

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

YESSENIA DEYSI YARI RAMOS

**Estimativa da pose da cabeça em imagens
monoculares usando um modelo
no espaço 3D**

Dissertação apresentada como requisito parcial
para a obtenção do grau de
Mestre em Ciência da Computação

Prof. Dr. Jacob Scharcanski
Orientador

Porto Alegre, junho de 2013

CIP – CATALOGAÇÃO NA PUBLICAÇÃO

Yari Ramos, Yessenia Deysi

Estimativa da pose da cabeça em imagens monoculares usando um modelo no espaço 3D / Yessenia Deysi Yari Ramos. – Porto Alegre: PPGC da UFRGS, 2013.

62 f.: il.

Dissertação (mestrado) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, BR–RS, 2013. Orientador: Jacob Scharcanski.

1. Pose da cabeça. 2. Modelo da face em 3D. 3. Modelo ativo de forma. 4. Imagens monoculares. I. Scharcanski, Jacob. II. Título.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. Carlos Alexandre Netto

Pró-Reitor de Coordenação Acadêmica: Prof. Rui Vicente Oppermann

Pró-Reitor de Pós-Graduação: Prof. Vladimir Pinheiro do Nascimento

Diretor do Instituto de Informática: Prof. Luis da Cunha Lamb

Coordenador do PPGC: Prof. Álvaro Freitas Moreira

Bibliotecária-chefe do Instituto de Informática: Beatriz Regina Bastos Haro

"Aos meus pais Rubert e Deysi."

AGRADECIMENTOS

Ao longo de nossas vidas sempre existem pessoas que são difíceis de esquecer, elas brindam-nos seu apoio, companheirismo, dando-nos forças para alcançar nossos objetivos, pois sem o auxílio delas nada teria sido possível.

Primeiramente, agradeço a Deus, por estar sempre ao meu lado, sendo esse Pai incondicional que ilumina meu caminho.

Aos meus pais Rubert e Deysi, pela educação brindada, pelo apoio irrestrito em todos os momentos de minha vida, em especial quando decidi me tornar mestre.

À minha irmã Yaneth, por me escutar e aconselhar cada vez que eu tinha algum problema.

Ao meu orientador, Professor Dr. Jacob Scharcanski, pelo seu permanente apoio, sua disponibilidade, sua forma exigente e crítica fundamental no meu crescimento como pesquisadora.

Aos meus colegas e amigos da Universidade Federal do Rio Grande do Sul, os quais foram de grande apoio pra mim.

As pessoas maravilhosas que conheci aqui em Porto Alegre, muito obrigada pelos momentos inesquecíveis.

Ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), pelo apoio financeiro.

SUMÁRIO

LISTA DE ABREVIATURAS E SIGLAS	7
LISTA DE FIGURAS	8
LISTA DE TABELAS	10
RESUMO	11
ABSTRACT	12
1 INTRODUÇÃO	13
1.1 Considerações iniciais	13
1.2 Conceitos sobre a estimativa da pose da cabeça	13
1.3 Objetivo deste trabalho	15
1.4 Organização do documento	16
2 MARCO TEÓRICO	17
2.1 Revisão bibliográfica sobre a detecção da pose da cabeça	17
2.2 Modelo ativo de forma	22
2.2.1 Modelos de distribuição de pontos	23
2.2.2 Modelo da aparência dos níveis de cinza	27
2.2.3 Localização dos pontos característicos numa nova imagem	28
2.3 Conclusões	30
3 MÉTODO PROPOSTO	33
3.1 Considerações iniciais	33
3.2 Detecção de características faciais	34
3.3 Cálculo dos ângulos de movimento	36
3.3.1 Estimativa do ângulo γ	36
3.3.2 Estimativa dos ângulos α e β	37
3.4 Refinamento do modelo	38
3.5 Conclusões	41
4 EXPERIMENTOS E RESULTADOS	43
4.1 Considerações iniciais	43
4.2 Primeiro experimento	44
4.3 Segundo experimento	46
5 CONCLUSÕES E TRABALHOS FUTUROS	51

REFERÊNCIAS	53
APÊNDICE A ALGORITMO DE VIOLA JONES	57
APÊNDICE B TRANSFORMAÇÕES GEOMÉTRICAS	60
B.1 Transformações geométricas em 2D	60
B.2 Transformações geométricas em 3D	61

LISTA DE ABREVIATURAS E SIGLAS

ASM	Active Shape Model
AAM	Active Appearance Model
GWN	Gabor Wavelet Network
HCI	Human Computer Interaction
NMWF	Non-negative Multi-way Factorization
LGBP	Local Gabor Binary Patterns
LBP	Local Binary Pattern
SVM	Singular Vector Machine
POSIT	Pose from Orthography and Scaling with Iterations
PCA	Principal Components Analysis
LGL	Local Gray-Level
PDM	Point Distribution Model
KPCA	Kernel Principal Component Analysis
MLP	Multilayer Perceptron
PNMF	Poses-based Non-negative Matrix Factorization
BSCR	Block based Sparse Representation Classifier
SCI	Sparsity Concentration Index
RANSAC	RANdom SAmples Consensus
ASD	Allowable Shape Domain
SCR	Sparse Representation Classifier
HOSVD	High Order Singular Value Decomposition
CCA	Canonical Correlation Analysis
PC	Phase Congruency
AdaBoost	Adaptive Boosting
LDA	Linear Discriminative Analysis

LISTA DE FIGURAS

Figura 1.1:	Movimentos de rotação da cabeça: <i>roll</i> , <i>yaw</i> e <i>pitch</i>	14
Figura 1.2:	Estimativa da pose da cabeça: (a) Imagem monocular, (b) Modelo em 3D o qual representa a pose da cabeça.	15
Figura 2.1:	Exemplo de pontos principais em imagens médicas (a-b). Fonte (BOOKSTEIN, 1989)	24
Figura 2.2:	Modelo de 32 pontos de um resistor. Fonte (BOOKSTEIN, 1989)	24
Figura 2.3:	Identificação da face humana usando o algoritmo de Viola-Jones, (VIOLA; JONES, 2001).	29
Figura 3.1:	Movimentos do modelo 3D da face: (a)roll, (b)yaw e (c)pitch.	33
Figura 3.2:	Movimentos da cabeça humana que podem ser mostrados em imagens mono- culares.	33
Figura 3.3:	Pontos usados para estimar a pose da cabeça, (GOURIER; HALL; CROWLEY, 2004).	34
Figura 3.4:	Modelo 3D da face: (a) Modelo CANDIDE; (b) Modelo da face modificado	35
Figura 3.5:	Estimativa do ângulo γ_A	37
Figura 3.6:	(a) Localização dos pontos na face,(b) máscara a ser usada para a estimativa da pose.	39
Figura 3.7:	Estimativa da melhor pose da cabeça. As figuras (a), (c), (e), (g) mostram a face com os pontos característicos marcados em cor amarelo, e os pontos da máscara sobre ela marcados em cor vermelho. As figuras (b), (d), (f), (h) mostram as possíveis máscaras que minimizam as distâncias entre os pontos encontrados na face e os pontos projetados sobre ela; (b) R: $\alpha_A = -3^\circ$, $\beta_A = -6^\circ$, $\gamma_A = -1^\circ$ and R_N : $\alpha' = 0.0014^\circ$, $\beta' = 0.0038^\circ$, $\gamma' = 0.0309^\circ$. (d) R: $\alpha_A = -5^\circ$, $\beta_A = -10^\circ$, $\gamma_A = 1^\circ$ e R_N : $\alpha' = 0.0391^\circ$, $\beta' = 0.0443^\circ$, $\gamma' = 0.0021^\circ$. (f) R: $\alpha_A = -4^\circ$, $\beta_A = 4^\circ$, $\gamma_A = -3^\circ$ e R_N : $\alpha' = 0.0148^\circ$, $\beta' = 0.0930^\circ$, $\gamma' = 0.0522^\circ$. (h) R: $\alpha_A = 0^\circ$, $\beta_A = -9^\circ$, $\gamma_A = 1^\circ$ e R_N : $\alpha' = 0.0494^\circ$, $\beta' = 0.0276^\circ$, $\gamma' = 0.0040^\circ$	40
Figura 3.8:	Ilustração da estimativa da pose da cabeça: (a) máscara final encontrada e ajustada à face(b) máscara com os ângulos $\alpha = -2.99^\circ$, $\beta = -5.99^\circ$ e $\gamma = -0.97^\circ$	41
Figura 4.1:	Imagens usadas para o teste do método	44

Figura 4.2:	Ilustração do resultado da estimativa da pose da cabeça: (a) $mask_f$ ajustada à imagem da face. (b) $mask_f$ com ângulos: $\alpha = 3.88^\circ$, $\beta = 1.89^\circ$, $\gamma = 0.02^\circ$	45
Figura 4.3:	Ilustração do resultado da estimativa da pose da cabeça: (a) $mask_f$ ajustada à imagem da face. (b) $mask_f$ com ângulos: $\alpha = 2.84^\circ$, $\beta = -16.07^\circ$, $\gamma = 0.03^\circ$	45
Figura 4.4:	Ilustração do resultado da estimativa da pose da cabeça: (a) $mask_f$ ajustada à imagem da face. (b) $mask_f$ com ângulos: $\alpha = -73.34^\circ$, $\beta = 1.99^\circ$, $\gamma = 1.04^\circ$	45
Figura 4.5:	Ilustração do resultado da estimativa da pose da cabeça:(a) $mask_f$ ajustada à face. (b) $mask_f$ com os ângulos: $\alpha = -10.71^\circ$, $\beta = 1.04^\circ$, $\gamma = -2.96^\circ$	46
Figura 4.6:	Ilustração do resultado da estimativa da pose da cabeça:(a) $mask_f$ ajustada à face. (b) $mask_f$ com os ângulos: $\alpha = 7.98^\circ$, $\beta = 1.03^\circ$, $\gamma = 0.97^\circ$	46
Figura 4.7:	Ilustração do resultado da estimativa da pose da cabeça:(a) $mask_f$ ajustada à face. (b) $mask_f$ com os ângulos: $\alpha = 4.84^\circ$, $\beta = -20.69^\circ$, $\gamma = 1.84^\circ$	47
Figura 4.8:	Ilustração do resultado da estimativa da pose da cabeça:(a) $mask_f$ ajustada à face. (b) $mask_f$ com os ângulos: $\alpha = -13.03^\circ$, $\beta = 24.97^\circ$, $\gamma = 1.18^\circ$	47

LISTA DE TABELAS

Tabela 2.1:	Classificação de métodos segundo Murphy-Chutorian	22
Tabela 4.1:	Resultados do erro produzido nos ângulos <i>pitch</i> , <i>yaw</i> e <i>roll</i> usando inicialização manual e automática.	48
Tabela 4.2:	Resultados do erro produzido nos ângulos <i>pitch</i> , <i>yaw</i> e <i>roll</i> usando <i>cross – validation</i>	48
Tabela 4.3:	Resultados do desvio padrão produzido nos eixos X, Y e Z.	49
Tabela 4.4:	Resultados do erro produzido nos ângulos <i>pitch</i> , <i>yaw</i> e <i>roll</i>	49
Tabela 4.5:	Resultados dos erros obtidos por outros métodos.	49

RESUMO

Esta dissertação apresenta um novo método para cálculo da pose da cabeça em imagens monoculares. Este cálculo é estimado no sistema de coordenadas da câmera, comparando as posições das características faciais específicas com as de múltiplas instâncias do modelo da face em 3D.

Dada uma imagem de uma face humana, o método localiza inicialmente as características faciais, como nariz, olhos e boca. Estas últimas são detectadas e localizadas através de um modelo ativo de forma para faces. O algoritmo foi treinado sobre um conjunto de dados com diferentes poses de cabeça.

Para cada face, obtemos um conjunto de pontos característicos no espaço de imagem 2D. Esses pontos são usados como referências na comparação com os respectivos pontos principais das múltiplas instâncias do nosso modelo de face em 3D projetado no espaço da imagem. Para obter a profundidade de cada ponto, usamos as restrições impostas pelo modelo 3D da face por exemplo, os olhos tem uma determinada profundidade em relação ao nariz. A pose da cabeça é estimada, minimizando o erro de comparação entre os pontos localizados numa instância do modelo 3D da face e os localizados na imagem.

Nossos resultados preliminares são encorajadores e indicam que a nossa abordagem produz resultados mais precisos que os métodos disponíveis na literatura.

Palavras-chave: Pose da cabeça, Modelo da face em 3D, Modelo ativo de forma, Imagens monoculares.

Estimation of the Head Pose Based on Monocular Images

ABSTRACT

This dissertation presents a new method to accurately compute the head pose in monocular images. The head pose is estimated in the camera coordinate system, by comparing the positions of specific facial features with the positions of these facial features in multiple instances of a prior 3D face model.

Given an image containing a face, our method initially locates some facial features, such as nose, eyes, and mouth; these features are detected and located using an Adaptive Shape Model for faces, this algorithm was trained using on a data set with a variety of head poses. For each face, we obtain a collection of feature locations (i.e. points) in the 2D image space.

These 2D feature locations are then used as references in the comparison with the respective feature locations of multiple instances of our 3D face model, projected on the same 2D image space. To obtain the depth of every feature point, we use the 3D spatial constraints imposed by our face model (i.e. eyes are at a certain depth with respect to the nose, and so on). The head pose is estimated by minimizing the comparison error between the 3D feature locations of the face in the image and a given instance of the face model (i.e. a geometrical transformation of the face model in the 3D camera space).

Our preliminary experimental results are encouraging, and indicate that our approach can provide more accurate results than comparable methods available in the literature.

Keywords: Head Pose, 3D Face Model, ASM, Monocular Images, Pattern Matching.

1 INTRODUÇÃO

1.1 Considerações iniciais

A análise da face humana é um dos temas de estudo nas áreas de visão computacional, processamento de imagens e robótica, dado a sua aplicação na autenticação biométrica. Esta análise tem como um objetivo a estimativa da pose da cabeça através de imagens monoculares, imagens estéreo ou uma sequência de imagens (a câmera com uma velocidade conhecida).

As interações entre os humanos e a humano-computador (HCI- *Human Computer Interaction*) estão baseadas na interpretação dos gestos e movimentos do corpo, tais como braços, mãos ou cabeça. Por exemplo, quando uma pessoa está conversando, provavelmente um dos primeiros movimentos é o de rotação da cabeça na direção da pessoa com que está falando. Outro exemplo é quando uma pessoa deseja dar ordens ao computador. Neste caso, é necessário que a pessoa rotacione a cabeça. Este movimento poderia ser interpretado pelo computador como um sinal para que alguma janela da área de trabalho se feche ou se abra, ou simplesmente para que ele termine alguma ação. Em todos os casos mencionados anteriormente é preciso obter a pose da cabeça no espaço 3D.

1.2 Conceitos sobre a estimativa da pose da cabeça

A estimativa da pose da cabeça na área de visão computacional é o processo realizado com o objetivo de obter a orientação da posição da cabeça humana em imagens digitais. Para isso precisamos de diferentes procedimentos que irão transformar uma representação baseada em pixel numa de alto nível de representação (*high-level*) (MURPHY; MANUBHAI, 2009). Obter a orientação da cabeça no espaço das coordenadas do sistema significa calcular os ângulos de rotação no espaço tridimensional (*pitch*, *yaw* e *roll*), os quais têm uma relação direta com os eixos X , Y , e Z , como se observa na Figura 1.1.

Segundo Sforza et al. (SFORZA et al., 2002) a amplitude dos movimentos da cabeça para um homem adulto inclui uma flexão e extensão sagital (movimento para a frente e para trás do pescoço) de -60.4° a 69.6° , uma dobra lateral frontal (direita para a esquerda, ou seja, flexão do pescoço) de -40.9° a 36.3° e uma rotação horizontal axial (à direita, ou seja, a rotação esquerda da cabeça) de -79.8° a 75.3° .

A abordagem apresentada neste trabalho é de grande interesse, já que existem diferentes aplicações que possuem como base a estimativa da pose da cabeça. Nestas aplicações existem interações humano-computador, como por exemplo, estimativa do estado de um condutor (BAKER et al., 2004), identificação dos gestos da cabeça (TRIVEDI, 2003), a simulação dos movimentos em robô (FUJIE et al., 2004), estimativa da pose da



Figura 1.1: Movimentos de rotação da cabeça: *roll*, *yaw* e *pitch*.

cabeça para uso em sistemas inteligentes de *air bag* (YANG; ZHANG; ZHU, 2008), identificação dos movimentos da cabeça durante uma conversa (DONG et al., 2007). Todas as aplicações abordadas anteriormente mostram que o tema a ser desenvolvido é de interesse e serve como base para aplicações que tragam benefícios às pessoas.

Para poder calcular a pose da cabeça é necessário obter informações no espaço 3D (nos eixo X, Y e Z). Com isso poderemos reconstruir a cabeça do indivíduo, porém é necessário equipamento especializado e um ambiente controlado.

Na literatura existem diferentes métodos para obtenção da pose da cabeça. Alguns autores os classificam segundo as características e propriedades que eles usam. Por exemplo, Dervinis (DERVINIS, 2006) classificou em duas categorias: baseadas em modelos e baseadas em propriedades. Neste trabalho, nos baseamos nas classificações propostas por Murphy-Chutorian et al (MURPHY; MANUBHAI, 2009) que serão descritas a seguir.

Os métodos com um modelo de aparência são aqueles que comparam um novo modelo da pose da cabeça com um conjunto de treinamento, onde cada uma das poses está agrupada e rotulada. A resposta será dada por aquela que tenha a pose mais similar. Geralmente são usadas comparações entre imagens ou comparações entre imagens filtradas (método de mínimos quadrados, *wavelets-gabor*, etc).

Os métodos com um vetor de detectores estimam a pose da cabeça, treinando um conjunto de múltiplos detectores de face, os quais especificam poses diferentes. A pose é dada pelo detector de maior suporte. Usam-se algoritmos de aprendizagem de máquinas (SVM) ou de redes neurais (*router networks*).

Os métodos de regressão linear usam métodos para desenvolver um mapeamento funcional da imagem ou de dados característicos da pose da cabeça. Estes métodos usam geralmente redes neurais (MLP - *Multilayer Perceptron*) e ferramentas de regressão.

Os métodos baseados em *manifold* procuram os *manifolds* de dimensionalidade mais baixa do modelo de pose da cabeça. As novas imagens são incorporadas dentro destes *manifolds* que serão usados, em seguida, para fazer a correlação. Trabalha-se com métodos de espaços sublineares (PCA - *Principal Component Analysis*), subespaços lineares (Isomap), etc.

Os métodos flexíveis ajustam um modelo não-rígido, grafo deformável que contém as características faciais, à estrutura facial de cada indivíduo. A pose da cabeça é calculada

comparando as características da face aos parâmetros do modelo. Os autores aplicam modelos ativos da forma (ASM - *Active Shape Model*) e descritores de características.

Os métodos geométricos usam a localização das características da face como olhos, boca, nariz, os quais determinam a pose da cabeça. Para isso os autores trabalham com geometria projetiva, pontos *vanishing* e métodos em 3D.

Os métodos de rastreamento trabalham seguindo o movimento da cabeça entre os *frames* consecutivos de uma seqüência de vídeo, usando características para o rastreamento, modelos de rastreamento (templates dinâmicos) e filtro de partículas baseadas em aparência.

Os métodos híbridos combinam um ou mais métodos anteriormente mencionados para a estimativa da posição da face. Podem ser combinados métodos geométricos e rastreamento, métodos de aparência e rastreamento, regressão não-linear e rastreamento, etc.

Neste trabalho usaremos métodos híbridos para estimar a pose da cabeça em imagens monoculares, (Figura 1.2a) as quais contém uma face da pessoa, em seguida localizamos pontos característicos (tais como olhos, nariz, boca, etc) que nos ajudam a estimar a pose da cabeça com a colaboração de um modelo da face em 3D (Figura 1.2b).

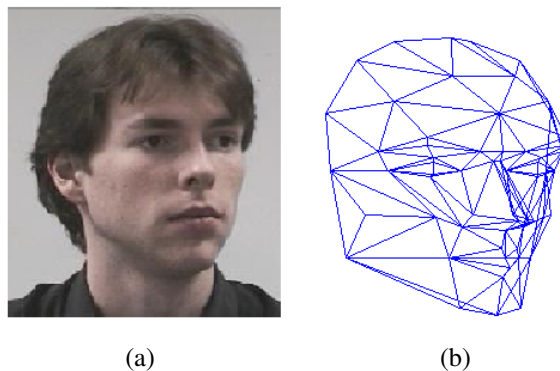


Figura 1.2: Estimativa da pose da cabeça: (a) Imagem monocular, (b) Modelo em 3D o qual representa a pose da cabeça.

O método proposto está desenvolvido para o uso de imagens monoculares, já que este tipo de imagens é mais fácil de obter através de uma câmera.

Este método é mais rápido porque não precisa do processamento de frames de vídeo ou imagens estéreo, os quais usam alguns métodos um pouco complexos e com um erro maior ao obtido no trabalho.

1.3 Objetivo deste trabalho

O método descrito neste trabalho tem como objetivo principal propor um novo método para a estimativa da pose da cabeça no espaço 3D, retornando os ângulos de rotação correspondentes nos eixos X, Y e Z. Neste método, primeiramente, obtemos os pontos característicos nas imagens de entrada com o algoritmo do modelo ativo de forma (ASM). Depois, localizam-se os pontos correspondentes nas instâncias do modelo 3D da face. Por fim, os pontos do modelo são ajustados aos da face usando operações geométricas como a rotação e escala.

1.4 Organização do documento

Neste capítulo foram apresentadas as considerações iniciais para o entendimento do trabalho de pesquisa desta dissertação. O conteúdo do trabalho está estruturado da seguinte forma:

- O Capítulo 2 mostra os diferentes métodos propostos pelos autores para estimar a pose da cabeça, também se explica a teoria usadas no trabalho.
- O Capítulo 3 explica o método proposto para estimar a pose da cabeça, o qual recebe como entrada a posição das características principais da face e um modelo 3D desta. Depois as transformações geométricas auxiliam na estimativa dos ângulos de rotação nos diferentes eixos (X, Y e Z).
- O Capítulo 4 mostra os resultados obtidos com o método proposto e mostra comparações com outros métodos apresentados na literatura.
- O Capítulo 5 apresenta as conclusões do trabalho e os trabalhos futuros que poderão ser desenvolvidos tendo como base este trabalho.

2 MARCO TEÓRICO

2.1 Revisão bibliográfica sobre a detecção da pose da cabeça

A maioria de autores trabalha com métodos híbridos (aqueles que combinam dois ou mais métodos) para a estimativa da pose da cabeça humana. Alguns destes métodos são descritos a seguir.

Chen et al. (CHEN et al., 1998) estima a pose da cabeça em imagens monoculares usando apenas informação da região da pele e da região do cabelo das pessoas. Primeiro, o autor propõe usar um algoritmo eficiente baseado na cor uniforme e usa a teoria *fuzzy* para extrair as regiões de pele e cabelo as quais são usadas para detectar a face. Depois, as áreas, os centros delas e os eixos da mínima inércia das duas regiões são calculados e usados para estimar a pose da cabeça no espaço 3D.

Shinn-Ying e Hui-Ling (SHINN; HUI, 1998) propõem um novo método o qual trabalha com a localização dos extremos dos olhos e da boca em imagens faciais. Em seguida, utilizam-se conhecimentos obtidos a priori tais como: os extremos dos olhos são coplanares; a linha horizontal formada pelos olhos e a formada pela boca são paralelas; o eixo da simetria facial é perpendicular à linha formada pelos olhos. Outros pontos usados neste método são marcados manualmente, como as distâncias existentes entre os cantos dos olhos e os da boca.

Fitzpatrick (FITZPATRICK, 2001) descreve um sistema automático que recupera os componentes rígidos da pose da cabeça. Este método segue os procedimentos convencionais de rastreamento de pose mudando configurações relativas para configurações referenciais. Para determinar a pose da cabeça é necessário começar o rastreamento dela na posição frontal e desenhar uma elipse sobre a face para depois calcular os movimentos laterais da cabeça e determinar a pose dela.

Cordea et al. (CORDEA et al., 2001) propõe o uso de um modelo elíptico da cabeça e o uso do filtro de Kalman para estimar e rastrear a pose da cabeça em tempo real em sequências de vídeo. Em seguida, Cordea et al. (CORDEA et al., 2002) apresenta um refinamento da proposta anterior de rastreamento e estimativa da pose da cabeça em movimento no espaço 3D, usando a geometria da câmera, a cena e o filtro de Kalman modificado.

Sujith e Kim (SRINIVASAN; BOYER, 2002) trabalham com auto-espacos os quais são necessários para estimar regularmente a pose da cabeça. Este método é robusto e computacionalmente não custoso. Para construir uma pose baseada em *eigenspaces* se usa PCA em um grupo de nove imagens de diferentes pessoas e com diferentes poses da cabeça. Um auto-espaco está definido pelos autovetores da matriz de covariância deste conjunto de treinamento.

Qiang Ji (JI, 2002) propõe uma nova abordagem para a estimativa e rastreamento em

três dimensões (3D) da pose de um rosto humano a partir das imagens obtidas de uma imagem monocular com projeção perspectiva. A detecção da face é dada explorando as propriedades fisiológicas dos olhos sob uma iluminação especial IR (*Infrared*) e algumas restrições geométricas. Primeiro detecta-se uma elipse no rosto da pessoa e esta é rastreada nos quadros subsequentes, a qual permite acompanhar a pose da face no espaço 3D de quadro em quadro.

Li e Fan (LIN; FAN, 2003) estimam a pose da face automaticamente. Eles desenvolveram um sistema baseado em regras o qual procura as potenciais regiões da face onde podem existir combinações das posições dos olhos e da boca que formem um triângulo isósceles. Depois é proposto um algoritmo que classifica a pose usando uma máscara de ponderação para poder obter os ângulos de rotação da cabeça no espaço 3D.

Pan et al. (PAN; ZHU; JI, 2005) trabalha com um novo método para a estimativa da pose da cabeça usando imagens monoculares e os conhecimentos das características geométricas baseadas na teoria de imagens *pinhole* e do modelo de rotação da cabeça no espaço 3D. A principal idéia do algoritmo é obter a informação 3D da cabeça em relação as projeções e a localização dos sete pontos faciais (ponto médio da boca, extremos dos olhos, etc.).

Dervinis (DERVINIS, 2006) descreve um método baseado em modelo para estimar a posição da cabeça em 3D. Primeiro, é gerado um modelo da cabeça, onde o centro de rotação 3D fica entre os olhos. Depois, usa-se um algoritmo para encontrar os pontos característicos do rosto, como olhos, nariz e boca. Por fim, aplica-se um algoritmo de detecção do ângulo de rotação da cabeça usando como base os pontos encontrados na face.

Em Zhang et al. (MA et al., 2006) calcula-se a pose da cabeça usando um LGBP (*Local Gabor Binary Patterns*) o qual trabalha com a forma e orientação da cabeça em múltiplas vistas. No LGBP, o filtro de Gabor pode trabalhar com oito diferentes orientações da cabeça e o LBP (*Local Binary Pattern*) reflete a informação das diferentes orientações e escalas. Para a classificação das características usa-se SVM, o qual é treinado com diferentes poses da cabeça.

Vatahska et al. (VATAHSKA; BENNEWITZ; BEHNKE, 2007) propõe um método que estime a pose da cabeça a partir de imagens monoculares. O algoritmo proposto tem três passos principais. Primeiro o detector de face classifica a cabeça em três grupos: pose frontal, direita e esquerda. Depois um classificador é treinado com AdaBoost (*Adaptive Boosting*) usando um *Hair-Like* de características, detectando assim as características faciais, tais como, a ponta do nariz, os olhos, a boca, etc. Por fim, a localização dessas características são usadas como entrada para uma rede neural, a qual estima os três ângulos de rotação usando um modelo da cabeça em 3D.

Tu et al. (TU et al., 2007) descrevem um método que estima a pose da cabeça usando a localização do nariz. Primeiramente a ponta do nariz é marcada num conjunto de faces (*groundtruth*). Depois a pose da cabeça é calculada usando um modelo tensor, o qual utiliza coeficientes obtidos no HOSVD (*High Order Singular Value Decomposition*).

Gourier et al. (GOURIER et al., 2007) propõem um método que calcula a pose da cabeça em imagens de baixa resolução. Primeiro as faces são detectadas usando características baseadas em cromaticidade, então, os níveis de cinza da face são normalizados, os quais servem as entradas para uma auto-associativa memória linear. Uma das memórias são calculadas para cada pose usando regras de aprendizagem *Widrow-Hoff*. Por fim, as poses são calculadas e classificadas usando um processo de *Winner-takes-all*.

Martins e Batista (MARTINS; BATISTA, 2008) combinam técnicas como AAM (*Active Appearance Models*) e POSIT (*Pose from Orthography and Scaling with Iterations*). O algoritmo considera a cabeça humana como um corpo rígido e trabalha com AdaBoost (*Adaptive Boosting*) para a detecção da face. Além disso, usa-se a técnica de AAM para obter a localização dos pontos característicos da face, tais como olhos, boca, sobrancelhas, etc. Esses parâmetros formam a entrada do algoritmo POSIT, o qual se encarrega de estimar a pose da cabeça no espaço 3D.

Yan et al. (YANG; ZHANG; ZHU, 2008) investigam a pose da cabeça como um problema que é considerado *front-end*. Neste método propõem-se um modelo computacional da cabeça que tenha as poses baseadas na fatoração de múltiplas vias não-negativas (NMWF - *Non-negative Multi-way Factorization*). Para encontrar a base de representação facial, o algoritmo NMWF é aplicado sobre os dados de formação de imagens faciais. O *tensor face* inclui três fatores dessas imagens, posturas e pessoas. Os dois primeiros fatores são usados para construir o modelo computacional para estimativa de pose. A medida discriminatória para descobrir a pose da cabeça é definida pela semelhança entre a base e o *tensor* representativo dos testes da imagem facial, que é a projeção dos rostos no subespaço gerado pela base *tensor faces*.

Cordea et al. (CORDEA; PETRIU; PETRIU, 2008) descrevem um modelo baseado em rastreamento em 3D, o qual recupera a posição, orientação e expressões faciais do movimento da cabeça. Este método usa um músculo antropométrico baseado no modelo ativo de forma em 3D, um algoritmo baseado na correspondência das características e o filtro de Kalman estendido para estimar a pose e a expressão.

Dornaika et al. (DORNAIKA; RADUCANU, 2009) propõem um algoritmo de rastreamento em tempo real no espaço 3D da cabeça e das ações faciais em seqüência de vídeo monocular onde a câmera é de baixa qualidade. Este método possui duas contribuições principais, a primeira propõe a inicialização da pose no espaço 3D o qual tem como base o detector de face em 2D e um sistema de autovetores. Já na segunda, utilizam-se métodos já propostos de inicialização e rastreamento para realçar a interação humano-computador de um robô AIBO (*Artificial Intelligence roBOt*). Esta aplicação está baseada na imitação da pose da cabeça em aplicações de realidade virtual, *video games*, etc.

Liu et al. (LIU; LU; LUO, 2009) apresentam um novo método para a estimar a pose da cabeça. Inicialmente as imagens são processadas para realçar as características faciais e eliminar as informações de identidade pelos métodos da cor da pele e a transformada Laplaciana da Gaussiana (LoG). Em seguida, as imagens pré-processadas são usadas para construir um subespaço de *eigenposes* com ajuda do método de PNMF (*Poses-based Non-negative Matrix Factorization*). Finalmente as imagens de prova são representadas pelas projeções do espaço de *eigenposes*, no qual facilmente pode-se estimar a pose da cabeça por métodos de classificação de padrões ou representações não-lineares.

Zabulis et al. (ZABULIS; SARMIS; ARGYROS, 2009) descrevem um método para a estimativa de pose da cabeça humana em ambientes multicâmera. O método combina o detector de face (FDn) com a informação de uma estrutura 3D da cabeça. Para isso, primeiro localiza-se a cabeça nas imagens adquiridas usando o FDn. Depois as múltiplas vistas da face e da cabeça são projetadas numa hipotética superfície esférica, a qual é concêntrica à cabeça da pessoa. Detectando o centro da face, pode-se estimar a orientação 3D da cabeça e calcular os ângulos *pitch*, *yaw* e *roll*.

Terissi e Gómez (TERISSI; GOMEZ, 2010) apresentam um novo algoritmo que se para a pose da cabeça e as expressões faciais. O algoritmo combina abordagens baseadas

em modelos e em características para a estimativa da pose. Finalmente um modelo 3D genérico da cabeça é adaptado às diferentes faces da pessoa.

Yu et al. (XINGUO et al., 2010) estimam a pose da cabeça usando imagens térmicas. Este método segmenta o corpo da pessoa, ou seja, separa os *pixels* do *background* do *foreground*. Em seguida localiza-se a cabeça humana usando um modelo de cabeça-ombro. Nesta parte, o autor usa o algoritmo de *convex-hull* para identificar a localização do pescoço e do queixo, para então projetar a localização da imagem da cabeça dentro de um *eigenspace* definido pelo conjunto de treinamento. Finalmente, encontra-se a pose da cabeça calculando a distância mínima euclidiana entre o vetor da face obtido e o do conjunto de treinamento.

Ma et al. (MA; WANG, 2010) propõem um método de representação esparsa para melhorar o funcionamento do método de estimativa da pose da cabeça. Para isso os autores tem como prioridade eliminar o *background* das imagens, usando o método de BSCR (*Block based Sparse Representation Classifier*), o qual tem como entrada a imagem da face detectada. Em seguida, ela é dividida em blocos para poder eliminar o *background*. Então, as faces das pessoas são classificadas pelo SRC (*Sparse Representation Classifier*) para calcular a pose da cabeça, onde o fator SCI (*Sparsity Concentration Index*) indica a acurácia do método.

Foytik et al. (FOYTIK et al., 2010) trabalham com um método que está orientado ao uso de PC (*Phase Congruency*) característico e de CCA (*Canonical Correlation Analysis*) para definir um subespaço de estimativa de pose. As características de congruência fornecem uma representação da pose sensível das imagens da face original, eliminando as variações de iluminação, bem como, as variações de identidade atenuantes. Além disso, as funcionalidades orientadas servem para sintetizar vários vetores da base canônica de correspondência.

Liu et al. (LIU; LU; LI, 2010) propõem o estudo da estimativa da pose da cabeça usando variações de aparência, levando em conta que os dados da pose estejam em múltiplos *manifolds*. Primeiro, eles constroem um conjunto de n-pessoas usando uma pose similar. Depois usa-se um método supervisionado para obter a baixa dimensionalidade do *manifold* embutido para cada conjunto de n-pessoas. Finalmente, com o método de agrupamento de *k-manifold* que eles propõem, integram-se os *manifolds* embutidos com os agrupados para fazer com que cada *manifold* aprenda uma única estrutura geométrica.

Foytik et al. (FOYTIK et al., 2010) apresentam um método de estimativa da pose da cabeça usando análises de correlação canônica, onde a variação da pose é calculada como uma continuação variável e representada por um *manifold* num espaço característico. A técnica identifica diretamente a dimensão fundamental que maximiza a correlação entre a imagem e a pose. Adicionalmente usa-se o filtro de Gabor para proporcionar uma entrada suscetível à pose.

Huang et al. (HUANG; DING; FANG, 2010) propõem a utilização de características de Gabor tendo como base florestas aleatórias. As florestas aleatórias são técnicas de classificação que conseguem lidar com o problema de classificação em multi-classes com precisão e rapidez. Eles trabalham com duas fontes de aleatoriedade, entradas aleatórias e características aleatórias. Além disso, eles programam um LDA (*Linear Discriminative Analysis*) como nó de prova para melhorar o poder discriminatório de cada árvore, e também aumentar a profundidade desta.

Krinidis et al. (KRINIDIS; NIKOLAIDIS; PITAS, 2007) apresentam uma nova abordagem para estimar a pose da cabeça em sequências únicas de vídeo. Eles inicializam o

algoritmo usando um detector de face. Depois, uma técnica de rastreamento que utiliza um modelo deformável de superfície é usada para rastrear a face na sequência de vídeo. A estimativa da pose da cabeça é calculada usando um vetor de características obtido nas equações que controlam a deformação do modelo usado para o rastreamento. O vetor calculado é usado para treinar o SVM e assim estimar a cabeça pose.

Valenti et al. (VALENTI; YUCEL; GEVERS, 2009) trabalham com um esquema híbrido no qual a matriz de transformação é obtida a partir da pose da cabeça e é usada para normalizar a região dos olhos. Por sua vez, a matriz de transformação gerada pela localização dos olhos é utilizada para corrigir o procedimento de estimativa da pose. O sistema é projetado para aumentar a precisão das estimativas da localização dos olhos em vídeos de baixa resolução e para estender a faixa de operação do localizador de olhos, melhorando assim a precisão e capacidade de reinicialização do rastreamento da pose.

Gurbuz et al. (GURBUZ; OZTOP; INOUE, 2011) apresentam o uso de visão estéreo para a estimativa da pose da cabeça (orientação e posição). O sistema trabalha obtendo um plano da face com os dados da reconstrução em 3D. A novidade essencial neste trabalho é a utilização desse plano e a localização dos olhos no espaço 3D. Esta localização é usada para restringir o processo de busca de disparidade da reconstrução facial 3D e orientar a seleção dos *voxels* da face para a estimativa do plano da face. Depois o plano da face e a localização dos olhos são usados para calcular a pose da cabeça.

Yari et al. (YARI; CAVALCANTI; SCHARCANSKI, 2010) (<http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6148835>) apresentam um método para estimar a pose da cabeça em imagens monoculares usando um modelo 3D da face. Dada uma imagem facial, localizam-se os pontos característicos que logo são comparados com os pontos encontrados nas instâncias do modelo 3D da face projetado no espaço 2D da imagem. Finalmente, a pose é estimada minimizando o erro de comparação entre as características faciais na imagem e as determinadas na instância do modelo 3D.

Dahmane et al. (DAHMANE; LARABI; DJERABA, 2010) propõem um novo método para a estimativa da pose da cabeça usando a simetria existente na face da pessoa. Este método precisa de uma boa localização das características faciais e da inicialização frontal, já que nesta posição a simetria existente entre as duas partes da face (esquerda e direita) podem ser observadas. Neste primeiro passo calculam-se três pares de características, olhos, nariz e boca. Depois são calculadas as medidas (largura e comprimento) de cada um dos pares e associadas a cada categoria de movimento (*yaw* e *pitch*). Para calcular a pose da cabeça é usado um classificador que está treinado com as medidas dos movimentos das faces.

Yari e Scharcanski (YARI; SCHARCANSKI, 2011) (<http://dx.doi.org/10.1117/12.884138>) propõem um método simplificado para estimar a pose da cabeça em imagens monoculares de forma rápida. A pose da cabeça é calculada comparando a posição dos pontos localizados na face, que correspondem aos pontos pré-selecionados em um modelo da face em 3D. Para localizar os pontos na imagem da face (tais como olhos, boca e nariz), usamos o modelo ativo de forma (ASM). Estes pontos são detectados no espaço 2D e, em seguida, o modelo 3D da face é ajustado por transformações geométricas aos pontos encontrados na imagem. Se chamou *fast* posto que algoritmo proposto é muito mais rápido já que não precisamos a criação de instâncias para fazer a comparação de pontos, senão se usa a distância de mínimos quadrados para ajustar a modelo em 3D à face. Para obter a pose da face de uma imagem o processador demora aproximadamente 30 segundos por imagem.

A continuação se apresenta a Tabela 2.1 de classificação dos métodos propostos pelos autores classificados pelo uso de tipo de imagens e de método usado segundo a classificação de Murphy-Chutorian (MURPHY; MANUBHAI, 2009):

Tabela 2.1: Classificação de métodos segundo Murphy-Chutorian

Métodos	Autor	Tipo de Imagem
Vetor de detectores	Zhang et al. (MA et al., 2006)	Monocular
	Yu et al. (XINGUO et al., 2010)	Monocular
	Yan et al. (YANG; ZHANG; ZHU, 2008)	Frames video
Baseado em <i>manifolds</i>	Liu et al. (LIU; LU; LI, 2010)	Monocular
	Foytik et al. (FOYTIK et al., 2010)	Monocular
	Sujith e Kim (SRINIVASAN; BOYER, 2002)	Monocular
Flexíveis	Martins e Batista (MARTINS; BATISTA, 2008)	Monocular
	Zabulis et al. (ZABULIS; SARMIS; ARGYROS, 2009)	Frame de vídeo
	Gurbuz et al. (GURBUZ; OZTOP; INOUE, 2011)	Estéreo
Geométricos	Shinn-Ying e Hui-Ling (SHINN; HUI, 1998)	Monocular
	Pan et al. (PAN; ZHU; JI, 2005)	Monocular
Rastreamento	Fitzpatrick (FITZPATRICK, 2001)	Frame de vídeo
	Cordea et al. (CORDEA et al., 2001)	Frame de vídeo
	Cordea et al. (CORDEA; PETRIU; PETRIU, 2008)	Frame de vídeo
	Dornaika et al. (DORNAIKA; RADUCANU, 2009)	Frame de vídeo
	Krinidis et al. (KRINIDIS; NIKOLAIDIS; PITAS, 2007)	Frame de vídeo
Híbridos	Quian Ji (JI, 2002)	Monocular
	Lin e Fan (LIN; FAN, 2003)	Monocular
	Dervinis (DERVINIS, 2006)	Estereo
	Vatahska et al. (VATAHSKA; BENNEWITZ; BEHNKE, 2007)	Monocular
	Gourier et al. (GOURIER et al., 2007)	Monocular
	Tu et al. (TU et al., 2007)	Monocular
	Huang et al. (HUANG; DING; FANG, 2010)	Monocular
	Teressi e Gómez (TERISSI; GOMEZ, 2010)	Monocular
	Valenti et al. (VALENTI; YUCEL; GEVERS, 2009)	Frame de vídeo
	Yari et al. (YARI; CAVALCANTI; SCHARCANSKI, 2010)	Monocular
Dahmane et al. (DAHMANE; LARABI; DJERABA, 2010)	Frame de vídeo	
Yari et al. (YARI; SCHARCANSKI, 2011)	Monocular	

Este trabalho propor um novo método o qual estima a pose da cabeça usando imagens monoculares, já que na literatura existem muitos métodos os quais fazem o procedimento usando maiormente frames de vídeo para poder fazer rastreamento da cabeça, e em outros casos se usam imagens estéreo, todo este tipo de informação geralmente se obtém em ambientes controlados e com uso de equipamento provavelmente custoso.

2.2 Modelo ativo de forma

O modelo ativo de forma (ASM) é um modelo estatístico que obtém a forma final de um objeto iterativamente tendo como base um conjunto de treinamento. Este modelo foi desenvolvido por Tim Cootes e Chris Taylor, em 1995 (COOTES et al., 1995).

No ASM, a forma do objeto está representada por um conjunto de pontos características (*landmark points*) e as formas do modelo são limitadas pelo modelo de distribuição de

pontos (PDM) e pelas variações dos níveis de cinza dos pontos característicos (LGL). A metodologia original sugere o uso da distância de Mahalanobis entre cada um dos pontos da face e da forma média para detectar a melhor posição para cada ponto de referência (COOTES et al., 1995). O ASM usa um conjunto de treinamento com imagens de mesma forma com a finalidade que o modelo possa variar e se consiga obter a forma final do objeto usando outra imagem que não se encontra no conjunto de treinamento.

A técnica tem sido amplamente utilizada para fazer análises de imagens de rostos, conjuntos mecânicos ou imagens médicas (em 2D e 3D). Ela está intimamente relacionada com o modelo de aparência ativa, por isso, o ASM é também conhecido como o Método de "*Smart Snakes*" (COOTES et al., 1995), já que tem muita similaridade com o modelo de contornos ativos (ou Snakes) de Kass et al. (KASS; WITKIN; TERZOPOULOS, 1988) com a diferença de que o ASM usa um conjunto de treinamento.

As principais componentes de um ASM são:

- Pontos do modelo de distribuição (PDM) usados para desenhar a forma do modelo do objeto, sendo também são chamados pontos característicos.
- Um conjunto de níveis de cinza (LGL) usados para obter as variações dos níveis de cinza para cada um dos pontos característicos.

2.2.1 Modelos de distribuição de pontos

O modelo de distribuição de pontos (PDM) é usado para descrever as variações das formas da face que foram observadas durante o treinamento. O PDM está baseado na variação da posição de cada ponto característico que pertençam ao conjunto de treinamento. Ou seja, cada um dos pontos característicos tem uma específica distribuição no espaço da imagem.

Com a finalidade de obter os PDM corretos, é necessário determinar os pontos característicos a serem usados para alinhar as formas do conjunto de treinamento e obter uma forma média que represente o objeto. Em nosso caso, que represente a forma média da face.

A seguir descrevem-se os passos para encontrar os pontos de distribuição do modelo de um objeto:

Rotular o conjunto de treinamento : Precisamos determinar o número de pontos característicos que sejam adequados para representar a forma do objeto, sendo este procedimento realizado manualmente. Isto é, para cada imagem do conjunto de treinamento (1, 2, ..., N) localiza-se a forma e identifica-se cada um dos pontos característicos mais significantes do objeto. É importante que estes pontos característicos tenham uma boa localização e uma correspondência exata entre cada uma das diferentes imagens do conjunto de treinamento.

Deve-se ter as correspondências exatas no momento de marcar os pontos característicos nas imagens. A localização destes pontos deveria ser o mais exata possível, dado que estas localizações controlam o resultado da variação dos pontos numa nova imagem.

Segundo Bookstein (BOOKSTEIN, 1989, 1997), os pontos característicos que são usados nas imagens biológicas e médicas e servem para definir as formas específicas do objeto. Estas podem ser de três tipos:

1. Pontos característicos dependentes da aplicação, tais como, o centro dos olhos num modelo de rosto ou nos cantos de uma borda (Figura 2.1).

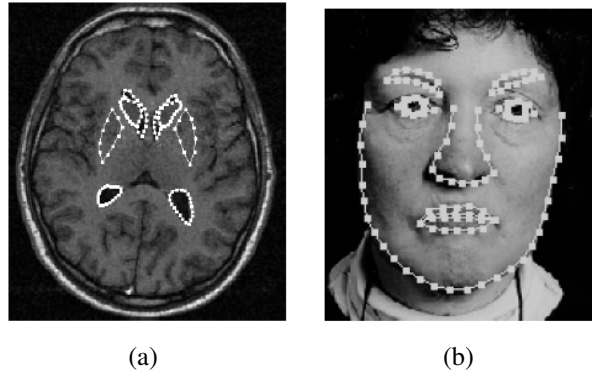


Figura 2.1: Exemplo de pontos principais em imagens médicas (a-b). Fonte (BOOKSTEIN, 1989)

2. Pontos característicos independentes da aplicação, tais como o ponto central de um objeto que indica a orientação ou curvatura;
3. Pontos característicos que misturam os dois tipos mencionados anteriormente, os quais podem ser interpolados dos pontos do tipo 1 e 2, como por exemplo pontos marcados a uma distância igual numa forma.

Por exemplo, na Figura 2.2 os pontos 0, 3, 5, 10 são os pontos que identificam as características do modelo e são de tipo 1, e os outros são pontos espaçados igualmente no contorno do modelo sendo do tipo 3.

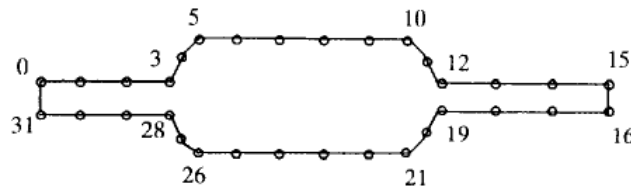


Figura 2.2: Modelo de 32 pontos de um resistor. Fonte (BOOKSTEIN, 1989)

Como resultado temos, os n pontos marcados nas N imagens do conjunto de treinamento, onde o ponto (x_{ij}, y_{ij}) pertence à j -ésima coordenada da forma da i -ésima imagem,

$$\mathbf{x}_i = (x_{i0}, y_{i0}, x_{i1}, y_{i1}, \dots, x_{ik}, y_{ik}, \dots, x_{in-1}, y_{in-1})^T, \quad (2.1)$$

onde o tamanho de \mathbf{x}_i , é $2n$ e $1 \leq i \leq N$.

Alinhamento das formas: A fim de estudar as variações das posições das formas do conjunto de treinamento (isto é, cada um dos vetores do conjunto de treinamento está representado por \mathbf{x}_i) primeiro precisamos alinhar cada uma delas.

O alinhamento é feito pela mudança de pose da forma do conjunto de treinamento. São utilizadas as transformações geométricas como a rotação, translação e escala, as quais devem ter como prioridade a minimização dos pesos das distâncias ao quadrado entre os pontos característicos equivalentes das diferentes formas.

Para introduzir o problema de alinhamento das formas, começaremos alinhando apenas duas formas \mathbf{x}_1 e \mathbf{x}_2 , as quais estão representadas por:

$$\begin{aligned}\mathbf{x}_1 &= (x_{10}, y_{10}, x_{11}, y_{11}, \dots, x_{1(n-1)}, y_{1(n-1)})^T, \\ \mathbf{x}_2 &= (x_{20}, y_{20}, x_{21}, y_{21}, \dots, x_{2(n-1)}, y_{2(n-1)})^T,\end{aligned}$$

onde cada um dos vetores contém coordenadas (x, y) de cada ponto característico. Depois precisamos encontrar uma rotação θ , um fator de escala s e uma translação \mathbf{t} que mapeiem \mathbf{x}_1 em $M(s, \theta)[\mathbf{x}_2] + \mathbf{t}$ e minimizem o somatório dos pesos:

$$E = (\mathbf{x}_1 - M(s, \theta)[\mathbf{x}_2] - \mathbf{t})^T \mathbf{W} (\mathbf{x}_1 - M(s, \theta)[\mathbf{x}_2] - \mathbf{t}), \quad (2.2)$$

onde a rotação, escala e translação de uma só coordenada (x_{2k}, y_{2k}) são dadas por:

$$M(s, \theta) \begin{bmatrix} x_{2k} \\ y_{2k} \end{bmatrix} = \begin{pmatrix} (s \cos \theta)x_{2k} - (s \sin \theta)y_{2k} \\ (s \sin \theta)x_{2k} + (s \cos \theta)y_{2k} \end{pmatrix}, \quad (2.3)$$

$$\mathbf{t} = (t_x, t_y, \dots, t_x, t_y)^T, \quad (2.4)$$

e \mathbf{W} é uma matriz diagonal de pesos de cada um dos pontos característicos da forma:

$$\mathbf{W} = \text{diag}(w_{0x}, w_{0y}, w_{1x}, w_{1y}, \dots, w_{(n-1)x}, w_{(n-1)y}). \quad (2.5)$$

A matriz \mathbf{W} ajuda a dar maior significância entre os pontos característicos da forma que tendem a ser mais estáveis. Para calcular estes pesos, $(w_0, w_1, \dots, w_{n-1})$, devemos obter a distância entre os pontos a e b que pertencem à \mathbf{x}_i (R_{iab}). Posteriormente deve-se calcular a variância entre a distância dos pontos a e b denotada por $\mathbf{V}_{R_{ab}}$. Finalmente, o peso do a -ésimo ponto característico é dado por:

$$\mathbf{W}_a = \left(\sum_{i=0}^{n-1} \mathbf{V}_{R_{ab}} \right)^{-1}, \quad (2.6)$$

onde $0 \leq a \leq (n-1)$ e n é o número de pontos característicos.

O Algoritmo 1 descreve o alinhamento de todo o conjunto de treinamento (N imagens):

Algoritmo 1 Algoritmo para alinhar N formas

Entrada : Conjunto de N formas.

Saída: N formas alinhadas.

Para ($i=2$ to N) **Faça**

Alinhar cada forma \mathbf{x}_i com \mathbf{x}_1 .

Fim

Repita

Calcular a forma média das formas alinhadas,

Alinhar a forma média com \mathbf{x}_1 ,

Realinhar todas as formas com a média.

Até o processo convergir.

No Algoritmo 1, alinhar significa encontrar uma rotação, translação e escala que sejam aplicadas a um par de formas que sejam o mais semelhantes possíveis. Para isso, é preciso

minimizar o peso dos quadrados das distâncias entre os pontos equivalentes das diferentes formas (COOTES et al., 1995).

Obtendo as estatísticas: As i -ésimas formas alinhadas do conjunto de treinamento estão representadas por \mathbf{x}_i , onde \mathbf{x}_i contém as novas coordenadas das formas já alinhadas.

Cada um dos \mathbf{x}_i 's (onde $i = 1, 2, \dots, N$) são de dimensão $2n$, então eles podem ser representado por um ponto no espaço $2n$ -dimensional. Os N vetores representam as N formas alinhadas as quais estão numa nuvem de N pontos no espaço $2n$ -dimensional. Esta região se chama *Allowable Shape Domain* (ASD), onde cada ponto desta região tem uma forma semelhante às outras formas que pertencem à mesma região.

Depois, devemos calcular a similaridade existente entre os pontos característicos de cada forma usando a equação de distância Euclidiana, onde, se a distância entre os pontos é menor, a similaridade entre as formas é maior. A distância Euclidiana d entre os pontos que representam as forma \mathbf{x}_1 , e \mathbf{x}_2 é denotada por:

$$\begin{aligned} d_{12} &= \sqrt{(\mathbf{x}_1 - \mathbf{x}_2)^T \mathbf{W} (\mathbf{x}_1 - \mathbf{x}_2)}, \\ d_{12}^2 &= w_0(x_{10} - x_{20})^2 + w_0(y_{10} - y_{20})^2 + w_1(x_{11} - x_{21})^2 + w_1(y_{11} - y_{21})^2 + \\ &\quad \dots + w_{n-1}(x_{1(n-1)} - x_{2(n-1)})^2 + w_{n-1}(y_{1(n-1)} - y_{2(n-1)})^2, \end{aligned} \quad (2.7)$$

onde \mathbf{W} é a matriz diagonal de pesos obtida anteriormente (Equação 2.5).

Em seguida, encontramos o eixo principal que controla o comportamento das variações dos N pontos que se encontram no espaço $2n$ -dimensional. Aplicamos a análise dos principais componentes (PCA) para gerar um novo conjunto de dados chamado componentes principais. Estes componentes principais são ortogonais entre si, onde cada eixo informa um modo diferente de variação, em que os pontos tendem a se movimentarem juntos segundo as formas.

Em muitas aplicações só se trabalha com os primeiros componentes principais, já que eles descrevem a maior porcentagem da total variação do conjunto original de dados. Desta forma, a dimensão do modelo é reduzida e as variações são descritas por um número menor de variáveis.

Depois de obter os principais componentes, podemos expressar cada um dos \mathbf{x}_i como uma combinação linear de estes, para logo poder expressar o valor médio $\bar{\mathbf{x}}$ como:

$$\bar{\mathbf{x}} = \frac{1}{N} \sum_{i=1}^N \mathbf{x}_i. \quad (2.8)$$

e a diferença $d\mathbf{x}_i$ entre o vetor médio e cada um dos \mathbf{x}_i :

$$d\mathbf{x}_i = \mathbf{x}_i - \bar{\mathbf{x}}. \quad (2.9)$$

A matriz de covariância para os pontos característicos da forma é dada como:

$$\mathbf{C}_x = \frac{1}{N} \sum_{i=1}^N (\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T, \quad (2.10)$$

$$\mathbf{C}_x = \frac{1}{N} \sum_{i=1}^N (d\mathbf{x}_i)(d\mathbf{x}_i)^T. \quad (2.11)$$

Representamos a diferença $d\mathbf{x}_i$ como uma combinação linear dos componentes principais:

$$d\mathbf{x}_i = b_{i0}\mathbf{p}_0 + b_{i1}\mathbf{p}_1 + \dots + b_{i2n-1}\mathbf{p}_{2n-1}, \quad (2.12)$$

onde \mathbf{p}_l é o l -ésimo componente principal do eixo e b_{il} é o peso escalar de \mathbf{p}_l , então a Equação 2.12 é escrita como:

$$d\mathbf{x}_i = \mathbf{P}\mathbf{b}_i, \quad (2.13)$$

onde:

$$\mathbf{P} = (\mathbf{p}_0, \mathbf{p}_1, \dots, \mathbf{p}_{t-1}), \quad (2.14)$$

$$\mathbf{b} = (b_0, b_1, \dots, b_{t-1})^T. \quad (2.15)$$

E a forma final obtida é dada por:

$$\mathbf{x}_i = \bar{\mathbf{x}} + \mathbf{P}\mathbf{b}_i. \quad (2.16)$$

A partir da equação anterior é possível obter que:

$$\mathbf{b}_i = \mathbf{P}^{-1}(\mathbf{x}_i - \bar{\mathbf{x}}). \quad (2.17)$$

Como \mathbf{P} é uma matriz quadrada com colunas ortogonais, podemos afirmar que $\mathbf{P}^{-1} = \mathbf{P}^T$, já que $\mathbf{P}^T\mathbf{P} = \mathbf{P}\mathbf{P}^T = \mathbf{I}$. Então temos que:

$$\mathbf{b}_i = \mathbf{P}^T(\mathbf{x}_i - \bar{\mathbf{x}}). \quad (2.18)$$

2.2.2 Modelo da aparência dos níveis de cinza

Para cada ponto da forma a aparência dos níveis de cinza descreve a aparência típica da vizinhança que está localizada no contorno do ponto. A principal idéia é examinar as regiões de níveis de cinza para cada um dos pontos característicos encontrados nas imagens do conjunto de treinamento. Isto é, para todos os j 's que pertençam as imagens i , onde $i = 0, 1, \dots, N-1$ e $j = 1, 2, \dots, n$. A informação dos níveis de cinza é usada para estimar o melhor movimento dos pontos característicos numa nova imagem.

Existem muitas formas para fazer isto, sendo que a forma mais básica e efetiva foi descrita por Cootes (COOTES et al., 1995) usando contornos em $1D$ (também chamados *whiskers*). Para poder modelar a vizinhança de cada um dos pontos característicos da forma usando só uma dimensão ($1D$) é preciso extrair o contorno de cada um deles.

Logo, para cada um dos pontos característicos j na imagem i do conjunto de treinamento extraímos os níveis de cinza g_{ij} de tamanho n_p que formam o contorno. Este tem como centro o ponto característico j (COOTES et al., 1993):

$$\mathbf{g}_{ij} = (g_{ij0}, g_{ij1}, \dots, g_{ijn_p-1})^T. \quad (2.19)$$

Depois calculamos a derivada de g_{ij} a qual é de tamanho $n_p - 1$ dada por :

$$d\mathbf{g}_{ij} = (g_{ij1} - g_{ij0}, g_{ij2} - g_{ij1}, \dots, g_{ijn_p-1} - g_{ijn_p-2}). \quad (2.20)$$

Para cada um dos pontos característicos j na imagem i , devemos normalizar suas derivadas:

$$\mathbf{y}_{ij} = \frac{d\mathbf{g}_{ij}}{\sum_{k=0}^{n_p-2} |d\mathbf{g}_{ijk}|}. \quad (2.21)$$

Também é necessário obter a média das derivadas normalizadas para cada ponto característico do conjunto de treinamento, onde para cada j temos:

$$\bar{\mathbf{y}}_j = \frac{1}{N} \sum_{i=1}^N d\mathbf{g}_{ij}. \quad (2.22)$$

A matriz de covariância das derivadas é dada por:

$$\mathbf{C}_{yj} = \frac{1}{N} \sum_{i=1}^N (\mathbf{y}_{ij} - \bar{\mathbf{y}}_j)(\mathbf{y}_{ij} - \bar{\mathbf{y}}_j)^T. \quad (2.23)$$

Com isso, obtemos um modelo de nível de cinza em torno de qualquer ponto característico j representado por $\bar{\mathbf{y}}_j$ e a covariância deste \mathbf{C}_{yj} , através do conjunto de treinamento.

O Algoritmo 2 é um resumo da obtenção de $\bar{\mathbf{y}}_j$ e \mathbf{C}_{yj} para cada ponto característico j .

Algoritmo 2 Algoritmo para encontrar o nível de cinza em cada ponto característico.

Entrada : Conjunto de N formas .

Saída: $\bar{\mathbf{y}}_j$, \mathbf{C}_{yj} .

Para ($i=1$ to N) **Faça**

Para ($j=1$ to n) **Faça**

Obter g_{ij} que tem um tamanho n_p :

$$\mathbf{g}_{ij} = (g_{ij0}, g_{ij1}, \dots, g_{ijn_p-1})^T;$$

Calcular a derivada de cada \mathbf{g}_{ij} :

$$d\mathbf{g}_{ij} = (g_{ij1} - g_{ij0}, g_{ij2} - g_{ij1}, \dots, g_{ijn_p-2} - g_{ijn_p-1});$$

Normalizar a derivada:

$$\mathbf{y}_{ij} = \frac{d\mathbf{g}_{ij}}{\sum_{k=0}^{n_p-2} |d\mathbf{g}_{ijk}|}.$$

Fim

Fim

Para ($j=1$ to n) **Faça**

Calcular a média de \mathbf{y}_{ij} :

$$\bar{\mathbf{y}}_j = \frac{1}{N} \sum_{i=1}^N d\mathbf{g}_{ij};$$

Calcular a matriz de covariância das derivadas normalizadas:

$$\mathbf{C}_{yj} = \frac{1}{N} \sum_{i=1}^N (\mathbf{y}_{ij} - \bar{\mathbf{y}}_j)(\mathbf{y}_{ij} - \bar{\mathbf{y}}_j)^T.$$

Fim

2.2.3 Localização dos pontos característicos numa nova imagem

Para localizar os pontos característicos numa nova imagem devemos considerar os PDM e LGL obtidos com o conjunto de treinamento. Isto é, usamos os parâmetros já obtidos anteriormente como: $\bar{\mathbf{x}}$, \mathbf{P} , $\bar{\mathbf{y}}_j$ and \mathbf{C}_{yj} .

Primeiro, localizamos a face nas imagens monoculares. Para isto, nós usamos o algoritmo de Viola-Jones (VIOLA; JONES, 2001)(Anexo A). Este algoritmo usa como

entrada uma imagem que contém uma face e retorna a localização desta na imagem (Figura 2.3).



Figura 2.3: Identificação da face humana usando o algoritmo de Viola-Jones, (VIOLA; JONES, 2001).

Obtendo a localização da face na imagem, localizamos também a forma média do conjunto de treinamento ($\bar{\mathbf{x}}$) obtida pela Equação 2.8, onde $\bar{\mathbf{x}}$ contém os pontos característicos usados no modelo.

Em seguida, calculamos os níveis de cinza dos pontos característicos na nova imagem de tamanho n_s , onde $n_s < n_p$:

$$\mathbf{f}_j = (f_{j0}, f_{j1}, \dots, f_{jn_s-1})^T. \quad (2.24)$$

Calculamos a derivada de \mathbf{f}_j de tamanho $n_s - 1$:

$$d\mathbf{f}_j = (f_{j1} - f_{j0}, f_{j2} - f_{j1}, \dots, f_{jn_s-2} - f_{jn_s-1}). \quad (2.25)$$

Normalizando a derivada dos pontos:

$$\mathbf{fy}_j = \frac{d\mathbf{f}_j}{\sum_{k=0}^{n_s-2} |d\mathbf{f}_{jk}|}. \quad (2.26)$$

Tendo todos estes dados calculamos a distância mínima que possa existir entre cada um dos \mathbf{fy}_j da imagem e a média obtida no conjunto de treinamento $\bar{\mathbf{y}}_j$. Para isso usamos a distância de Mahalanobis(MAHALANOBIS, 1936):

$$d = \sqrt{(\mathbf{fy}_j - \bar{\mathbf{y}}_j)^T \mathbf{C}_{y_j}^{-1} (\mathbf{fy}_j - \bar{\mathbf{y}}_j)}, \quad (2.27)$$

onde \mathbf{C}_{y_j} é a matriz de covariância obtida na Equação 2.23.

Calculamos o menor movimento que é feito pelo conjunto de pontos característicos na posição inicial e determinamos o $\mathbf{b} = \mathbf{P}^T (\mathbf{x}\mathbf{f} - \bar{\mathbf{x}})$. Para que o valor \mathbf{b} seja o correto usamos os autovalores calculados no conjunto de treinamento de tal forma que:

$$-3\sqrt{\lambda_i} < b_i < 3\sqrt{\lambda_i}. \quad (2.28)$$

onde b_i é o primeiro elemento de \mathbf{b} , e λ_i é o i -ésimo componente principal, se o limiar aumentar o diminuir a quantidade de amostras a serem usadas variariam o modelo médio

obtido, além disso muitas amostras que possuíam ruído (amostras com os pontos não marcados corretamente) formariam parte do conjunto de treinamento.

Atualizamos os valores de \mathbf{b} e obtemos o modelo final:

$$\mathbf{xf} = \bar{\mathbf{x}} + \mathbf{Pb}. \quad (2.29)$$

Fazemos todo este procedimento até que a distância entre os valores encontrados no conjunto de treinamento e a possível posição dos pontos característicos na imagem seja mínima. O Algoritmo 3 resume como a localização dos pontos característicos numa nova imagem é obtida.

Algoritmo 3 Algoritmo para encontrar a posição dos pontos característicos numa nova imagem.

Entrada : $\mathbf{P}, \bar{\mathbf{x}}, \bar{\mathbf{y}}_j, \mathbf{C}_{yj}$.

Saída: Localização dos pontos característicos numa nova imagem.

Localizar a face na imagem de entrada usando o Algoritmo de Viola-Jones;
 Localizar a forma média $\bar{\mathbf{x}}$, no centro da imagem facial;
 Inicializar $\mathbf{b} = 0$.

Repita

Para ($j=1$ to n) **Faça**

Calcular os níveis de cinza na nova imagem,

$$\mathbf{f}_j = (f_{j0}, f_{j1}, \dots, f_{jn_s-1})^T;$$

Calcular as derivadas de \mathbf{f}_j ,

$$d\mathbf{f}_j = (f_{j1} - f_{j0}, f_{j2} - f_{j1}, \dots, f_{jn_s-2} - f_{jn_s-1});$$

Normalizar as derivadas: $\mathbf{fy}_j = \frac{d\mathbf{f}_j}{\sum_{k=0}^{n_s-2} |d\mathbf{f}_{jk}|}$;

Minimizar a distância entre $d\mathbf{f}_j$ e \mathbf{y}_j usando a distância de Mahalanobis.

Fim

Obter o vetor de movimentos e calcular o menor deles para deslocar os pontos;

Movimentar os pontos até a posição correta;

Determinar $\mathbf{b} = \mathbf{P}^T (\mathbf{xf} - \bar{\mathbf{x}})$;

Limitar \mathbf{b} baseada nos autovalores obtidos no conjunto de treinamento e atualizar b ;

Estimar $\mathbf{xf} = \bar{\mathbf{x}} + \mathbf{Pb}$.

Até distância seja mínima.

2.3 Conclusões

Na revisão bibliográfica observaram-se diferentes métodos para obtenção da pose da cabeça através de imagens monoculares, estéreo ou *frames* de vídeo. No caso de trabalhar com *frames* de vídeo, o método aplicado é a localização de pontos característicos. Em seguida, é feito o rastreamento em imagens diferentes. Para manipulação de imagens estéreo pode-se calcular a profundidade da posição dos pontos característicos (localização de olhos, nariz, boca), no entanto, é necessário ter parâmetros da câmera.

Em imagens monoculares usam-se pontos principais da face e modelos de cabeça em 3D que forneçam a informação necessária para estimar a pose da cabeça, além de precisar de métodos treinados e rotulados.

O modelo ativo de forma (ASM) é um modelo deformável que usa um conjunto de treinamento para obter uma nova forma do objeto. Além disso, o ASM é composto por um modelo de distribuição de pontos e de um modelo de níveis de cinza de cada ponto, onde juntos ajudam a obter uma nova forma.

O modelo de distribuição de pontos (PDM) descreve a forma do objeto. Os pontos característicos devem ser rotulados em cada uma das imagens do conjunto de treinamento. Em seguida, estas formas são alinhadas, sendo depois calculada a similaridade que existe entre cada uma das formas de tal maneira que a distância seja menor entre os pontos característicos das formas. Finalmente, com ajuda dos análises de componentes principais, calcula-se a forma média de todo o conjunto de treinamento.

Os níveis de cinza são calculados para obter a intensidade esperada ao redor de cada ponto característico na imagem. Para calculá-los trabalha-se com um vetor de tons de cinza de tamanho n_p que tem como centro o ponto característico. Depois eles são derivados e normalizados para que finalmente seja obtida a média de cada ponto característico \bar{y} e a matriz de covariância dele.

3 MÉTODO PROPOSTO

3.1 Considerações iniciais

Neste capítulo explicaremos detalhadamente o método desenvolvido para a estimativa da pose da cabeça no espaço 3D usando imagens monoculares. O método calcula os ângulos no espaço 3D, *pitch*, *yaw* e *roll*, como se mostram na Figura 3.1 (estes estão relacionados aos movimentos da cabeça na Figura 3.2) os quais são calculados seguindo os passos a serem descritos posteriormente.

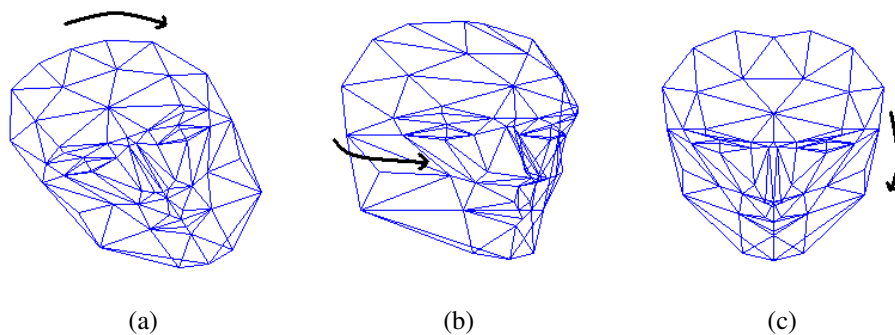


Figura 3.1: Movimentos do modelo 3D da face: (a)roll, (b)yaw e (c)pitch.

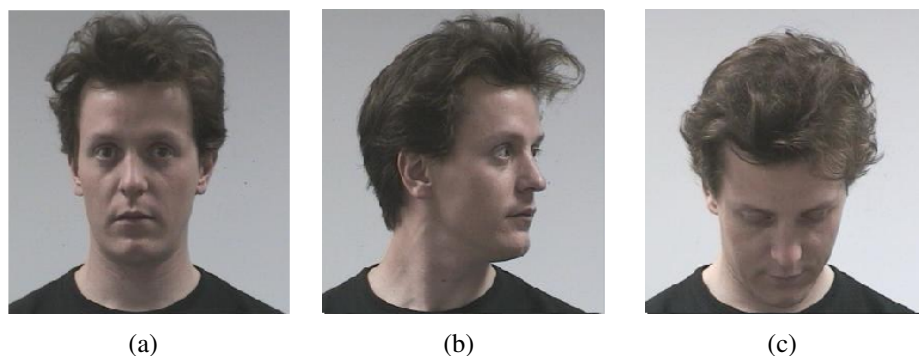


Figura 3.2: Movimentos da cabeça humana que podem ser mostrados em imagens monoculares.

A abordagem proposta é dividida nos seguintes passos: (1) detecção das características faciais em imagens da face em 2D usando o algoritmo de modelo ativo de forma, e a extração dos pontos principais no conjunto do modelo de instâncias em 3D, os quais fazem correspondência com os pontos extraídos na face com o ASM; (2) estimativa dos

ângulos α , β e γ usando as informações obtidas no passo anterior, e (3) refinamento do modelo calculando a distância mínima que é obtida através do conjunto de instâncias e os pontos da face.

3.2 Detecção de características faciais

Para a detecção das características faciais numa imagem monocular usamos o modelo ativo de forma (ASM), o qual foi explicado no Capítulo 2. Este modelo recebe como entrada uma imagem monocular. Para inicializar o modelo precisamos usar o algoritmo de Viola-Jones (VIOLA; JONES, 2001). Como foi mencionado anteriormente, este algoritmo recebe como parâmetro uma imagem que contém uma face nos dando como resultado a posição desta na imagem. Com esta posição podemos inicializar o modelo ativo de forma.

O modelo ativo de forma foi treinado com uma base de imagens diferente á ser usada para testar o método. Esta base de imagens contem a cabeça em diferentes posições. Este modelo treinado nos proporciona a forma média do objeto (neste caso a forma média das faces que foram usadas no conjunto de treinamento) para que logo possamos inicializar o processo de busca dos pontos principais na nova imagem.

Este processo começa posicionando a forma média na imagem tendo como centro o centro da janela que localiza a face. Depois disto, este modelo começa a se ajustar à nova imagem apresentada até conseguir a melhor localização dos pontos na imagem. Estes pontos são armazenados em Xf :

$$Xf = \{(xf_1, yf_1), (xf_2, yf_2), \dots, (xf_n, yf_n)\}, \quad (3.1)$$

onde $n = 15$. O que significa que a quantidade de pontos a ser usada para fazer a comparação entre o modelo em 3D e as características faciais é 15. Estes pontos são mostrados na Figura 3.3.



Figura 3.3: Pontos usados para estimar a pose da cabeça, (GOURIER; HALL; CROWLEY, 2004).

Por outro lado, precisamos criar o conjunto de instâncias com o modelo 3D de face, que representará todos os prováveis movimentos da cabeça humana. O modelo usado neste trabalho tem como base o modelo CANDIDE (RYDFALK, 1987), que foi criado por Mikael Rydfalk no Grupo de *Image Coding* de Linköping em 1987 e contém 75 vértices e 100 triângulos os quais cobrem toda a face humana. Uma pequena modificação

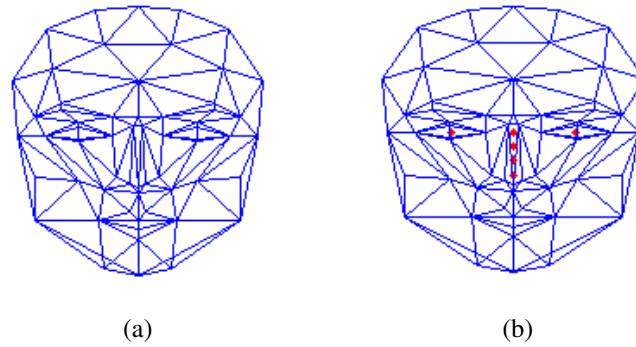


Figura 3.4: Modelo 3D da face: (a) Modelo CANDIDE; (b) Modelo da face modificado .

foi feita a este modelo, para um melhor ajuste dele nas imagens de face, acrescentando alguns pontos nos olhos e no nariz (Figura 3.4).

Os pontos da máscara estão denotados por:

$$mask = (m_1, m_2, \dots, m_q), \quad (3.2)$$

onde $m_i = (mx_i, my_i, mz_i)$ e $q = 81$, enquanto que $mask$ denota todo o conjunto de pontos contidos no modelo 3D da face.

Além de criar o conjunto de instâncias precisamos também obter um subconjunto de pontos principais chamado Xm , onde Xm contém 15 pontos representados no espaço 3D:

$$Xm = \{(xm_1, ym_1, zm_1), (xm_2, ym_2, zm_2), \dots, (xm_{15}, ym_{15}, zm_{15})\}. \quad (3.3)$$

O processo de normalização é feito do seguinte jeito, primeiro se normalizam os pontos da máscara (Xm) e os pontos da face (Xf), isto se realiza calculando os pontos máximos de cada eixo para logo dividir entre o máximo dele, assim nos temos os pontos num intervalo de $[-1, 1]$. Trasladamos o ponto do meio da boca para a origem de coordenadas do sistema e atualizamos os pontos encontrados em Xm e Xf .

Esta normalização é realizada para que o conjunto de pontos da máscara e da face tenham as mesmas dimensões com o mesmo centro de origem, facilitando de esta forma o melhor ajuste dos pontos à máscara. Até este momento, os pontos Xf e da máscara se encontram em coordenadas do sistema.

Para a criação de nossos candidatos do modelo de instâncias $\{MI\}$ usaremos $mask$ e a transformação geométricas de rotação. Os modelos de instâncias em 3D estão dados por :

$$mask_{R(\gamma, \alpha, \beta)} = mask \cdot R(\gamma, \alpha, \beta), \quad (3.4)$$

onde $mask_{R(\gamma, \alpha, \beta)}$ é a máscara rotacionada segundo a combinação dos ângulos α , β e γ e cada um dos ângulos tem uma variação de $[-90^\circ, -89^\circ, \dots, -1^\circ, 0^\circ, 1^\circ, \dots, 89^\circ, 90^\circ]$ criando no total $m=181^3$ instâncias do modelo 3D, as quais estão armazenadas em $\{MI\}$ (estes movimentos são realizados por causa que no a fazer a pesquisa sobre o tema, e revisando a base de imagens disponíveis para o teste, ditas imagens tinham movimentos desde -90 até 90 . Esta informação fez que desde o começo do trabalho os teste e implementação estejam baseados com esta premissa. Logo, segundo Sforza et al. (SFORZA et al., 2002) um adulto tem uma restrição de movimento, o qual é a média a qual pode

variar segundo cada pessoa. Além disso, na base de dados (Poing04) os ângulos de movimentos vão desde -90 até 90 graus nos eixos X e Y, e por causa de isso que no momento de fazer os testes eles foram realizados tendo essa premissa):

$$\{MI\} = \{mask_{R(\gamma_1, \alpha_1, \beta_1)}, \dots, mask_{R(\gamma_{\vartheta}, \alpha_{\vartheta}, \beta_{\vartheta})}, \dots, mask_{R(\gamma_{181^3}, \alpha_{181^3}, \beta_{181^3})}\}, \quad (3.5)$$

onde $\vartheta = 1, \dots, 181^3$. Mas nem todas as instâncias do modelo 3D armazenadas em $\{MI\}$ serão usadas para fazer a correspondência com os pontos que se encontram em Xf .

3.3 Cálculo dos ângulos de movimento

Para estimar os ângulos da pose da cabeça, usaremos como entrada os pontos encontrados na face com o ASM, e o conjunto de instâncias $\{MI\}$. Primeiramente estimaremos o ângulo γ que ajudará a reduzir a complexidade de nossos dados e assim poderemos pré-selecionar um subconjunto de instâncias que se encontram no conjunto $\{MI\}$. Em seguida, estimaremos os ângulos α e β , os quais são calculados através do cálculo das distâncias entre os pontos da face e do modelo de instâncias.

3.3.1 Estimativa do ângulo γ

O primeiro cálculo a ser realizado é a estimativa do ângulo no eixo Z (γ), já que só precisamos apenas da informação em 2D. Esta informação é obtida depois do cálculo das características faciais da imagem. Além disso nós necessitamos reduzir nosso espaço de busca para evitar fazer $181 \times 181 \times 181$ comparação, por isso no momento de calcular o ângulo γ este ajudará a diminuir as comparações.

Neste caso, para o cálculo do ângulo precisamos os pontos que se encontram nos olhos (Er e El). Para obtê-los seguimos os passos:

- Sejam $er1$, $er2$, $el1$, $el2$ os pontos extremos dos olhos. Calculamos o ponto médio para cada um deles:

$$\begin{aligned} Er_x &= \frac{er2_x - er1_x}{2}, \\ Er_y &= \frac{er2_y - er1_y}{2}. \end{aligned} \quad (3.6)$$

onde Er_x e Er_y são as coordenadas x e y da posição do ponto médio do olho direito.

- Com a Equação 3.6 calculamos o ponto central do olho esquerdo (El).
- Obtemos o ângulo γ_A com a seguinte equação:

$$\gamma_A = \tan^{-1} \frac{(Er_y - El_y)}{(Er_x - El_x)}. \quad (3.7)$$

A Figura 3.5 mostra graficamente o cálculo do ângulo γ em imagens faciais.

A estimação deste ângulo nos ajuda a reduzir a quantidade de instâncias do modelo da face em 3D o qual se encontra armazenado em $\{MI\}$, para que logo só use-se um subconjunto das instâncias (quer dizer, que os ângulos a serem usados estão dados



Figura 3.5: Estimativa do ângulo γ_A .

pelas combinações do γ_A com α e β). Estas serão comparadas nos passos posteriores do método.

Posto que a estimativa deste ângulo pode produzir algum erro, assumimos uma tolerância de $\pm 2^\circ$. Os valores do ângulo γ são armazenados em γ , o qual está definido por $[\gamma_A] \pm 2^\circ$:

$$\gamma \in [\gamma_A - 2, \gamma_A - 1, \gamma_A, \gamma_A + 1, \gamma_A + 2]. \quad (3.8)$$

O Algoritmo 4 resume todos os passos para a obtenção de γ :

Algoritmo 4 Algoritmo para encontrar a rotação no eixo Z.

Entrada : Xf .

Saída: Ângulos γ .

Calcular o ponto médio dos olhos El e Er ;

Calcular o ângulo γ_A formado pelos pontos El e El :

$$\gamma_A = \tan^{-1} \frac{(Er_y - El_y)}{(Er_x - El_x)};$$

Assumir um intervalo de erro de ± 2 ; $\gamma: [\gamma_A - 2, \gamma_A - 1, \gamma_A, \gamma_A + 1, \gamma_A + 2]$.

3.3.2 Estimativa dos ângulos α e β

Depois de ter selecionado os valores do ângulo γ_A , construímos um subconjunto de instâncias de $\{MI\}$ chamado $\{MI_1\}$, que inclui apenas os k modelos da face em $3D$ com $\gamma \in [\gamma_A - 2, \gamma_A - 1, \gamma_A, \gamma_A + 1, \gamma_A + 2]$ e com ângulos α e β , onde $\alpha, \beta \in [-90^\circ, -89^\circ, \dots, 0, \dots, 89^\circ, 90^\circ]$ (em nosso experimentos $k = 5 \times 181 \times 181$).

Para cada uma das instâncias k do modelo de faces em $3D$ armazenados em $\{MI_1\}$ selecionamos os n pontos principais que serão relacionados aos n pontos contidos em Xf . Os n pontos selecionados do modelo de face em $3D$ são representados por Xm^i :

$$Xm^i = \{(xm_1^i, ym_1^i, zm_1^i), (xm_2^i, ym_2^i, zm_2^i), \dots, (xm_{15}^i, ym_{15}^i, zm_{15}^i)\}, \quad (3.9)$$

onde $i = 1, \dots, k$ e com $n = 15$ (como foi mencionado anteriormente). Como as características faciais em Xf estão em $2D$, precisamos projetar paralelamente Xm^i no plano XY (guardando as coordenadas x e y) para obter, desta forma, os pontos de Xm^i em $2D$ chamado pm^i :

$$pm^i = \{(xm_{p1}^i, ym_{p1}^i), (xm_{p2}^i, ym_{p2}^i), \dots, (xm_{pn}^i, ym_{pn}^i)\}. \quad (3.10)$$

sendo $n = 15$ em nossos experimentos.

Depois, para cada k modelos de face em 3D armazenados em $\{MI_1\}$, selecionamos as melhores instâncias que mimimizem a distância seguindo a equação de Distância Euclidiana:

$$dist(pm^i, Xf) = \sum_{t=1}^n \sqrt{(xm_{pt}^i - xf_t)^2 + (ym_{pt}^i - yf_t)^2}. \quad (3.11)$$

En nossos experimentos obtemos as w melhores instâncias para cada um dos ângulos $\gamma \in [\gamma_A - 2, \gamma_A - 1, \gamma_A, \gamma_A + 1, \gamma_A + 2]$ selecionando um total de r candidatos das intâncias ($r = 5 \times w$, $k > r$, e $w = 40$). Depois eliminamos os $(k - r)$ modelos de intâncias em 3D do conjunto $\{MI_1\}$ reduzindo nosso espaço de busca.

Resumindo, em nosso conjunto $\{MI_1\}$ so estarão armazenadas as melhores instâncias que possam representar o pose da cabeça de nossa imagem de entrada. Para cada uma das instâncias armazenadas em $\{MI_1\}$, se têm associados o respectivos ângulos $(\alpha_j, \beta_j$ e $\gamma_j)$, $j = 1, \dots, r$.

Para resumir os passos anteriores o Algoritmo 5 é apresentado a seguir.

Algoritmo 5 Algoritmo que calcula a distância entre os ponto da face e da máscara.

Entrada : $Xf, \gamma_A, \{MI\}$.

Saída: MI_1 , que contêm as r melhores instâncias do modelo 3D da face.

Armazenar MI_1 todas as combinações existentes de γ_A com os ângulos $\alpha, \beta \in [-90^\circ, -89^\circ, \dots, 0, \dots, 89^\circ, 90^\circ]$;

Atualizar $Xm^i, i = 1, \dots, k$;

Obter as projeções paralelas de Xm^i armazenadas em pm^i ;

Para (i=1 to k) **Faça**

 Calcular a distância Euclidiana $dist(pm^i, Xf)$ com :

$$dist(pm^i, Xf) = \sum_{t=1}^n \sqrt{(xm_{pt}^i - xf_t)^2 + (ym_{pt}^i - yf_t)^2};$$

Fim

 Salvar em $\{MI\}$ só as r intâncias $Xm^j, j = 1, \dots, r$ que minimize $dist(pm^i, Xf)$.

3.4 Refinamento do modelo

Até este momento, os ângulos α, β e γ foram estimados como valores inteiros, mas na prática eles deveriam ser fracionários. Para o refinamento destes ângulos devemos levar em conta que eles se encontram dentro intervalo de $[-90^\circ, -89^\circ, \dots, 0, \dots, 89^\circ, 90^\circ]$, e que esses valores podem ser menores que 1° .

Cada um dos valores dos ângulos da j -ésima instância em $\{MI_1\}$ estão dados por α_j, β_j e γ_j . Logo para cada um dos ângulos encontrados em $\{MI\}$ calculamos uma pequena variação (Δ):

$$\begin{aligned}
\alpha'_j &= \alpha_j + \Delta\alpha_j, \\
\beta'_j &= \beta_j + \Delta\beta_j, \\
\gamma'_j &= \gamma_j + \Delta\gamma_j,
\end{aligned} \tag{3.12}$$

$j = 1, \dots, r$ e Δ é a variação produzida para que os ângulos α_j , β_j e γ_j podam se ajustar aos pontos característicos encontrados na face Xf . O valor desta variação é optida usando otimização quem calcula o valor mínimo da função, tenfo como valor inicial 0 Para este calculo usamos otimização o qual ajuda a resolver o problema estimando o mínimo da função, tendo como valor inicial 0° .

Em nossos experimentos, $\Delta\alpha_j = \Delta\beta_j = \Delta\gamma_j = 2^\circ$. Estes ângulos indicam as pequenas rotações que são executadas para ajustar o modelo de instância em $3D$ que se encontra em $\{MI_1\}$.

Logo para estimar α^{best} , β^{best} e γ^{best} se usa:

$$(\alpha^{best}, \beta^{best}, \gamma^{best}) = \underset{\alpha'_j, \beta'_j, \gamma'_{j,j}}{\operatorname{argmin}} \operatorname{dist}(pm^j(\alpha'_j, \beta'_j), Xf), \tag{3.13}$$

onde $pm^j(\alpha'_j, \beta'_j)$ expressam a projeção paralela de uma rotação $(\alpha'_j, \beta'_j, \gamma'_j)$ armazenada em Xm^j dos j -ésimos modelos de instâncias da face em $3D$ de $\{MI_1\}$. A instância rotacionada é escolhida por $(\alpha^{best}, \beta^{best}, \gamma^{best})$, que minimiza a distância $\operatorname{dist}(pm^j(\alpha'_j, \beta'_j), Xf)$.

Vale lembrar que a Equação 3.13 é só aplicada em r instâncias de $\{MI_1\}$ que minimizam $\operatorname{dist}(pm^j, Xf)$, já que desta forma reduzimos a complexidade de nosso algoritmo (tudo isso comparado com a complexidade que poderia ser gerada se aplicássemos à Equação 3.13 a nossas k instâncias do modelo da face em $3D$).

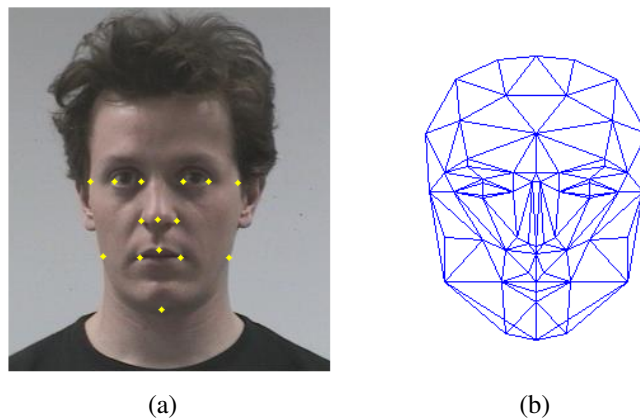


Figura 3.6: (a) Localização dos pontos na face, (b) máscara a ser usada para a estimativa da pose.

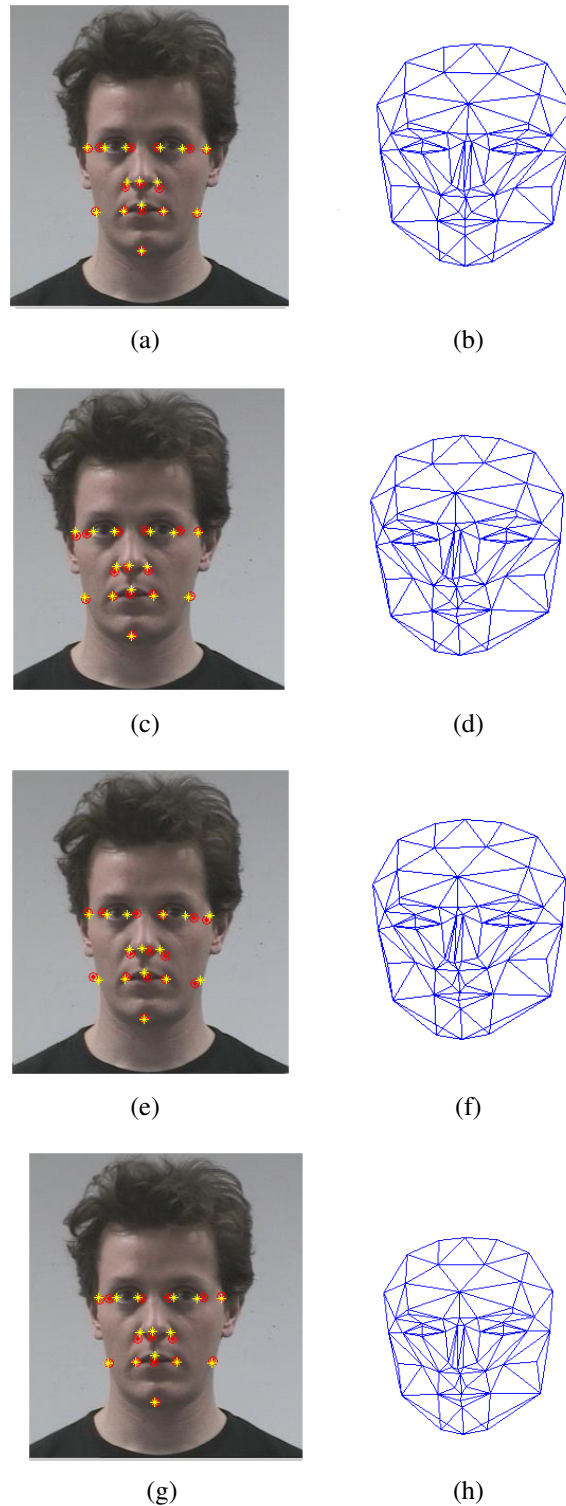


Figura 3.7: Estimativa da melhor pose da cabeça. As figuras (a), (c), (e), (g) mostram a face com os pontos característicos marcados em cor amarelo, e os pontos da máscara sobre ela marcados em cor vermelho. As figuras (b), (d), (f), (h) mostram as possíveis máscaras que minimizam as distâncias entre os pontos encontrados na face e os pontos projetados sobre ela; (b) R: $\alpha_A = -3^\circ$, $\beta_A = -6^\circ$, $\gamma_A = -1^\circ$ and R_N : $\alpha' = 0.0014^\circ$, $\beta' = 0.0038^\circ$, $\gamma' = 0.0309^\circ$. (d) R: $\alpha_A = -5^\circ$, $\beta_A = -10^\circ$, $\gamma_A = 1^\circ$ e R_N : $\alpha' = 0.0391^\circ$, $\beta' = 0.0443^\circ$, $\gamma' = 0.0021^\circ$. (f) R: $\alpha_A = -4^\circ$, $\beta_A = 4^\circ$, $\gamma_A = -3^\circ$ e R_N : $\alpha' = 0.0148^\circ$, $\beta' = 0.0930^\circ$, $\gamma' = 0.0522^\circ$. (h) R: $\alpha_A = 0^\circ$, $\beta_A = -9^\circ$, $\gamma_A = 1^\circ$ e R_N : $\alpha' = 0.0494^\circ$, $\beta' = 0.0276^\circ$, $\gamma' = 0.0040^\circ$.

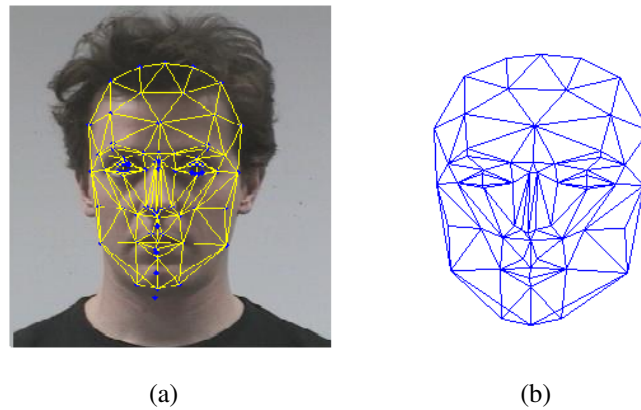


Figura 3.8: Ilustração da estimativa da pose da cabeça: (a) máscara final encontrada e ajustada à face (b) máscara com os ângulos $\alpha = -2.99^\circ$, $\beta = -5.99^\circ$ e $\gamma = -0.97^\circ$.

Os passos para obtenção da pose final da cabeça são ilustrados nas Figuras 3.6, 3.7 e 3.8. Primeiro obtemos as características faciais na imagem (Figura 3.6(a)) e no modelo 3D da face (Figura 3.6(b)). A Equação 3.13 permite o ajuste do modelo da face em 3D como se observa nas Figuras 3.7(a), 3.7(c), 3.7(e) e 3.7(g), onde temos as características faciais de Xf marcadas de cor amarelo e o ajuste dos correspondentes pontos característicos de $pm^j(\alpha_j + \Delta\alpha_j, \beta_j + \Delta\beta_j)$ que estão marcados de cor vermelho. A correspondência entre Xm^j e os modelos de instância da face em 3D ajustado para $(\alpha_j + \Delta\alpha_j, \beta_j + \Delta\beta_j, \gamma_j + \Delta\gamma_j)$ são mostradas nas Figuras. 3.7(b), (d), (f) e (h). Finalmente, a Figura 3.8(a) mostra a melhor instância do modelo da face em 3D que minimiza a Equação 3.13. O modelo da face em 3D é ilustrada em Figura 3.8(b), a qual esta relacionada com os ângulos α^{best} , β^{best} e γ^{best} .

3.5 Conclusões

A combinação dos métodos, método geométrico e o modelo ativo de forma, ajudam na estimativa da pose da cabeça. Podemos dizer que o método proposto é mais rápido posto que a quantidade de operações a serem realizadas para a estimativa da pose da cabeça são poucas e de custo menor, já que nós só precisamos de usar uma só imagem a comparação dos métodos que precisam a usar frames de vídeo ou imagens estéreo. Além de isso geralmente quando se usa vídeo ou imagens estéreo se usam os parâmetros de câmara dos quais nós não precisamos.

Em nosso caso usamos um modelo da face em 3D, a partir do qual criamos um conjunto de instâncias que serão projetadas no espaço da imagem. Em seguida, os pontos encontrados na imagem (pontos característicos) são relacionados aos encontrados no conjunto de instâncias. Nosso resultado é dado pela combinação dos ângulos α , β , γ que originam o modelo de instância que obtém a menor distância entre ambos conjuntos de pontos.

No Capítulo 4 apresentaremos os experimentos realizados com o método desenvolvido.

4 EXPERIMENTOS E RESULTADOS

4.1 Considerações iniciais

Neste Capítulo mostraremos os resultados obtidos pelo método proposto no Capítulo 3. Seguindo os passos apresentados primeiro obtemos o ângulo γ que corresponde ao eixo Z , fazendo o nosso raio de busca diminuir. Em seguida, ao obter a combinação destes ângulos (α , β e γ), refinamos cada um deles para que a distância entre pm^j e Xf seja mínima.

As imagens usadas neste trabalho foram obtidas a partir de uma base de imagens pública, Pointing04 (GOURIER; HALL; CROWLEY, 2004). Esta base contém imagens de quinze pessoas com diferentes poses da cabeça e com expressão facial neutra. A Figura 4.1, apresenta os movimentos da cabeça de uma pessoa que se encontra na base de imagens usadas para os testes do método desenvolvido. Nas imagens podem se observar que os movimentos da cabeça variam tanto no eixo X como no Y .

Cada uma das imagens de teste é de um tamanho de 384×288 , e sem variações de luminosidade nem modificação do tamanho dela (base de dados de Poing04). Para obter os pontos característicos o método de ASM trabalha bem com o tamanho da imagem, além disso devemos ter em conta que depois de ter ditos pontos nós normalizamos eles para evitar algum tipo de perda de informação.

Para o treinamento do ASM, as imagens não devem ter nenhum tipo de sobreposição ou perda de algum ponto característico da face, já que eles são necessários para obter os novos pontos nas imagens de teste do algoritmo. A base de imagens usada para o treinamento é a base de Huang et al. (GARY et al., 2007) (*LabeledFacesintheWild*, <http://vis-www.cs.umass.edu/lfw/#download>), da qual foram usadas 350 imagens as quais tem visibilidade de os pontos característicos a serem usados. A implementação do método de ASM tem como base a implementação de ASM para mãos desenvolvida por Dirk-Jan Kroon (<http://www.mathworks.com/matlabcentral/fileexchange/26706>), a qual foi modificada para a detecção de pontos característicos em imagens de face.

Para a inicialização do ajuste dos pontos à nova face se usou o Algoritmo de Viola-Jones o qual é uma implementação já treinada por Vahid Kazemi (<http://www.csc.kth.se/~vahidk/>). Este algoritmo possui a limitação de que se perder muita informação da face da pessoa esta não é localizada na imagem. Geralmente isto acontece quando os ângulos de rotação nos eixos X e Y são maiores ou iguais a $\pm 40^\circ$.

O ASM tem como base o Algoritmo de Viola-Jones posto que para ser inicializado é necessário a localização da cabeça na imagem, uma boa inicialização ajuda a encontrar mais rapidamente os pontos principais dentro da face na imagem



Figura 4.1: Imagens usadas para o teste do método

Neste trabalho apresentamos dois conjuntos de experimentos: (1) estimativa da pose da cabeça usando características faciais marcadas manualmente em cada uma das imagens das faces e (2) a estimativa da pose da cabeça esta baseada usando o ASM treinado para detectar as características faciais.

4.2 Primeiro experimento

Os primeiros testes realizados foram feitos para mostrar que o método de estimativa da pose da cabeça possui bom desempenho (YARI; CAVALCANTI; SCHARCANSKI, 2010). Para realizar este teste, primeiro é necessário localizar cada ponto característico na imagem da face, este procedimento foi realizado manualmente e por uma só pessoa e sem perder algum tipo de informação, quer dizer, encontrando os pontos necessários para poder ajustar o modelo 3D à face da imagem. Se fez as marcações de 15 pessoas com diferentes poses da cabeça.

Depois de obter para cada uma das pessoas a localização dos pontos característicos, a continuação é aplicado o método propostos para o calculo da estimativa de pose da cabeça.

As Figuras 4.2, 4.3, 4.4 mostram visualmente os resultados obtidos usando inicialização manual, onde as Figuras 4.2a, 4.3a, 4.4a mostram a máscara ajustada à imagem da face e as Figuras 4.2b, 4.3b, 4.4b mostram as máscaras obtidas depois de encontrar os ângulos de rotação nos diferentes eixos (X,Y e Z).

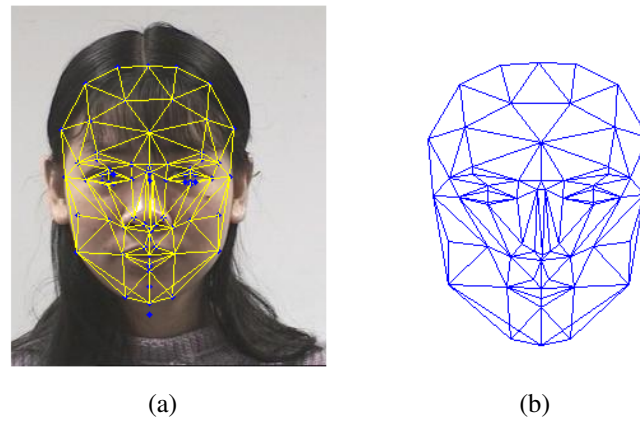


Figura 4.2: Ilustração do resultado da estimativa da pose da cabeça: (a) $mask_f$ ajustada à imagem da face. (b) $mask_f$ com ângulos: $\alpha = 3.88^\circ$, $\beta = 1.89^\circ$, $\gamma = 0.02^\circ$.

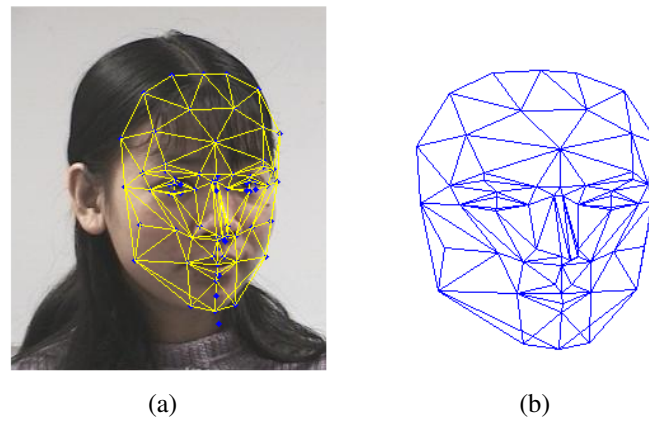


Figura 4.3: Ilustração do resultado da estimativa da pose da cabeça: (a) $mask_f$ ajustada à imagem da face. (b) $mask_f$ com ângulos: $\alpha = 2.84^\circ$, $\beta = -16.07^\circ$, $\gamma = 0.03^\circ$.

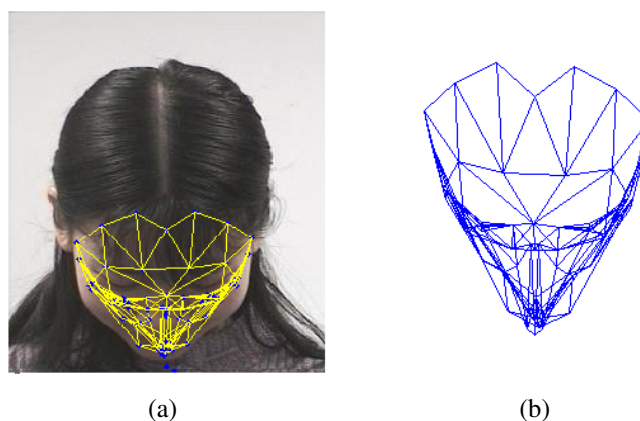


Figura 4.4: Ilustração do resultado da estimativa da pose da cabeça: (a) $mask_f$ ajustada à imagem da face. (b) $mask_f$ com ângulos: $\alpha = -73.34^\circ$, $\beta = 1.99^\circ$, $\gamma = 1.04^\circ$.

Para o teste das imagens de perfil (Figura 4.4), os pontos foram marcados manualmente para que de esta forma não se perca nenhum tipo de informação (isto significa que se localizaram todos os pontos característicos), posto que para o uso do ASM é necessário primeiro usar o Algoritmo de Viola-Jones o qual provocaria um erro no momento de lo-

calizar a face na imagem.

4.3 Segundo experimento

Nosso segundo experimento foi obtido usando uma inicialização automática dos pontos característicos. Para isso usamos o algoritmo de ASM, explicado anteriormente neste trabalho, o qual recebe como entrada a imagem da face e rotorna a localização dos pontos característicos. Em seguida, é usado o método de estimativa da pose da cabeça para obter os respectivos ângulos de movimento (*pitch*, *yaw* e *roll*).

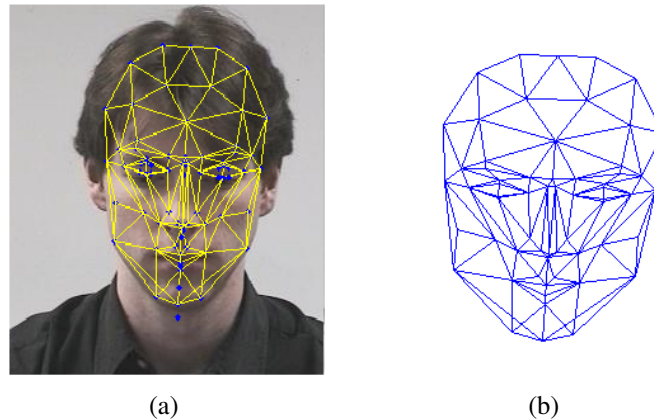


Figura 4.5: Ilustração do resultado da estimativa da pose da cabeça:(a) $mask_f$ ajustada à face. (b) $mask_f$ com os ângulos: $\alpha = -10.71^\circ$, $\beta = 1.04^\circ$, $\gamma = -2.96^\circ$.

As Figuras 4.5, 4.6, 4.7, 4.8 mostram visualmente os resultados obtidos usando o algoritmo de ASM para a localização dos pontos característicos nas imagens de face. As Figuras 4.5a, 4.6a, 4.7a, 4.8a mostram a máscara obtida e ajustada à face. Já as Figuras 4.5b, 4.6b, 4.7b, 4.8b mostram as máscaras finais calculadas depois de encontrar os respectivos ângulos de rotação da cabeça.

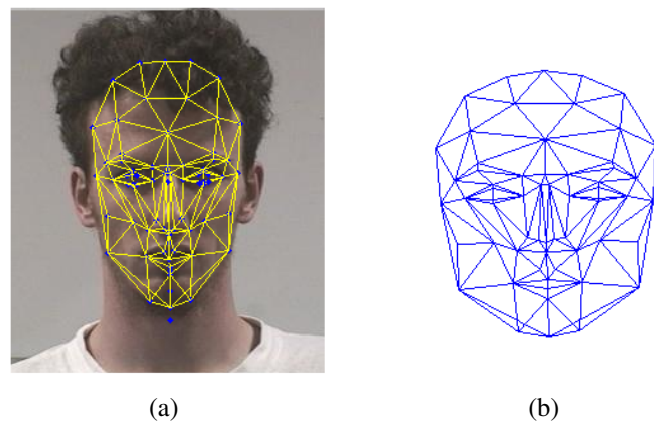


Figura 4.6: Ilustração do resultado da estimativa da pose da cabeça:(a) $mask_f$ ajustada à face. (b) $mask_f$ com os ângulos: $\alpha = 7.98^\circ$, $\beta = 1.03^\circ$, $\gamma = 0.97^\circ$.

onde imagens frontais significa que a pessoa que se encontra na imagem está observando para frente e com ângulos $(0, 0, 0)$ no espaço tridimensional, as imagens frontais são aquelas que possuem uma variação de ângulo no eixo X ou Y ou em ambos eixos.

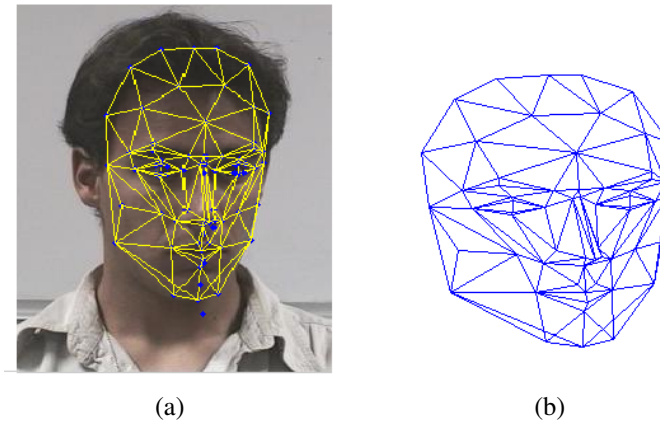


Figura 4.7: Ilustração do resultado da estimativa da pose da cabeça:(a) $mask_f$ ajustada à face. (b) $mask_f$ com os ângulos: $\alpha = 4.84^\circ$, $\beta = -20.69^\circ$, $\gamma = 1.84^\circ$.

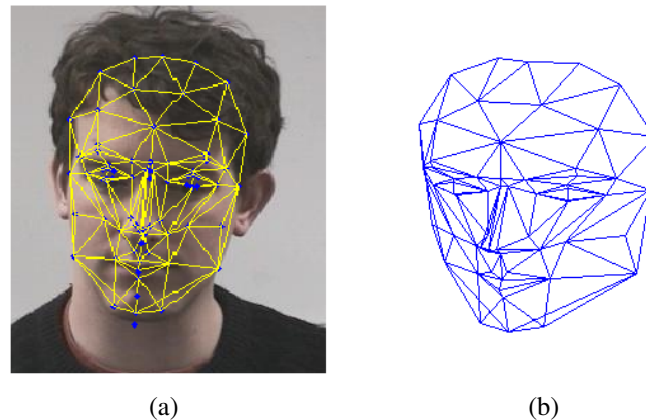


Figura 4.8: Ilustração do resultado da estimativa da pose da cabeça:(a) $mask_f$ ajustada à face. (b) $mask_f$ com os ângulos: $\alpha = -13.03^\circ$, $\beta = 24.97^\circ$, $\gamma = 1.18^\circ$.

Na Tabela 4.1 podemos observar que os erros produzidos com a inicialização automática são maiores que os com a inicialização manual, posto que estes além de ter o erro da estimação também possuem o erro produzido com o modelo ativo de forma.

O método proposto tem duas partes para serem analisadas. O primeiro é aquela que tem a inicialização manual. Neste ponto a localização dos pontos encontrados nas imagens da face posto que estes dependem da pessoa que faz a marcação deles. Esta localização é importante já que é feita por uma pessoa a qual consegue observar todos os pontos necessários para inicializar o método proposto sem ter algum tipo de restrição. Este é o motivo principal porque os resultados apresentados na Tabela 4.1 são melhores com esta inicialização.

A Tabela 4.1 a continuação mostra os erros médios obtidos no cálculo da pose da cabeça usando inicialização manual e automática:

O segundo é com uma inicialização automática, esta depende de dois métodos, o primeiro é o Algoritmo de Viola-Jones, o qual só detecta a face da pessoa quando esta não perde as características principais, isto significa que o algoritmo de Viola-Jones limita nossos ângulos da pose da cabeça. Os ângulos com que trabalha o algoritmo de Viola-Jones se encontram no intervalo de $[-40 \ 40]$.

O modelo ativo de forma encontra os pontos na face da pessoa segundo as imagens

Tabela 4.1: Resultados do erro produzido nos ângulos *pitch*, *yaw* e *roll* usando inicialização manual e automática.

	Pitch	Yaw	Roll
Imagens frontais usando inicialização automática	7.58°	7.09°	2.09°
Imagens de perfil usando inicialização automática	9.13°	8.07°	2.29°
Imagens frontais com inicialização manual	3.67°	3.83°	0.96°
Imagens de perfil com inicialização manual	4.59°	3.90°	1.57°

Tabela 4.2: Resultados do erro produzido nos ângulos *pitch*, *yaw* e *roll* usando *cross – validation*.

	Pitch	Yaw	Roll
Imagens frontais	7.24°	6.82°	1.81°
Imagens de perfil	8.75°	8.13°	2.15°

que foram usadas no treinamento, mas como foi explicado anteriormente este depende do algoritmo de Viola-Jones para a inicialização, em caso que este algoritmo não encontre a face na imagem a inicialização é realizada manualmente.

As imagens mostradas como exemplo nos experimentos realizados se observa que o método desenvolvido funciona corretamente, por exemplo, no caso da Figura 4.4 se fosse avaliada com o método de inicialização automática não teria o mesmo resultado posto que existe perda de informação já que não é possível localizar todos os pontos com o modelo ativo de forma além disso o Algoritmo de Viola-Jones não detectaria a face.

A Tabela 4.2 mostras os erros médios produzidos usando a tecnica de *cross – validation*, onde algumas imagens da base de imagens foram usadas para o treinamento do ASM, e as outras usadas para a estimativa da pose da cabeça:

Os erros maximos produzidos aumentam a medida que o angulo aumenta, no caso que o ângulo da imagem no eixo X é 60° (grand true), temos um erro maximo de 18°, isto é devido a que primeiro o algoritmo de Viola-Jones so trabalha no intervalo de $[-40\ 40]$ e então a inicialização do método de ASM é inicializado manualmente e este não consegue localizar os pontos nas posições corretas.

O desvio padrão de cada um dos ângulos avaliados no eixos X,Y e Z, se encontram na Tabela 4.3:

Usando outra base de imagens (www.cs.bu.edu/fac) para testar o método automático os erros médios obtidos segundos os eixos são apresentados na Tabela 4.4:

Para avalidar o método proposto, outros métodos foram pesquisados. Eles faziam a estimativa da pose da cabeça usando imagens monoculares e tambem usavam a mesma base de imagens usada neste trabalho. Os métodos encontrados foram do Tu et al. (TU et al., 2007), Shinn-Ying and Hui-Ling (SHINN; HUI, 1998) e de Foytik et al. (FOYTIK et al., 2010). Os resultados dos erros médios obtidos pelos autores são encontrados na Tabela 4.5.

O erro médio apresentado na Tabela 4.5 corresponde aos ângulos *pitch* e *yaw* (ângulos de movimento nos eixos X e Y), os quais se encontram nos artigos de ditos autores, a estimação do ângulo *roll* (ângulo de movimento no eixo Z) não é apresentado por eles.

Tabela 4.3: Resultados do desvio padrão produzido nos eixos X, Y e Z.

Eixo	Ângulo	Desvio Padrão	
		Inicialização Automática	Inicialização Manual
X	60°	8.43°	7.33°
	30°	5.09°	1.65°
	15°	4.29°	3.63°
	0°	3.67°	1.71°
	-15°	5.08°	4.35°
	-30°	5.58°	5.27°
	-60°	7.73°	7.09°
	Y	60°	9.09°
Y	45°	7.74°	7.28°
	30°	6.32°	5.54°
	15°	4.21°	3.24°
	0°	3.18°	3.02°
	-15°	7.48°	4.02°
	-30°	6.46°	3.41°
	-45°	8.37°	5.82°
	-60°	8.25°	7.38°
Z	0°	2.89°	1.98°

Tabela 4.4: Resultados do erro produzido nos ângulos *pitch*, *yaw* e *roll*.

	Pitch	Yaw	Roll
Imagens	7.34°	7.83°	3.08°

Tabela 4.5: Resultados dos erros obtidos por outros métodos.

	Pitch	Yaw
Método de Tu et al. com localização manual do nariz (TU et al., 2007)	5.63°	5.70°
Método de Tu et al. com localização automática do nariz (TU et al., 2007)	17.44°	15.88°
Método de Shinn et al. (SHINN; HUI, 1998)	25.42°	20.36°
Método de Foytik et al. (FOYTIK et al., 2010)	--	24.91°

O método apresentado neste trabalho foi implementado em MATLAB 7.9, num processador Core 2 Duo. Os tempos de processamento variam segundo aos experimentos efetuados:

- No caso da inicialização manual, o tempo de processamento é de 60 segundos, neste caso cada imagem já tem os pontos localizados na face e o método tem a finalidade de encontrar o modelo 3D da face que se ajusta à imagem.
- Quando a inicialização é feita automaticamente com o modelo ativo de formas este processo é de 2 minutos aproximadamente, posto que se precisa encontrar os pontos principais na face da imagem e logo se deve ajustar o modelo da face em 3D à face que se encontra na imagem.

5 CONCLUSÕES E TRABALHOS FUTUROS

Este trabalho propõe um novo método para resolver o problema da estimativa da pose da cabeça em imagens monoculares (imagens no espaço 2D) usando um modelo de face no espaço tridimensional.

Dada uma imagem monocular que contém uma face da pessoa humana, obtemos as características faciais dela, como a posição dos olhos, nariz, boca, utilizando o algoritmo de modelo ativo de formas. Este método estatístico usa um conjunto de treinamento para obter a posição das características numa imagem. A localização das características no espaço 2D são usadas como referência para a comparação com as respectivas características nas múltiplas instâncias do modelo da face em 3D projetadas no espaço 2D da imagem. Finalmente, a pose da cabeça é estimada minimizando a comparação do erro obtido entre a localização das características da imagem e do modelo de instâncias.

Para avaliar o desempenho do método proposto, fizemos dois experimentos que estimam a pose da cabeça. O primeiro foi feito marcando manualmente as características faciais. Já no segundo foi usando o algoritmo de ASM. Ambos testes usaram a mesma base de imagens e a mesma quantidade de pessoas (quinze pessoas diferentes e com expressão facial neutra).

Os resultados apresentados por este método são promissores, já que ele estima erros pequenos em relação aos outros métodos propostos para imagens monoculares que usam a mesma base de imagens.

O trabalho apresentado pode ser usado como base para outros, no caso de trabalhar com estimativa da pose da cabeça em tempo real, ou fazer rastreamento das características faciais. Em alguns casos, será necessário fazer avaliação extensiva de outros métodos para obter características faciais, por exemplo, métodos que trabalhem com modelo ativo de aparência, redes neurais, etc.

Além disso, pode-se investigar sobre outros métodos para a detecção da cabeça em imagens faciais, as quais podem ter algum tipo de sobreposição (perda de informação das características faciais).

REFERÊNCIAS

BAKER, S. et al. Real-time non-rigid driver head tracking for driver mental state estimation. In: WORLD CONGRESS ON INTELLIGENT TRANSPORTATION SYSTEMS, 11., Pittsburgh, PA. **Anais...** [S.l.: s.n.], 2004. n.CMU-RI-TR-04-10.

BLANZ, V. et al. A statistical method for robust 3D surface reconstruction from sparse data. In: INT. SYMP. ON 3D DATA PROCESSING, VISUALIZATION AND TRANSMISSION. **Anais...** [S.l.: s.n.], 2004. p.293–300. (3DPVT '04).

BOOKSTEIN, F. Principal warps: thin-plate splines and the decomposition of deformations. **Pattern Analysis and Machine Intelligence, IEEE Transactions on**, Washington, DC, USA, v.11, n.6, p.567–585, June 1989.

BOOKSTEIN, F. L. **Morphometric tools for landmark data : geometry and biology.** [S.l.]: Cambridge University Press, 1997.

CHEN, Q. et al. 3D head pose estimation without feature tracking. **Automatic Face and Gesture Recognition, IEEE International Conference on**, Washington, DC, USA, v.0, p.88 – 93, 1998.

COOTES, T. et al. Active shape models-their training and application. **Computer Vision and Image Understanding**, [S.l.], v.61, n.1, p.38 – 59, January 1995.

COOTES, T. F. et al. The use of active shape models for locating structures in medical images. In: INTERNATIONAL CONFERENCE ON INFORMATION PROCESSING IN MEDICAL IMAGING, 13., London, UK. **Proceedings...** Springer-Verlag, 1993. p.33–47. (IPMI '93).

CORDEA, M. D. et al. Real-time 2(1/2)-D head pose recovery for model-based video-coding. **IEEE Transactions on Instrumentation and Measurement**, [S.l.], v.50, n.4, p.1007–1013, 2001.

CORDEA, M. D. et al. 3-D head pose recovery for interactive virtual reality avatars. **IEEE Transactions on Instrumentation and Measurement**, [S.l.], v.51, n.4, p.640–644, 2002.

CORDEA, M. D.; PETRIU, E. M.; PETRIU, D. C. Three-dimensional head tracking and facial expression recovery using an anthropometric muscle-based active appearance model. **IEEE Transactions on Instrumentation and Measurement**, [S.l.], v.57, n.8, p.1578–1588, 2008.

DAHMANE, A.; LARABI, S.; DJERABA, C. Detection and analysis of symmetrical parts on face for head pose estimation. In: IMAGE PROCESSING (ICIP), 2010 17TH IEEE INTERNATIONAL CONFERENCE ON. **Anais...** [S.l.: s.n.], 2010. p.3249–3252.

DERVINIS, D. Head orientation estimation using characteristic points of face. **Electronics and Electrical Engineering.–Kaunas: Technologija**, [S.l.], v.8, p.72, 2006.

DONG, L. et al. Recognition of multi-pose head gestures in human conversations. In: FOURTH INTERNATIONAL CONFERENCE ON IMAGE AND GRAPHICS, Washington, DC, USA. **Proceedings...** IEEE Computer Society, 2007. p.650–654.

DORNAIKA, F.; RADUCANU, B. Three-dimensional face pose detection and tracking using monocular videos: tool and application. **Trans. Sys. Man Cyber. Part B**, [S.l.], v.39, p.935–944, August 2009.

FITZPATRICK, P. **Head pose estimation without manual initialization**. [S.l.]: AI Lab, MIT, Cambridge, USA, 2001. (A718434).

FOYTIK, J. et al. Phase space for face pose estimation. In: ADVANCES IN VISUAL COMPUTING 6TH INTERNATIONAL SYMPOSIUM, ISVC 2010, LAS VEGAS, NV, USA, NOVEMBER 29 - DECEMBER 1, 2010, PROCEEDINGS, PART III. **Anais...** Springer Berlin / Heidelberg, 2010. v.6455, p.49–58.

FOYTIK, J. et al. Head pose estimation from images using Canonical Correlation Analysis. In: APPLIED IMAGERY PATTERN RECOGNITION WORKSHOP (AIPR), 2010 IEEE 39TH. **Anais...** [S.l.: s.n.], 2010. p.1–7.

FUJIE, S. et al. A conversation robot using head gesture recognition as para-linguistic information. In: IEEE INTL. WORKSHOP ON ROBOT AND HUMAN COMMUNICATION, RO-MAN 2004, 13., Japan. **Anais...** [S.l.: s.n.], 2004. p.159–164.

GARY, H. et al. **Labeled Faces in the Wild**: a database for studying face recognition in unconstrained environments. [S.l.]: University of Massachusetts, Amherst, 2007. (07-49).

GOURIER, N. et al. Head pose estimation on low resolution images. In: STIEFELHAGEN, R.; GAROFOLO, J. (Ed.). **Multimodal Technologies for Perception of Humans**. [S.l.]: Springer Berlin / Heidelberg, 2007. p.270–280. (Lecture Notes in Computer Science, v.4122).

GOURIER, N.; HALL, D.; CROWLEY, J. Estimating face orientation from robust detection of salient facial features. In: POINTING 2004, ICPR, INTERNATIONAL WORKSHOP ON VISUAL OBSERVATION OF DEICTIC GESTURES, CAMBRIDGE, UK. **Proceedings...** [S.l.: s.n.], 2004.

GURBUZ, S.; OZTOP, E.; INOUE, N. Model free head pose estimation using stereovision. **Pattern Recognition**, [S.l.], v.In Press, Corrected Proof, p.–, 2011.

HUANG, C.; DING, X.; FANG, C. Head pose estimation based on random forests for multiclass classification. In: PATTERN RECOGNITION (ICPR), 2010 20TH INTERNATIONAL CONFERENCE ON. **Anais...** [S.l.: s.n.], 2010. p.934–937.

JI, Q. 3D Face pose estimation and tracking from a monocular camera. **Image and Vision Computing**, [S.l.], v.20, n.7, p.499–511, 2002.

KASS, M.; WITKIN, A.; TERZOPOULOS, D. Snakes: active contour models. **International Journal of Computer Vision**, [S.l.], v.1, n.4, p.321–331, 1988.

KRINIDIS, M.; NIKOLAIDIS, N.; PITAS, I. 3D head pose estimation using support vector machines and physics-based deformable surfaces. In: SIGNAL PROCESSING AND ITS APPLICATIONS, 2007. ISSPA 2007. 9TH INTERNATIONAL SYMPOSIUM ON. **Anais...** [S.l.: s.n.], 2007. p.1 –4.

LIN, C.; FAN, K.-C. Pose classification of human faces by weighting mask function approach. **Pattern Recogn. Lett.**, New York, NY, USA, v.24, p.1857–1869, August 2003.

LIU, X.; LU, H.; LI, W. Multi-manifold modeling for head pose estimation. In: IMAGE PROCESSING (ICIP), 2010 17TH IEEE INTERNATIONAL CONFERENCE ON. **Anais...** [S.l.: s.n.], 2010. p.3277 –3280.

LIU, X.; LU, H.; LUO, H. A new representation method of head images for head pose estimation. In: INTERNATIONAL CONFERENCE ON IMAGE PROCESSING, ICIP. **Proceedings...** [S.l.: s.n.], 2009. p.3585–3588.

MA, B. et al. Robust head pose estimation using LGBP. In: ICPR '06: PROCEEDINGS OF THE 18TH INTERNATIONAL CONFERENCE ON PATTERN RECOGNITION, Washington, DC, USA. **Anais...** IEEE Computer Society, 2006. p.512–515.

MA, B.; WANG, T. Head pose estimation using sparse representation. **Computer Engineering and Applications, International Conference on**, Los Alamitos, CA, USA, v.2, p.389–392, 2010.

MAHALANOBIS, P. C. On the generalised distance in statistics. In: NATIONAL INSTITUTE OF SCIENCE, INDIA. **Proceedings...** [S.l.: s.n.], 1936. v.2, n.1, p.49–55.

MARTINS, P.; BATISTA, J. Single view head pose estimation. In: IEEE INTERNATIONAL CONFERENCE ON IMAGE PROCESSING. **Anais...** [S.l.: s.n.], 2008. p.1652–1655.

MURPHY, E.; MANUBHAI, M. Head pose estimation in computer vision: a survey. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, Los Alamitos, CA, USA, v.31, p.607–626, 2009.

PAN, Y.; ZHU, H.; JI, R. 3-D head pose estimation for monocular image. In: INTERNATIONAL CONFERENCE ON FUZZY SYSTEMS AND KNOWLEDGE DISCOVERY , SPR. **Anais...** [S.l.: s.n.], 2005. v.3614, p.293–301.

RYDFALK, M. **CANDIDE, a parameterized face**. [S.l.]: Dept. of Electrical Engineering, Linköping University, Sweden, 1987. (4TH-ISY-I-866).

SFORZA, C. et al. **Three-dimensional analysis of active head and cervical spine range of motion** : effect of age in healthy male subjects. [S.l.]: Elsevier Science, 2002. v.17, n.8, p.611 – 614.

SHINN, Y. H.; HUI, L. H. An analytic solution for the pose determination of human faces from a monocular image. **Pattern Recognition Letters**, [S.l.], v.19, n.11, p.1045–1054, 1998.

SRINIVASAN, S.; BOYER, K. L. Head pose estimation using view based eigenspaces. In: INTERNATIONAL CONFERENCE ON PATTERN RECOGNITION (ICPR'02), 16., Washington, DC, USA. **Proceedings...** IEEE Computer Society, 2002. v.4, p.302–305.

TERISSI, L. D.; GOMEZ, J. C. 3D head pose and facial expression tracking using a single camera. **Journal of Universal Computer Science**, [S.l.], v.16, n.6, p.903–920, 2010.

TRIVEDI, M. M. Human movement capture and analysis in intelligent environments. **Machine Vision and Applications**, [S.l.], v.14, p.215–217, 2003.

TU, J. et al. Evaluation of head pose estimation for studio data. In: STIEFELHAGEN, R.; GAROFOLO, J. (Ed.). **Multimodal Technologies for Perception of Humans**. [S.l.]: Springer Berlin / Heidelberg, 2007. p.281–290. (Lecture Notes in Computer Science, v.4122).

VALENTI, R.; YUCEL, Z.; GEVERS, T. Robustifying eye center localization by head pose cues. In: COMPUTER VISION AND PATTERN RECOGNITION, 2009. CVPR 2009. IEEE CONFERENCE ON. **Anais...** [S.l.: s.n.], 2009. p.612–618.

VATAHSKA, T.; BENNEWITZ, M.; BEHNKE, S. Feature-based head pose estimation from images. In: IEEE-RAS INTERNATIONAL CONFERENCE ON HUMANOID ROBOTS (HUMANOIDS). **Proceedings...** [S.l.: s.n.], 2007. p.330–335.

VIOLA, P.; JONES, M. Rapid object detection using a boosted cascade of simple features. In: IEEE COMPUTER SOCIETY CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION. **Proceedings...** [S.l.: s.n.], 2001. p.511–518.

XINGUO, Y. et al. Head pose estimation in thermal images for human and robot interaction. In: INDUSTRIAL MECHATRONICS AND AUTOMATION (ICIMA), 2010 2ND INTERNATIONAL CONFERENCE ON. **Anais...** [S.l.: s.n.], 2010. v.2, p.698–701.

YANG, W.; ZHANG, L.; ZHU, W. Head pose estimation based on tensor factorization. In: ISHIKAWA, M. et al. (Ed.). **Neural Information Processing**. [S.l.]: Springer Berlin / Heidelberg, 2008. p.831–840. (Lecture Notes in Computer Science, v.4984).

YARI, Y.; CAVALCANTI, P.; SCHARCANSKI, J. Estimation of the head pose based on monocular images. In: IEEE INTERNATIONAL CONFERENCE ON IMAGE AND VISION COMPUTING NEW ZEALAND, 25., Queenstown, New Zealand. **Proceedings...** [S.l.: s.n.], 2010. p.To Appear.

YARI, Y.; SCHARCANSKI, J. An approach for fast human head pose estimation. In: SPIE DEFENSE, SECURITY AND SENSING (MOBILE MULTIMEDIA/IMAGE PROCESSING, SECURITY AND APPLICATIONS), Orlando, USA. **Anais...** [S.l.: s.n.], 2011. p.To appear.

ZABULIS, X.; SARMIS, T.; ARGYROS, A. **3D head pose estimation from multiple distant views**. [S.l.]: Citeseer, 2009. p.1–12.

APÊNDICE A ALGORITMO DE VIOLA JONES

A detecção de faces é geralmente o primeiro passo para a detecção de características faciais na maioria dos sistemas. Existem diversas técnicas que podem ser empregadas para esta tarefa, mas o detector de Viola-Jones tem sido muito utilizado recentemente. Este algoritmo é capaz de detectar faces com precisão, alta taxa de acerto, baixa taxa de falsos positivos e baixo custo computacional.

O algoritmo é realizado em três etapas, a primeira é a representação da imagem em um espaço de características baseadas nos filtros de Haar (usando a imagem integral). Na segunda etapa é feito a montagem de um classificador baseado em *Boosting* capaz de selecionar as características mais relevantes. Finalmente os classificadores para criar um filtro em cascata eficiente são combinados.

A representação dos dados de treinamento no espaço de características é obtida através da imagem integral $I(e, c)$. Ela é definida por:

$$I(e, c) = \sum_{e \geq e', c \geq c'} g(e', c'), \quad (\text{A.1})$$

onde $g(e, c)$ é uma imagem de tamanho $E \times C$, $1 \leq e, e' \leq E$ e $1 \leq c, c' \leq C$.

A principal vantagem desta representação é a possibilidade de calcular a soma dos elementos de qualquer retângulo contido na imagem usando só quatro pontos de $I(e, c)$. Com isto a representação pode ser calculada com apenas uma passada na imagem. Na abordagem original de Viola-Jones foram utilizadas quatro tipos de características (Figura A.1):

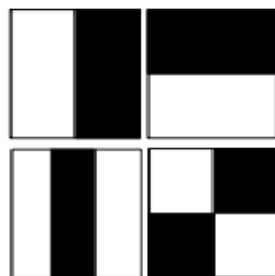


Figura A.1: Características utilizadas na abordagem do detector de objetos Viola-Jones.

onde o valor de uma dada característica é a diferença entre a soma dos *pixels* da região branca e a soma dos *pixels* da região preta:

$$pixels(\omega) = \sum^{\omega} R_{preta} - \sum^{\omega} R_{branco}. \quad (A.2)$$

Em versões mais recentes, um conjunto estendido de características é utilizado. Este novo conjunto inclui um novo tipo de característica e versões rotacionadas das características utilizadas na abordagem original.

O conjunto de treinamento do Viola-Jones é formado por amostras do tipo $(\mathbf{x}_\tau, y_\tau)$ com T amostras, onde \mathbf{x}_τ é uma imagem de dimensão 24×24 e $y_\tau = 0, 1$ é o rótulo de classe, onde $y_\tau = 1$ corresponde a uma imagem que contém uma face, caso contrário $y_\tau = 0$. A dimensão das amostras faz com que o número total de características seja maior que 180000. Isto torna necessário a seleção das características mais relevantes. Para tanto é utilizada uma versão do AdaBoost conhecida como Gentle Adaboost.

Em cada iteração do AdaBoost, um conjunto de classificadores fracos h_j é ajustado para minimizar o erro de classificação. Cada classificador possui uma característica $f_l(\mathbf{x}_\tau)$, onde $l = 1, \dots, J$, um linear ρ_l e uma paridade p_l . A regra de classificação é dada por:

$$[!t]h_j(\mathbf{x}_\tau) = \begin{cases} 1, & \text{se } p_l f_l(\mathbf{x}_\tau) > p_l \rho_l \\ 0, & \text{caso contrario} \end{cases} \quad (A.3)$$

onde a paridade p_l indica a direção da desigualdade. Em problemas práticos as taxas alcançadas por esta abordagem não são satisfatórias. Por isso é feita uma combinação de classificadores na forma de uma árvore degenerada, também conhecida como classificador de cascata.

Nesta combinação, cada estágio é chamado sequencialmente e corresponde a um classificador AdaBoost ajustado para obter uma taxa de falso negativo próxima a zero. O Adaboost é usado para selecionar características posto que é um algoritmo eficiente para encontrar um pequeno número de características relevantes no objeto. Este algoritmo compara as imagens que se encontram no conjunto positivo (aquele que possui a imagem do objeto) com as do conjunto negativo (não possui a imagem do objeto).

O processo de detecção está dado por um árvore de decisão (Figura A.2) chamado cascata:

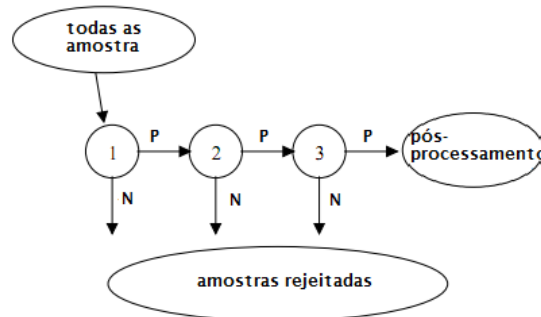


Figura A.2: Classificação de cascata realizada por Viola-Jones.

onde as imagens classificadas negativamente (N) são rejeitadas, e as classificadas positivamente (P) são passadas ao estágio seguinte.

Na detecção, como não se sabe a posição nem o tamanho da face na imagem de teste, as características selecionadas no treinamento são escalonadas do tamanho mínimo até o tamanho da imagem, nos respectivos estágios da cascata. Estas versões escalonadas são aplicadas em todas as amostras possíveis dentro da imagem de teste. A maioria delas são rejeitadas nos primeiros estágios, enquanto os últimos são responsáveis por classificar as amostras mais difíceis.

APÊNDICE B TRANSFORMAÇÕES GEOMÉTRICAS

As transformações geométricas, como rotação, translação e escala, modificam a posição, tamanho e orientação dos objetos e do observador numa cena. Elas permitem que um ponto ou um conjunto de pontos se transforme em outro dentro do sistema de referência global da cena. A partir das transformações geométricas mais comuns é possível construir outras.

As transformações geométricas podem ser divididas em dois grupos:

1. Transformações geométricas no espaço bidimensional (2D).
2. Transformações geométricas no espaço tridimensional (3D).

B.1 Transformações geométricas em 2D

1. Translação: Esta operação modifica a posição de um ou vários pontos no plano definidos pelas coordenadas (x, y) . Estes pontos se movimentam adicionando valores inteiros ao eixo X e Y. Ela pode ser escrita como:

$$\begin{aligned} Q' &= Q + T, \\ x' &= x + dx, \\ y' &= y + dy, \end{aligned} \tag{B.1}$$

onde dx e dy são as distâncias de translação do ponto Q ao ponto Q' . A fórmula matricial da translação é dada por:

$$Q' = \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & dx \\ 0 & 1 & dy \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}. \tag{B.2}$$

2. Escala: Modifica o tamanho dos objetos. Para definir esta operação tomam-se dois parâmetros importantes, s_x e s_y , chamados parâmetros de escala nos eixos X e Y.

$$\begin{aligned} Q' &= S \cdot Q, \\ x' &= s_x \cdot x, \\ y' &= s_y \cdot y. \end{aligned} \tag{B.3}$$

O ponto Q' é o resultado da multiplicação da forma inicial pelos parâmetros de escala, cuja equação matricial é dada por:

$$Q' = \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}. \quad (\text{B.4})$$

3. Rotação: Movimenta um ponto através de uma circunferência definida sobre um centro determinado seguindo um movimento circular. Matematicamente é dada por:

$$\begin{aligned} Q' &= R.Q, \\ x' &= x.\cos\theta - y.\sin\theta, \\ y' &= x.\sin\theta + y.\cos\theta, \end{aligned} \quad (\text{B.5})$$

onde o ângulo θ é o ângulo de rotação do ponto Q ao ponto Q'. Matricialmente pode-se escrever como:

$$Q' = \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}. \quad (\text{B.6})$$

B.2 Transformações geométricas em 3D

As transformações em 3D são uma generalização das 2D, tendo mesma capacidade de movimentar o objeto através da rotação, translação e escala. O sistema de coordenadas em 3D utiliza a regra da Mão Direita, onde o eixo Z é perpendicular ao papel, e saindo em direção ao observador. Neste caso, o ponto P é dado pelas três coordenadas x, y e z.

1. Translação: É uma extensão da translação em 2D:

$$Q' = \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & dx \\ 0 & 1 & 0 & dy \\ 0 & 0 & 1 & dz \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}, \quad (\text{B.7})$$

onde dx, dy, e dz indicam o deslocamento do ponto Q com respeito aos eixos, X, Y e Z. Esta equação pode também ser representada como:

$$Q' = T(d_x, d_y, d_z).Q. \quad (\text{B.8})$$

2. Escala: Também similar à 2D.

$$Q' = \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}. \quad (\text{B.9})$$

esta equação pode também ser representada como:

$$Q' = S(s_x, s_y, s_z).Q. \quad (\text{B.10})$$

3. Rotação: Esta se dará em qualquer um dos eixos, cada com um ângulo diferente. A rotação em 2D é justamente uma rotação em torno do eixo Z em 3D, ou seja:

$$R_\gamma = \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos \gamma & -\sin \gamma & 0 & 0 \\ \sin \gamma & \cos \gamma & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}. \quad (\text{B.11})$$

A matriz de rotação em relação ao eixo X é dado por:

$$R_\alpha = \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha & 0 \\ 0 & \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}. \quad (\text{B.12})$$

Já rotação em relação ao eixo Y é:

$$R_\beta = \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos \beta & 0 & \sin \beta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \beta & 0 & \cos \beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}. \quad (\text{B.13})$$

Nas rotações anteriores os ângulos α , β e γ indicam as rotações nos eixos X, Y e Z. Além disso, se queremos representar a rotação em um só eixo escreveremos:

$$Q' = Rt \cdot Q, \quad (\text{B.14})$$

onde Rt pode ser qualquer rotação (no eixo X ou Y ou Z).

Para obter a rotação total R com respeito aos três eixos (X, Y e Z) é preciso multiplicar as equações de rotação R_α , R_β e R_γ (BLANZ et al., 2004):

$$R_{\gamma,\alpha,\beta} = R_\gamma \cdot R_\alpha \cdot R_\beta, \quad (\text{B.15})$$

onde a multiplicação representa a rotação dos ângulos de Euler usando a convenção y-x-z.