10
- ❑ Dimensão das imagens normalizadas 32x32
- ❑ Número de componentes independentes 125

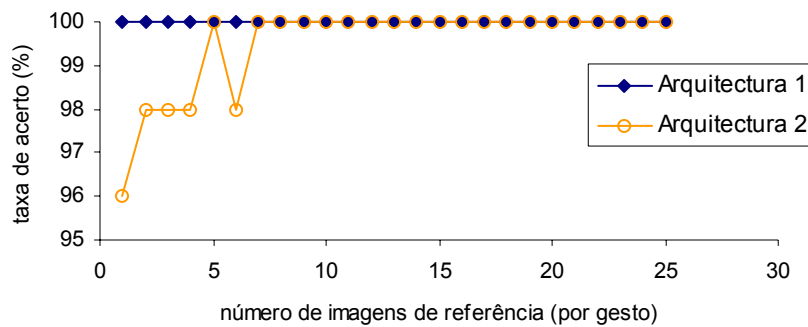


Figura 5.8 – Taxa de acerto em função do número de imagens de referência

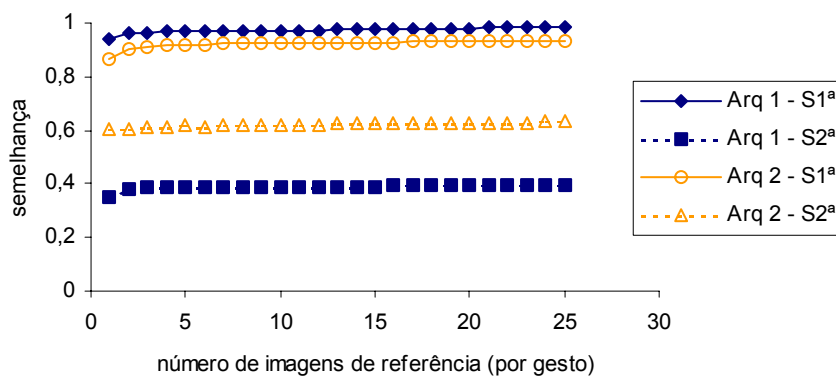


Figura 5.9 – Semelhança em função do número de imagens de referência

Enquanto que a taxa de acerto foi sempre de 100% para a Arquitetura I, na Arquitetura II houve alguma variação entre 96% e 100% até se ter 7 imagens de referência por gesto, o que se deveu a casos particulares de imagens que não foram devidamente classificadas. As imagens em questão são as duas que se encontram na coluna da esquerda da Tabela 5.4. Nesta tabela pode-se ver as semelhanças no caso em que se tinha apenas 1 imagem de referência por gesto e em que os gestos ‘travar’ foram erradamente classificados como ‘esquerda’. É de referir que a imagem de referência do gesto ‘travar’ (1ª linha da tabela) não está totalmente de acordo com a especificação dos gestos (secção 1.3) porque os dedos não estão bem abertos, porém optou-se por aceitar

que esta também fosse uma forma válida de executar o gesto ‘travar’ para não tornar o sistema demasiado inflexível.

semelhanças às imagens de referência						
		0,620	0,539	0,624	0,229	0,525
		0,602	0,463	0,612	0,262	0,502

Tabela 5.4 – Classificações erradas na Arquitectura II no caso de 1 imagem de referência por gesto

Duas possíveis soluções capazes de corrigir as classificações erradas que ocorreram são: reduzir a dimensionalidade dos dados para gerar apenas 60 componentes independentes ou adicionar uma das imagens mal classificadas como imagem de referência. Ambas as alternativas foram testadas e tiveram o efeito desejado.

5.5 Sensibilidade às condições de captura

A sensibilidade do reconhecedor a cenários de captura com iluminações diferentes ou câmaras diferentes também foi testada a fim de aferir qual a robustez deste reconhecedor relativamente a estas variáveis.

Assim treinou-se o sistema com 25 imagens de cada gesto, executadas por um mesmo utilizador, e estimou-se a matriz de transformação do espaço original para o espaço ICA com 125 componentes independentes. De seguida foram seleccionadas 13 imagens que apresentassem as maiores diferenças possíveis, nos níveis de cinzento, de entre as existentes do utilizador com que se treinou o sistema. Por fim, obteve-se a semelhança entre todas de forma a verificar qual a influência da luminosidade. Os resultados estão expressos na Tabela 5.5 e uma análise cuidada verifica que, ignorando a comparação com a própria imagem, pela Arquitectura I tem-se 100% de sucessos enquanto que pela Arquitectura II há 1 caso de insucesso (a 5ª imagem da vertical, gesto ‘esquerda’, é classificada como gesto ‘neutro’).

Arq I Arq II														
	1,000	0,996	0,706	0,701	0,919	0,942	0,939	0,938	0,890	0,887	0,882	0,879	0,553	
	1,000	0,865	0,527	0,520	0,541	0,462	0,660	0,626	0,551	0,614	0,635	0,553	0,500	
	0,996	1,000	0,708	0,703	0,936	0,956	0,954	0,954	0,916	0,911	0,907	0,905	0,571	
	0,865	1,000	0,429	0,407	0,422	0,378	0,495	0,422	0,533	0,458	0,493	0,451	0,364	
	0,706	0,708	1,000	1,000	0,795	0,787	0,781	0,774	0,756	0,796	0,797	0,781	0,672	
	0,527	0,429	1,000	0,969	0,774	0,588	0,618	0,515	0,747	0,750	0,734	0,776	0,552	
	0,701	0,703	1,000	1,000	0,790	0,782	0,775	0,769	0,751	0,792	0,793	0,777	0,670	
	0,520	0,407	0,969	1,000	0,757	0,612	0,652	0,560	0,713	0,767	0,741	0,762	0,589	
	0,919	0,936	0,795	0,790	1,000	0,987	0,991	0,991	0,961	0,964	0,963	0,959	0,692	
	0,541	0,422	0,774	0,757	1,000	0,675	0,735	0,716	0,830	0,838	0,851	0,846	0,678	
	0,942	0,956	0,787	0,782	0,987	1,000	0,999	0,997	0,950	0,952	0,950	0,946	0,646	
	0,462	0,378	0,588	0,612	0,675	1,000	0,900	0,813	0,644	0,638	0,621	0,659	0,419	
	0,939	0,954	0,781	0,775	0,991	0,999	1,000	0,999	0,958	0,959	0,958	0,954	0,663	
	0,660	0,495	0,618	0,652	0,735	0,900	1,000	0,927	0,675	0,750	0,760	0,705	0,580	
	0,938	0,954	0,774	0,769	0,991	0,997	0,999	1,000	0,964	0,964	0,963	0,960	0,668	
	0,626	0,422	0,515	0,560	0,716	0,813	0,927	1,000	0,641	0,761	0,773	0,645	0,584	
	0,890	0,916	0,756	0,751	0,961	0,950	0,958	0,964	1,000	0,997	0,996	0,998	0,715	
	0,551	0,533	0,747	0,713	0,830	0,644	0,675	0,641	1,000	0,842	0,851	0,908	0,485	
	0,887	0,911	0,796	0,792	0,964	0,952	0,959	0,964	0,997	1,000	1,000	0,999	0,747	
	0,614	0,458	0,750	0,767	0,838	0,638	0,750	0,761	0,842	1,000	0,968	0,921	0,629	
	0,882	0,907	0,797	0,793	0,963	0,950	0,958	0,963	0,996	1,000	1,000	0,999	0,754	
	0,635	0,493	0,734	0,741	0,851	0,621	0,760	0,773	0,851	0,968	1,000	0,912	0,676	
	0,879	0,905	0,781	0,777	0,959	0,946	0,954	0,960	0,998	0,999	0,999	1,000	0,737	
	0,553	0,451	0,776	0,762	0,846	0,659	0,705	0,645	0,908	0,921	0,912	1,000	0,498	
	0,553	0,571	0,672	0,670	0,692	0,646	0,663	0,668	0,715	0,747	0,754	0,737	1,000	
	0,500	0,364	0,552	0,589	0,678	0,419	0,580	0,584	0,485	0,629	0,676	0,498	1,000	

Tabela 5.5 – Semelhança entre imagens

Um facto interessante é que, nos casos em que se tem mais do que uma imagem da mesma classe, pela Arquitectura I todas as imagens dessa mesma classe são mais semelhantes à imagem em causa do que as das outras classes, o que não acontece na Arquitectura II.

Assim, conclui-se que o ICA apresenta uma boa robustez à luminosidade o que era de certa forma esperado porque o modelo ICA é desenvolvido de forma a maximizar a transmissão de informação na presença de ruído e portanto, deverá ter alguma robustez a variações como é o caso da luminosidade, que pode ser considerada uma forma de ruído [Bartlett, Movellan et al., 2002].

Porém, em casos em que existam mais classes, ou classes menos discriminadas, do que as deste trabalho, é possível que a robustez à luminosidade não seja suficiente para manter a taxa de acerto e, nesse caso, deve-se então aplicar um pré-processamento a fim

de normalizar o nível médio de cinzento nas imagens, o que pode ser conseguido, por exemplo, por equalização de histograma.

De referir que estas imagens de diferentes níveis de luminosidade pertencem a uma mesma captura em que houve um problema na câmara de filmar que originou que o controle automático de brilho estivesse constantemente a variar.

Uma outra forma de avaliar a sensibilidade às condições de captura é testar o sistema com imagens adquiridas de forma diferente. Assim capturaram-se imagens de um utilizador em dois cenários distintos:

- Cenário 1 câmara de vídeo de boa qualidade com boas condições de iluminação
- Cenário 2 câmara de vídeo de baixo custo e iluminação não cuidada

O treino do sistema foi realizado com imagens provenientes do Cenário 1 e o teste foi executado nas seguintes condições:

- ❑ Imagens de treino (por gesto) 10 do Cenário 1
- ❑ Imagens de referência (por gesto) 1
- ❑ Imagens de teste (por gesto) 10 do Cenário 2
- ❑ Dimensão das imagens normalizadas 32x32
- ❑ Número de componentes independentes 100

Quanto à proveniência das imagens de referência existem duas possibilidades ou vêm do Cenário 1 ou do Cenário 2 e, por isso, testou-se o sistema em ambas as alternativas. Os resultados obtidos apresentam-se na Tabela 5.6 e na Tabela 5.7 em que a primeira coluna das tabelas indica a origem das imagens do conjunto de referência, se vieram do Cenário 1 ou do Cenário 2.

conjunto de referência	conjunto de teste	T _{acerto}	S _{1a}	S _{2a}
Cenário 1	Cenário 2	82,0%	0,833 ±0,026	0,394 ±0,030
Cenário 2	Cenário 2	97,4%	0,938 ±0,017	0,513 ±0,019

Tabela 5.6 – Desempenho para cenários de captura diferentes - Arquitectura I

conjunto de referência	conjunto de teste	T _{acerto}	S _{1a}	S _{2a}
Cenário 1	Cenário 2	80%	0,766 ±0,036	0,540 ±0,033
Cenário 2	Cenário 2	100%	0,876 ±0,038	0,526 ±0,027

Tabela 5.7 – Desempenho para cenários de captura diferentes - Arquitectura II

Pela análise das taxas de acerto e das semelhanças às duas classes mais próximas verifica-se que, se o conjunto de referência for constituído por imagens adquiridas nas mesmas condições que as imagens de teste, então os resultados serão melhores.

De assinalar que treinando o sistema com imagens do Cenário 2 e testando com imagens do Cenário 1 os resultados seriam análogos.

Na Tabela 5.8 e Tabela 5.9 apresentam-se os resultados de mais um teste de sensibilidade do sistema às condições de captura. A particularidade deste teste é que as imagens de treino provêm de uma captura que se fez em que a mão não estava equipada com a luva, gerando assim diferenças significativas do conjunto de treino que utilizou imagens da mão com luva.

conjunto de referência	conjunto de teste	T_{acerto}	S_{1^a}	S_{2^a}
com luva	sem luva	100%	$0,929 \pm 0,014$	$0,571 \pm 0,025$
sem luva	sem luva	100%	$0,966 \pm 0,015$	$0,574 \pm 0,018$

Tabela 5.8 – Desempenho para teste sem luva – Arquitectura I

conjunto de referência	conjunto de teste	T_{acerto}	S_{1^a}	S_{2^a}
com luva	sem luva	68%	$0,842 \pm 0,019$	$0,744 \pm 0,020$
sem luva	sem luva	96%	$0,934 \pm 0,024$	$0,776 \pm 0,017$

Tabela 5.9 – Desempenho para teste sem luva – Arquitectura II

Como se conclui dos resultados apresentados, o ICA tolera bem pequenas diferenças no cenário de captura.

5.6 Sensibilidade ao utilizador

Depois de realizados todos os testes anteriores ainda há uma dúvida que persiste. Como reage o sistema quando lhe é apresentado um utilizador diferente do de treino? Para responder a esta questão testou-se o sistema com um conjunto de treino formado por imagens de um utilizador (A) e o conjunto de teste oriundo de outro utilizador (B), nas seguintes condições:

- ❑ Imagens de treino (por gesto) 25 do utilizador A
- ❑ Imagens de referência (por gesto) 1
- ❑ Imagens de teste (por gesto) 10 do utilizador B
- ❑ Dimensão das imagens normalizadas 32x32
- ❑ Número de componentes independentes 125

Não se utilizou mais do que 1 imagem de referência por gesto para não simplificar o trabalho do classificador e desta forma ser mais fácil existirem diferenças relativamente ao caso de se treinar e testar o sistema com um único utilizador.

Os resultados obtidos estão apresentados na Tabela 5.10 e na Tabela 5.11 com as duas variantes possíveis: imagens de referência do utilizador A ou do utilizador B. De referir que estes valores resultam da média de 8 simulações com conjuntos de referência distintos por forma a apresentar resultados que não fossem totalmente dependentes das imagens de referência escolhidas, independentemente de virem do utilizador A ou B.

conjunto de referência	conjunto de teste	T_{acerto}	S_{1a}	S_{2a}
utilizador A	utilizador B	89,3%	$0,877 \pm 0,020$	$0,631 \pm 0,030$
utilizador B	utilizador B	91,3%	$0,921 \pm 0,017$	$0,634 \pm 0,027$

Tabela 5.10 – Desempenho com utilizadores diferentes - Arquitectura I

conjunto de referência	conjunto de teste	T_{acerto}	S_{1a}	S_{2a}
utilizador A	utilizador B	77,3%	$0,828 \pm 0,025$	$0,719 \pm 0,025$
utilizador B	utilizador B	88,7%	$0,891 \pm 0,026$	$0,739 \pm 0,022$

Tabela 5.11 – Desempenho com utilizadores diferentes - Arquitectura II

A conclusão que se tira é que o desempenho do sistema reconhecedor depende do utilizador com que foi treinado visto que houve uma redução da ordem dos 9% na taxa de acerto. Ambas as arquitecturas sofreram uma degradação semelhante na taxa de acerto mas a Arquitectura I continua a dar resultados um pouco melhores que a Arquitectura II.

Um facto a assinalar, aliás esperado, é a melhoria conseguida quando se utilizam imagens de referência do mesmo utilizador que as do conjunto de teste. Logo, num caso real, se for possível, valerá a pena fornecer imagens de referência do utilizador que vai usar o sistema. O ideal seria mesmo treinar o sistema para o novo utilizador, no entanto essa tarefa é bastante trabalhosa e pode ser mesmo impossível. O fornecimento de várias imagens de referência por gesto será mais viável e permitirá já alguma melhoria no desempenho do reconhecedor.

Visto que os resultados obtidos foram pouco satisfatórios utilizando apenas 1 imagem de referência por gesto resolveu-se executar o mesmo teste, mas com 6 imagens de referência por cada gesto. Como se pode ver na Tabela 5.12 e na Tabela 5.13 a

utilização de mais do que uma imagem de referência teve um impacto positivo nos resultados.

conjunto de referência	conjunto de teste	T_{acerto}	S_{1a}	S_{2a}
utilizador A	utilizador B	96%	$0,916 \pm 0,014$	$0,701 \pm 0,019$
utilizador B	utilizador B	100%	$0,975 \pm 0,007$	$0,734 \pm 0,020$

Tabela 5.12 – Desempenho com utilizadores diferentes - Arquitectura I

conjunto de referência	conjunto de teste	T_{acerto}	S_{1a}	S_{2a}
utilizador A	utilizador B	82%	$0,859 \pm 0,031$	$0,757 \pm 0,029$
utilizador B	utilizador B	98%	$0,934 \pm 0,023$	$0,775 \pm 0,030$

Tabela 5.13 – Desempenho com utilizadores diferentes - Arquitectura II

Um facto a assinalar é que mais uma vez a Arquitectura I demonstrou melhor desempenho do que a Arquitectura II. O que vem confirmar a afirmação de [Draper, Baek et al., 2003] de que a Arquitectura I é mais vocacionada para reconhecimento de objectos, uma vez que o reconhecimento de gestos pode ser considerado um caso de reconhecimento de objectos por causa das diferentes classes de gesto terem configurações bem distintas.

5.7 Comparação com um classificador *ad hoc*

A utilização de ICA numa situação como o caso em estudo neste trabalho poderá não ser a mais indicada porque existem poucas classes de gestos que são suficientemente distintos uns dos outros, por isso, um classificador menos complexo poderia permitir realizar o reconhecedor de gestos também com sucesso.

Na Tabela 5.14 mostra-se um gesto de cada uma das classes de gestos juntamente com 4 características derivadas do rectângulo envolvente alinhado: altura, largura, alongamento vertical (quociente da altura pela largura) e ocupação espacial (quociente entre a área da mão e a área do rectângulo envolvente alinhado).

	altura	largura	$\frac{\text{altura}}{\text{largura}}$	ocupação espacial
	89	69	1,290	0,446
	92	19	4,842	0,624
	50	70	0,714	0,423
	39	89	0,438	0,450
	47	54	0,870	0,497

Tabela 5.14 – Características de diferentes gestos

Analisando as características dos gestos verifica-se que o gesto ‘acelerar’ pode ser facilmente detectado por ter uma altura bastante superior à largura do seu rectângulo envolvente alinhado. Por outro lado o gesto ‘direita’ é significativamente mais largo que alto. De entre os 3 gestos que restam, o de ‘travar’ é o único que tem a altura superior à largura. Como o gesto ‘neutro’ é executado com a mão fechada então deverá ter uma ocupação espacial superior à do gesto ‘esquerda’. E assim, com base nos princípios descritos, construiu-se um classificador *ad hoc* cujo funcionamento se encontra detalhado no fluxograma da Figura 5.10, tendo os limiares envolvidos sido determinados empiricamente.

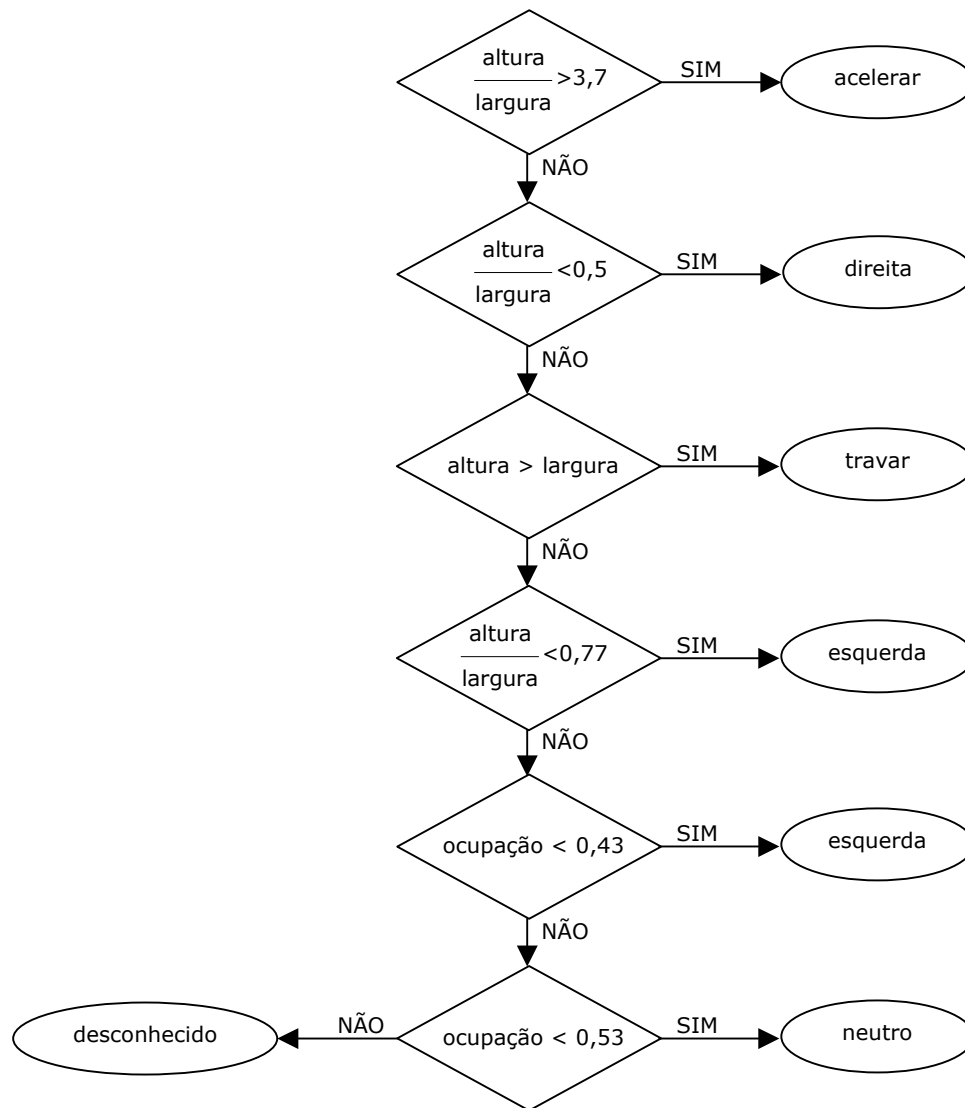


Figura 5.10 – Fluxograma do classificador *ad hoc*

De seguida escolheram-se aleatoriamente 50 imagens (10 de cada gesto), da mesma captura de onde provieram os dados da Tabela 5.14, e testou-se este classificador. O resultado foi uma taxa de acerto de 100%, o que vem demonstrar que, para o caso especial tratado neste trabalho de apenas 5 gestos bem distintos uns dos outros, uma abordagem mais simples que o ICA poderia ser suficiente. Porém, o classificador *ad hoc* aqui apresentado não seria uma escolha adequada porque é muito sensível à forma como se gesticula. Com pequenas variações na execução dos gestos, como por exemplo realizar o gesto 'neutro' com a mão não tão fechada como a presente na Tabela 5.14, ou então com a existência de sombras significativas entre os dedos da mão causadas por uma iluminação deficiente, facilmente se tem erros de classificação. Além disso, o

aumento do número de classes de gestos causa de imediato erros no classificador *ad hoc* porque este foi construído segundo uns princípios muito simplistas e os limiares de decisão foram determinados por tentativa e erro até se conseguir obter uma boa taxa de acerto.

5.8 Demonstrador de comando de movimento

Um dos objectivos deste trabalho passava por construir um demonstrador que permitisse exemplificar o controlo de movimento por análise de gestos simbólicos, baseado na Análise de Componentes Independentes (ICA). Apesar de neste trabalho não haver o compromisso do demonstrador funcionar em tempo-real houve um esforço nesse sentido, visto que num cenário real isso é uma obrigatoriedade para que a pessoa que comanda tenha consciência imediata do resultado das suas ordens.

Os resultados apresentados nas secções anteriores foram obtidos através de *scripts* do Matlab mas tinham o grande inconveniente de serem bastante lentos. Apenas a título indicativo, num computador pessoal com um processador Pentium III de 800MHz e com 444MBytes de memória RAM, o processamento necessário até à criação da imagem normalizada (saída do bloco imediatamente anterior ao do ICA) leva em média cerca de 14 segundos por cada imagem de dimensões 376×287. O que vem demonstrar a inadequação do Matlab para executar processamento de imagem em tempo-real.

Por conseguinte, recorreu-se à linguagem de programação C++ para se obter melhor *performance* e assim ser possível executar a tarefa pretendida. Além disso houve ainda outros cuidados como é o caso de fazer concatenação de linhas e não de colunas, na transformação de matriz imagem para vector imagem, visto que em memória as imagens estão armazenadas já dessa forma e então a transformação é imediata. Contudo, nem sempre as linhas da imagem estão armazenadas em espaços contíguos de memória, isso só acontece quando não há enchimento de bytes. O enchimento de bytes é uma técnica que é utilizada, por razões de aumento da velocidade de processamento devido à forma como os processadores lêem da memória, para garantir que em memória o início de cada linha de uma imagem se situe num endereço múltiplo de 4. Então, caso a imagem tenha uma codificação que implique que o número de bytes necessário para representar uma linha completa não seja múltiplo de 4, ocorrerá um enchimento de bytes no final da linha para que estas fiquem alinhadas em memória. Por esta razão, optou-se preferencialmente por capturar imagens digitais com largura múltipla de 4,

evitando dessa forma o enchimento de bytes e por conseguinte aumentando a performance da transformação matriz imagem em vector imagem.

Outra decisão que se tomou, para melhorar a performance do demonstrador, foi recorrer às bibliotecas de Intel IPP (Integrated Performance Primitives) [Intel, IPP] e ao OpenCV (Open Source Computer Vision) [Intel, OpenCV]. As bibliotecas IPP são um produto comercializado pela Intel que disponibiliza um conjunto de funções de baixo nível para processamento de sinal, processamento de imagem e cálculo matricial, especialmente optimizadas para os processadores Intel: Pentium, Pentium II, Pentium III, Pentium 4. Por outro lado, OpenCV é uma biblioteca de domínio público, lançada pela Intel, que disponibiliza funções de mais alto nível para processamento de imagem e que, ao contrário das IPP, já não é reservada apenas para processadores Intel. Caso estejam presentes as bibliotecas IPP no computador onde é executado o programa, o OpenCV irá utilizá-las, caso contrário utilizará rotinas desenvolvidas em código C optimizado.

Com este ambiente de desenvolvimento foi possível criar com sucesso o demonstrador (Figura 5.11) funcionando até em tempo-real.

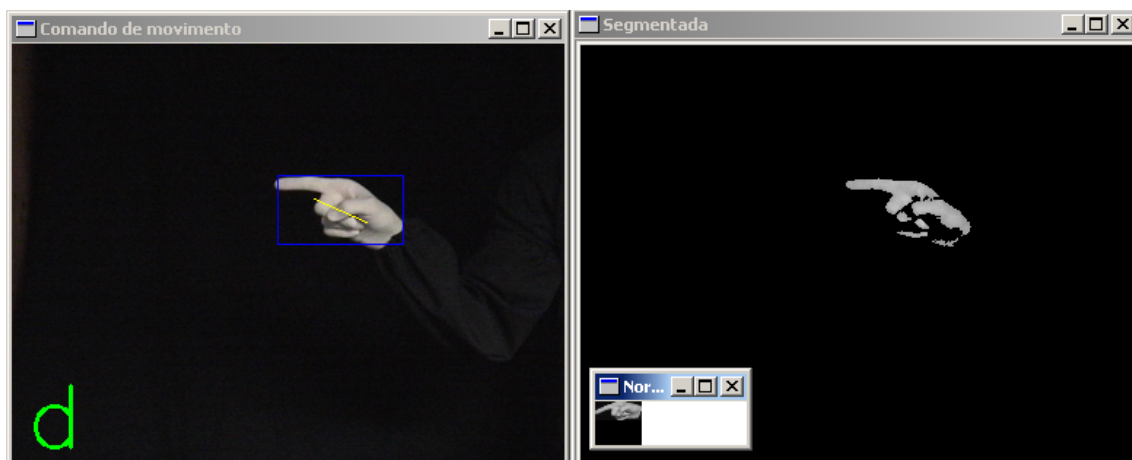


Figura 5.11 – Demonstrador de comando de movimento por gestos simbólicos

A velocidade de processamento é tal que se consegue até 25fps (*frames per second*) num Pentium III a 800MHz com 444MBytes de RAM, para processar um ficheiro AVI codificado em HuffYUV com imagens de dimensão 376×287. Num outro computador mais potente, Pentium 4 a 2,6GHz com 512 MBytes de RAM, é possível ir a mais de 90fps! O que vem demonstrar que, ao contrário que do que se poderia pensar, o ICA não impede processamento em tempo-real já que em funcionamento corrente consiste apenas numa multiplicação de uma matriz por um vector.

Como a Arquitectura I do ICA é a mais indicada para o reconhecimento de gestos, foi esta que se utilizou no demonstrador. Relativamente à fase de treino, esta é realizada no Matlab e o seu resultado, a matriz de transformação do espaço original para o espaço ICA, é fornecida ao demonstrador a fim de executar o reconhecimento dos gestos simbólicos no espaço ICA. Como no demonstrador será feita uma concatenação de linhas, no treino realizado em Matlab teve-se de fazer o mesmo tipo de concatenação por forma a ter uma matriz de transformação adequada.

Uma vez reconhecidos os gestos resta criar um modelo de movimento a partir dos 5 gestos existentes. Definiu-se então que o virar para a esquerda e virar para a direita correspondem a uma rotação da direcção em que o objecto se move. Por isso, sempre que um gesto ‘direita’ ou ‘esquerda’ é detectado, a direcção do objecto irá sofrer uma variação angular (de valor definido consoante a aplicação específica em causa) no sentido horário ou anti-horário, respectivamente. De uma forma análoga funcionará o ‘acelerar’ e o ‘travar’. Seguindo-se um modelo de aceleração constante, por cada detecção do gesto ‘acelerar’ será incrementada a velocidade do objecto enquanto que o gesto ‘travar’ causará um decremento de igual valor absoluto. Partindo da equação da velocidade (4.14), o valor absoluto do incremento ou decremento de velocidade será igual ao produto da aceleração (valor dependente da aplicação pretendida) pelo período de tempo em que se aplica o gesto detectado.

$$v = v_0 + a t \quad (4.14)$$

Como o processamento é executado imagem a imagem, o período de tempo em que se aplica o gesto corresponderá ao intervalo entre imagens, ou seja, o inverso da taxa de imagens por segundo que se está a processar.

Para evitar instabilidade nas fases de preparação e retracção do gesto (secção 1.1), e também para evitar classificações erradas, caso a semelhança do gesto à imagem de referência mais próxima seja inferior a 0,88 o gesto é classificado como desconhecido. Em aplicações mais exigentes que não possam suportar qualquer instabilidade na transição entre gestos, uma solução poderá passar por tomar a acção só após ocorrerem um conjunto de imagens consecutivas em que é detectado o mesmo gesto.

5.8.1 Sistema de reconhecimento da contagem gestual chinesa

De forma a averiguar qual o comportamento do ICA num caso em que há mais classes do que as dos 5 gestos simbólicos utilizados neste trabalho, e também num caso em que o classificador *ad hoc* apresentado na secção anterior é incapaz de classificar correctamente, treinou-se o sistema para reconhecer 10 gestos frequentemente utilizados pela comunidade chinesa para efectuar a contagem de 1 a 10 com uma só mão.

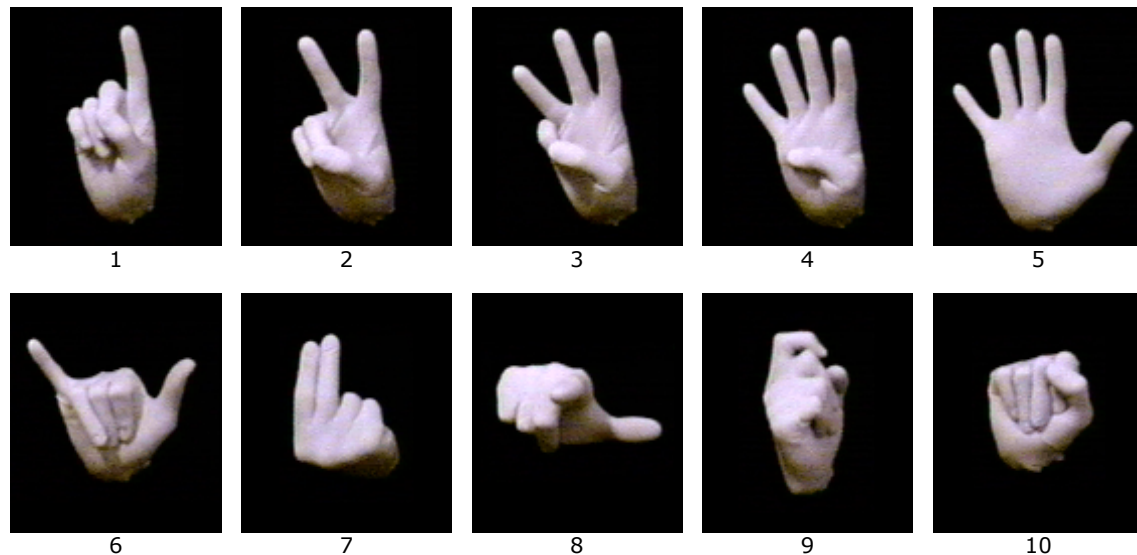


Figura 5.12 – Contagem gestual chinesa

Assim treinou-se o sistema com 25 imagens de cada gesto e testou-se com 30 imagens de cada gesto utilizando apenas 1 ou 3 imagens de referência por cada classe existente. O resultado obtido está expresso nas tabelas seguintes.

	T_{acerto}	S_{1^a}	S_{2^a}
Arquitectura I	98,7%	$0,962 \pm 0,032$	$0,612 \pm 0,021$
Arquitectura II	98,0%	$0,930 \pm 0,050$	$0,656 \pm 0,031$

Tabela 5.15 – Desempenho com 1 imagem de referência por gesto

	T_{acerto}	S_{1^a}	S_{2^a}
Arquitectura I	99,0%	$0,970 \pm 0,026$	$0,619 \pm 0,024$
Arquitectura II	99,0%	$0,942 \pm 0,039$	$0,673 \pm 0,025$

Tabela 5.16 – Desempenho com 3 imagem de referência por gesto

Conclui-se que, mesmo aumentando o número de classes com o ICA consegue-se obter bons resultados enquanto que com o classificador *ad hoc*, que foi criado heurísticamente

especialmente para o reconhecimento de 5 gestos simbólicos, isso não é possível a não ser que seja totalmente reconstruído.

CAPÍTULO 6

Conclusões e Trabalho futuro

6.1 Conclusões

Findo este trabalho, a principal conclusão que se extrai é que a utilização de ICA segundo a Arquitectura I, secundado por um classificador do vizinho mais próximo que utiliza a medida de semelhança do cosseno, é capaz de reconhecer um grupo de gestos simbólicos em número suficiente para controlar o movimento de um objecto num plano. Atendendo a que as classes de gestos existentes são pouco numerosas, talvez não fosse necessário realizar o reconhecedor baseado numa metodologia tão complexa quanto o ICA. No entanto, o ICA apresenta maior potencial do que um reconhecedor baseado apenas em análise discriminante e em heurísticas para lidar com casos mais complexos. O verdadeiro valor de um reconhecedor de gestos simbólicos baseado no ICA só poderá ser confirmado quando se tiver um grande número de classes de gestos.

A boa imunidade às condições de captura é uma característica a salientar no ICA. Gestos capturados em ambientes de diferente iluminação e/ou com câmaras de diferentes sensibilidades são bem tolerados pelo ICA, o que se deve à forma como o ICA foi formulado garantindo baixa sensibilidade a ruído/*outliers*.

Num sistema real previsto para ser usado por diferentes utilizadores é preciso alguma precaução porque um reconhecedor de gestos baseado no ICA não é independente do utilizador. O ideal será treinar e seleccionar imagens de referência dos diferentes utilizadores, o que num caso real pode não ser praticável por não ser possível prever quem irá usar o sistema. Assim uma alternativa mais viável será não utilizar apenas uma imagem de referência por cada gesto. Quantas mais imagens de referência se utilizar melhor, pois é mais provável existir uma que se aproxime da forma como determinado utilizador gesticula. Porém, isto terá de passar por uma solução de compromisso, porque aumentar em demasia o número de imagens de referência pode tornar o processo demasiado pesado, já que uma nova imagem apresentada ao sistema necessitará de ser comparada com todas as imagens de referência.

Quanto ao peso computacional do ICA há a referir que em funcionamento corrente não é tão pesado como à primeira vista possa parecer porque basicamente consiste numa multiplicação de um vector por uma matriz, tarefa que é de tal forma genérica que poderá ser realizada por hardware, o que melhora a performance da multiplicação. Em modo de treino o ICA é um pouco pesado mas isso não é relevante porque neste modo não há imperativos de tempo-real tal como existem em modo corrente. Se o treino demorar um pouco mais isso não é muito importante, porque o essencial é que em funcionamento corrente seja suficientemente rápido a fim de permitir, por exemplo, o controlo de um objecto, que só é viável se for executado em tempo-real.

Uma parte bastante significativa do peso computacional do reconhecedor desenvolvido encontra-se na etapa de segmentação de imagem. Apesar de neste trabalho se ter imposto algumas restrições no cenário de captura (*i.e.* mão equipada com luva branca sobre fundo preto) mesmo assim foram necessários processamentos com algum peso computacional a fim de localizar devidamente a mão nos casos em que existem pequenas sombras causadas por diferentes partes da mão e, também, porque é difícil que o gesticulador não apareça no ângulo de visão da câmara. É importante salientar que o bloco de segmentação de imagem, por ser um dos primeiros na cadeia de processamento e dada a sua complexidade, é o bloco que exige maior atenção porque se cometer um erro grosseiro inviabiliza por completo uma classificação correcta.

Apesar disto, o objectivo que se tinha de realizar um reconhecedor dos 5 gestos simbólicos definidos para este trabalho foi suplantado, visto que se conseguiu alcançar o funcionamento em tempo-real. Num computador actual, o sistema desenvolvido

consegue processar 25 a 30 imagens por segundo sem ocupar a 100% os recursos computacionais do computador!

No decorrer deste trabalho houve um facto importante a assinalar que foi a inexistência de um trabalho semelhante a este que permitisse comparar resultados. Isto deve-se principalmente a dois factores. O primeiro é, do que conseguimos constatar, não haver ainda aplicações de ICA em reconhecimento de gestos devido a ser uma técnica recente. O segundo factor prende-se com a inexistência de uma base de dados de gestos simbólicos que permita que diversos investigadores analisem a aplicação de diferentes metodologias aos mesmos casos de estudo.

Por estas razões, este trabalho consegue dar o contributo de apresentar resultados da aplicação do ICA em reconhecimento de gestos simbólicos que poderá servir de termo de comparação para outros trabalhos. E, além disso, dá boas perspectivas da utilização de ICA em aplicações mais complexas do que a que foi aqui abordada.

6.2 Trabalho futuro

No trabalho relatado nesta dissertação o caso de estudo tinha o pressuposto de as capturas se fazerem sobre um fundo preto e a mão ser equipada com uma luva branca a fim de facilitar a segmentação. Tinha-se assim um ambiente controlado, o que na realidade pode não ser exequível porque, por exemplo, é bastante desconfortável obrigar os utilizadores a equiparem-se com uma luva especial e a vestirem roupa escura. Logo um trabalho muito interessante será a segmentação em ambientes não controlados, que é ainda um problema em aberto e de bastante complexidade, que só por si é tema suficiente para um trabalho de dissertação. Além disso, já há garantias que o processamento a jusante, parte da classificação, é realizado com sucesso quando a segmentação, a parte situada a montante, cumpre devidamente as suas funções. Logo com uma segmentação correcta em ambientes não controlados será possível desenvolver reconhecedores de gestos com as mais variadas aplicações.

Outro trabalho interessante e também importante, é analisar o comportamento do ICA quando se têm mais classes de gesto, por exemplo, mais de 20 ou 30 classes. Dessa forma aproxima-se do número de gestos da linguagem gestual podendo-se testar a exequibilidade do desenvolvimento de um tradutor de linguagem gestual.

Tendo-se bastante mais gestos do que os que foram aqui utilizados também se justifica averiguar com maior rigor qual a influência da dimensão dos conjuntos de treino no resultado final e averiguar quando ocorre *overlearning* no ICA.

Referências

- Ahmad, S. (1994). "A Usable Real-Time 3D Hand Tracker." Conference Record of the Twenty-Eighth Asilomar Conference on Signals, Systems and Computers, Vol.2, 1257-1261 vol.2, 1994.
- Assan, M. and K. Grobel (1998). "Video-Based Sign Language Recognition Using Hidden Markov Models." Gesture and Sign Language in Human-Computer Interaction: International Gesture Workshop, Vol.1371, 1998.
- Bartlett, M. S., H. M. Lades, et al. (1998). "Independent Component Representations for Face Recognition." Proceedings of the SPIE Symposium on Electronic Imaging: Science and Technology, January 1998.
- Bartlett, M. S., J. R. Movellan, et al. (2002). "Face Recognition by Independent Component Analysis." IEEE Transactions on Neural Networks, Vol.13(6), November 2002.
- Bauer, B. and H. Hienz (2000). "Relevant Features for Video-Based Continuous Sign Language Recognition." Fourth IEEE International Conference on Automatic Face and Gesture Recognition, 440-445, 2000.
- Becker, D. and A. Pentland (1997). "Using A Virtual Environment to Teach Cancer Patients T'ai Chi, Relaxation and Self-Imagery." ACM Siggraph Symposium on Interactive 3D Graphics, April 1997.
- Bell, A. J. and T. J. Sejnowski (1995). "Information-Maximization Approach to Blind Separation and Blind Deconvolution." Neural computation, Vol.7(6), 1129, November 1995.
- Birk, H. and T. B. Moeslund (1996), "Recognizing Gestures From the Hand Alphabet Using PCA," Master's Thesis, 1996.
- Bobick, A. F. and A. D. Wilson (1997). "State-Based Approach to the Representation and Recognition of Gesture." IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.19(12), 1325-1337, 1997.
- Cipolla, R. and N. J. Hollinghurst (1996). "Human-Robot Interface by Pointing with Uncalibrated Stereo Vision." Image and Vision Computing, Vol.14(3), 171-178, 1996.

- Cover, T. M. and J. A. Thomas (1991). Elements of Information Theory, John Wiley & Sons.
- Cui, Y. and J. Weng (1999). "A Learning-Based Prediction-and-Verification Segmentation Scheme for Hand Sign Image Sequence." IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.21(8), 798-804, 1999.
- Darrell, T., I. A. Essa, et al. (1996). "Task-Specific Gesture Analysis in Real-Time Using Interpolated Views." IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.18(12), 1236-1242, 1996.
- Davis, J. and M. Shah (1995). "Recognizing Hand Gestures." Proceedings of the International Workshop on Automatic Face and Gesture Analysis Recognition, June 1995.
- Davis, J. W. and A. F. Bobick (1998). "Virtual PAT: Virtual Personal Aerobics Trainer." Proceedings of the Workshop on Perceptual User Interfaces, 13-18, 1998.
- Draper, B. A., K. Baek, et al. (2003). "Recognizing Faces with PCA and ICA." Computer Vision and Image Understanding, Vol.91(1), 115-137, August 2003.
- Freeman, W. T. and M. Roth (1995). "Orientation Histogram for Hand Gesture Recognition." Int'l Workshop on Automatic Face and Gesture-Recognition, June 1995.
- Gao, W., J. Ma, et al. (2000). "Sign language recognition based on HMM/ANN/DP." International Journal of Pattern Recognition and Artificial Intelligence, Vol.14(5), 587-602, 2000.
- Haralick, R. M. and L. G. Shapiro (1992). Computer and Robot Vision, Addison-Wesley Publishing Company, Inc.
- Hyvärinen, A. (1997). "New Approximations of Differential Entropy for Independent Component Analysis and Projection Pursuit." Advances in Neural Information Processing Systems, Vol.10, 273-279, August 1997.
- Hyvärinen, A. (1999). "Survey on Independent Component Analysis." Neural Computing Surveys, Vol.2, 94-128, April 1999.
- Hyvärinen, A., J. Karhunen, et al. (2001). Independent Component Analysis, John Wiley & Sons Inc.
- Hyvärinen, A. and E. Oja (1997). "A Fast Fixed-Point Algorithm for Independent Component Analysis." Neural Computation(9(7)), 1483-1492, 1997.
- Hyvärinen, A. and E. Oja (1999a). "Fast and Robust Fixed-Point Algorithms for Independent Component Analysis." IEEE Transactions on Neural Networks(10(3)), 626-634, 1999a.
- Hyvärinen, A. and E. Oja (1999b). Independent Component Analysis: A Tutorial.
<http://www.cis.hut.fi/~aapo/ps/NN00.pdf>, April 1999b.
- Imagawa, K., S. Lu, et al. (1998a). "Color-Based Hands Tracking System for Sign Language Recognition." Third IEEE International Conference on Automatic Face and Gesture Recognition, 462-467, 1998a.

- Imagawa, K., S. Lu, et al. (1998b). "Color-Based Hands Tracking System for Sign Language Recognition." IEEE International Conference on Automatic Face and Gesture Recognition, 462-467, 1998b.
- Intel IPP - Integrated Performance Primitives. <http://www.intel.com/software/products/ipp/>.
- Intel OpenCV - Open Source Computer Vision Library.
<http://www.intel.com/research/mrl/research/opencv>.
- Ju, S. X., M. J. Black, et al. (1997). "Analysis of Gesture and Action in Technical Talks for Video Indexing." IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 595-601, 1997.
- Kang, S. B. and K. Ikeuchi (1997). "Toward Automatic Robot Instruction From Perception-Mapping Human Grasps to Manipulator Grasps." IEEE Transactions on Robotics and Automation, Vol.13(1), 81-95, 1997.
- Kendon, A. (1986). The Biological Foundation of Gestures: Motor and Semiotic Aspects, Lawrence Erlbaum Associate.
- Lello, J. L. e. E. (1974). Dicionário Enciclopédico Luso-Brasileiro Lello Universal, Lello & Irmão Editores.
- Liang, R.-H. and M. Ouhyoung (1998). "A Real-Time Continuous Gesture Recognition System for Sign Language." Third IEEE International Conference on Automatic Face and Gesture Recognition, 558-567, 1998.
- Matsumoto, K., K. Shimada, et al. (2002). "Interface Design and Implementation for Mono-Functional Modular Robots." IEEE/RSJ International Conference on Intelligent Robots and Systems, Vol.3, 3018-3023, September 2002.
- Michael Stark, M. K. (1995), "Video Based Gesture Recognition for Human Computer Interaction," University of Dortmund 593, 1995.
- O'Hagan, R., A. Zelinsky, et al. (2000). "Visual Gesture Interfaces for Virtual Environments." First Australasian User Interface Conference, January 2000.
- Parashar, A. S. (2003), "Representation and Interpretation of Manual and Non-Manual Information for Automated American Sign Language," College of Engineering - University of South California, Master's Thesis, 2003.
- Pausch, R. and R. D. Williams (1992). "Giving Candy to Children - User-Tailored Gesture Input Driving an Articulator-Based Speech Synthesizer." Communications of the ACM, Vol.35(5), 58-66, 1992.
- Pavlovic, V. I., R. Sharma, et al. (1997). "Visual Interpretation of Hand Gestures for Human-Computer Interaction: A Review." IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.19(7), July 1997.

- Schlenzig, J., E. Hunter, et al. (1994). "Vision Based Hand Gesture Interpretation Using Recursive Estimation." Proceedings of the 28th Asilomar Conference on Signals, Systems & Computers, October 1994.
- Segen, J. and S. Kumar (1998). "Gesture VR: Vision-Based 3D Hand Interface for Spatial Interaction." Proceedings of the sixth ACM international conference on Multimedia, 1998.
- Starner, T. (1995), "Visual Recognition of American Sign Language Using Hidden Markov Models," Master's Thesis, 1995.
- Starner, T. and A. Pentland (1995), "Visual Recognition of American Sign Language Using Hidden Markov Models," Massachusetts Institute of Technology - Perceptual Computing Section, The Media Laboratory TR 316, 1995.
- Starner, T., J. Weaver, et al. (1998). "Real-Time American Sign Language Recognition Using Desk and Wearable Computer Based Video." IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.20(12), 1371-1375, 1998.
- Sturman, D. J. and D. Zeltzer (1994). "A Survey of Glove-Based Input." IEEE Computer Graphics and Applications, Vol.14(1), 30-39, 1994.
- Sturman, D. J. (1992), "Whole-Hand Input," Massachusetts Institute of Technology, Phd Thesis, 1992.
- Sweeney, G. J. and A. C. Downton (1997). "Sign Language Recognition Using a Cheremic Architecture." Proceedings of the 6th International Conference on Image Processing and its Applications(443 pt 2), 483-486, July 1997.
- Swets, D. L. and J. Weng (1996). "Using Discriminant Eigenfeatures for Image Retrieval." IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.18(8), 831-836, August 1996.
- Tan, K.-H., D. J. Kriegman, et al. (2002). "Appearance-Based Eye Gaze Estimation." IEEE Workshop on Applications of Computer Vision, December 2002.
- Teague, M. R. (1980). "Image Analysis Via the General Theory of Moments." Journal of the Optical Society of America, 920-930, August 1980.
- Teh, C.-H. and R. T. Chin (1988). "On Image Analysis by the Methods of Moments." IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.10(4), 496-513, July 1988.
- Vogler, C. and D. Metaxas (1998). "ASL Recognition Based on a Coupling Between HMMs and 3D Motion Analysis." Proceedings of the 6th IEEE International Conference on Computer Vision, 363-369, January 1998.
- Vogler, C. and D. Metaxas (1999). "Parallel Hidden Markov Models for American Sign Language Recognition." Proceedings of the 7th IEEE International Conference on Computer Vision (ICCV'99), Vol.1, 116-122, September 1999.

- Vogler, C. and D. Metaxas (2001). "A Framework for Recognizing the Simultaneous Aspects of American Sign Language." Computer Vision and Image Understanding, Vol.81(3), 358-384, 2001.
- Wang, C.-L., W. Gao, et al. (2002). "Real-Time Large Vocabulary Continuous Recognition System for Chinese Sign Language." Jisuanji Xuebao/Chinese Journal of Computers, Vol.25(6), 624-631, 2002.
- Watson, R. (1993), "A Survey of Gesture Recognition Techniques," Department of Computer Science - Trinity College TCD-CS-93-11, 1993.
- Wu, Y. and T. S. Huang (1999). "Vision-Based Gesture Recognition: A Review." Lecture Notes in Computer Science, Vol.1739, 103-?, 1999.
- Yang, J. and A. Waibel (1996). "Real-Time Face Tracker." Proceedings of the IEEE Workshop on Applications of Computer Vision, 1996.
- Zeller, M., J. C. Phillips, et al. (1997). "A Visual Computing Environment for Very Large Scale Biomolecular Modeling." IEEE International Conference on Application-Specific Systems, Architectures and Processors, 3-12, 1997.