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

LUIS FELIPE DE ARAUJO ZENI

Reconhecimento Facial Tolerante à Variação de Pose Utilizando uma Câmera RGB-D de Baixo Custo

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

Prof. Dr. Jacob Sharcanski
Orientador

Porto Alegre, fevereiro de 2014

CIP – CATALOGAÇÃO NA PUBLICAÇÃO

Zeni, Luis Felipe de Araujo

Reconhecimento Facial Tolerante à Variação de Pose Utilizando uma Câmera RGB-D de Baixo Custo / Luis Felipe de Araujo Zeni. – Porto Alegre: PPGC da UFRGS, 2014.

70 f.: il.

Dissertação (mestrado) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, BR–RS, 2014. Orientador: Jacob Sharcanski.

1. Reconhecimento Facial. 2. Reconhecimento de Padrões. 3. Visão Computacional. 4. Kinect. I. Sharcanski, Jacob. II. Título.

“I know nothing except the fact of my ignorance.”
— SOCRATES

AGRADECIMENTOS

É sempre importante poder retribuir àqueles que nos ajudam com algumas palavras de agradecimento. Embora sejam apenas palavras, elas são sinceras.

Agradeço primeiramente as pessoas que me deram a oportunidade de vir a este mundo, minha mãe Eloci e meu pai Luiz, sem eles, eu não existiria nestas circunstâncias, e muito menos este trabalho. Agradeço também pelo carinho, conselhos e ensinamentos que recebi dos meus pais. E mais importante obrigado por sempre terem me incentivado a seguir em frente apesar das minhas dificuldades.

Agradeço a todos os meus familiares, que de uma forma ou outra influenciaram a minha vida. Gostaria de deixar um agradecimento muito especial para minha avó Reomaria, a qual tenho um carinho muito especial e que me ensinou a enxergar a vida sempre de forma positiva.

Gostaria de deixar um agradecimento muito especial a minha companheira Carolina. Muito obrigado por me aturar nestes 5 anos, pelo apoio, amor, amizade e paciência. E mais importante obrigado por me aceitar como eu sou.

Aos meus amigos.

É claro que não poderia deixar de agradecer ao meu excelentíssimo orientador Jacob, afinal este trabalho não existiria nestas circunstancias se não fosse por ele. Assim agradeço aos ensinamentos, a aceitação da minha inscrição, passando por todas as etapas de desenvolvimento deste trabalho e até chegar agora à conclusão desta etapa.

Aos colegas de laboratório e disciplinas os quais aprendi muito durante este período.

Agradeço a empresa Guardian pelo suporte financeiro para minha pesquisa, gostaria de agradecer ao Thiarlei Macedo da Guardian por ter realizado a tarefa árdua de gravar a base de faces com restrições utilizada neste trabalho.

Agradeço a todas as pessoas que cederam a imagem de suas faces para realizar os experimentos do meu trabalho.

A todas as outras pessoas que não lembrei de mencionar, mas que cruzaram a minha vida ou de alguma forma contribuíram para a conclusão desta jornada, Muito Obrigado!

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

PCA	*Principal Component Analysis*
LDA	*Linear Discriminant Analysis*
FLD	*Fisher Linear Discriminant*
ICP	*Iterative Closest Point*
AAM	*Active Appearance Model*
RGB-D	*Red Green Blue and Depth*.
HOG	*Histogram of Oriented Gradient*
SDK	*Software Development Kit*
UFRGS	Universidade Federal do Rio Grande do Sul

SUMÁRIO

RESUMO

Reconhecer a identidade de seres humanos a partir de imagens digitais gravadas de suas faces é uma etapa importante para uma variedade de aplicações que incluem segurança de acesso, iteração humano computador, entretenimento digital, entre outras. Neste trabalho é proposto um novo método automático para reconhecimento facial que utiliza simultaneamente a informação 2D e 3D de uma câmera RGB-D(Kinect). O método proposto utiliza a informação de cor da imagem 2D para localizar faces na cena, uma vez que uma face é localizada ela é devidamente recortada e normalizada para um padrão de tamanho e cor. Posteriormente com a informação de profundidade o método estima a pose da cabeça em relação com à câmera. Com faces recortadas e suas respectivas informações de pose, o método proposto treina um modelo de faces robusto à variação de poses e expressões propondo uma nova técnica automática que separa diferentes poses em diferentes modelos de faces. Com o modelo treinado o método é capaz de identificar se as pessoas utilizadas para aprender o modelo estão ou não presentes em novas imagens adquiridas, as quais o modelo não teve acesso na etapa de treinamento. Os experimentos realizados demonstram que o método proposto melhora consideravelmente o resultado de classificação em imagens reais com variação de pose e expressão.

Palavras-chave: Reconhecimento Facial, Reconhecimento de Padrões, Visão Computacional, Kinect.

Face Recognition Using an Low Cost RGB-D Camera to Deal With the Problem of Pose Variation.

ABSTRACT

Recognizing the identity of human beings from recorded digital images of their faces is important for a variety of applications, namely, security access, human computer interation, digital entertainment, etc. This dissertation proposes a new method for automatic face recognition that uses both 2D and 3D information of an RGB-D(Kinect) camera. The method uses the color information of the 2D image to locate faces in the scene, once a face is properly located it is cut and normalized to a standard size and color. Afterwards, using depth information the method estimates the pose of the head relative to the camera. With the normalized faces and their respective pose information, the proposed method trains a model of faces that is robust to pose and expressions using a new automatic technique that separates different poses in different models of faces. With the trained model, the method is able to identify whether people used to train the model are present or not in new acquired images, which the model had no access during the training phase. The experiments demonstrate that the proposed method considerably improves the result of classification in real images with varying pose and expression.

Keywords: Face Recognition, Pattern Recognition, Computer Vision, Kinect.

1 INTRODUÇÃO

A face humana é um excelente descritor de identidade. Não é ao acaso que os seres humanos têm a habilidade de reconhecer uns aos outros pelas características faciais. Essa é uma tarefa que realizamos com muita facilidade em nosso cotidiano e que tem extrema importância para nossas relações sociais. Com o surgimento da fotografia digital, computadores mais poderosos e mais baratos. Surgiu um grande interesse em aplicações de visão computacional e reconhecimento de padrões que sejam capazes de reconhecer faces automaticamente. Alguns exemplos de aplicações de métodos de reconhecimento facial incluem, validação de identidade, controle de acesso, vigilância visual e interação humano computador (JAIN; LI, 2005).

Um sistema de reconhecimento facial consiste geralmente em quatro etapas (módulos) principais: detecção de faces, alinhamento das faces detectadas, extração de feições e busca de faces correspondentes (JAIN; LI, 2005). A Figura 1.1 demonstra em forma de diagrama as quatro etapas principais de um sistema de reconhecimento facial.

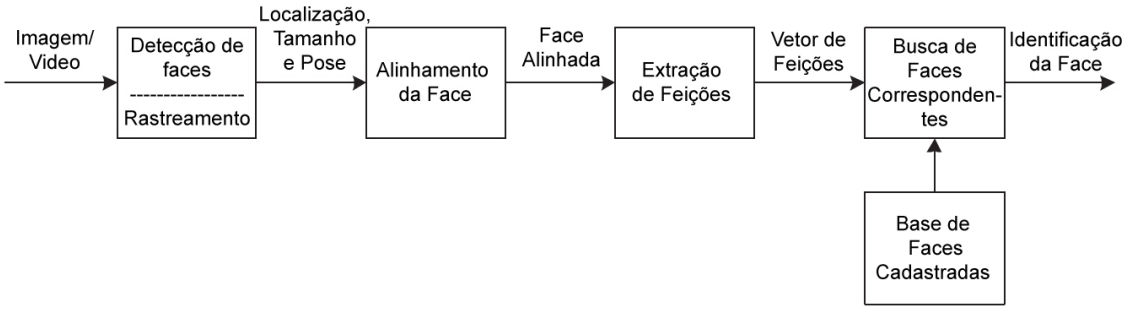


Figura 1.1: Fluxo padrão de um sistema de reconhecimento facial (JAIN; LI, 2005).

A etapa de detecção de faces tem como objetivo encontrar prováveis candidatos de faces humanas em uma imagem. Usualmente esses candidatos de faces são segmentados da imagem original, desta forma as regiões sem faces na imagem serão descartadas no restante do processo. No caso de um vídeo, uma face detectada pode ser rastreada utilizando um rastreador de faces para reduzir o custo computacional.

Faces que foram detectadas serão então processadas na etapa de alinhamento. Esta etapa tem como objetivo ajustar e normalizar as faces para um determinado padrão, à fim de melhorar as imagens segmentadas na etapa de detecção de faces. Componentes da face como olhos, nariz, boca e silhueta da face são localizadas e utilizadas para normalizar as faces para respeitarem algumas propriedades geométricas como tamanho e pose. A face também pode ser normalizada para respeitar algumas propriedades fotométricas como iluminação (JAIN; LI, 2005).

As etapas de detecção e alinhamento de faces são realizadas antes de iniciar o reconhecimento de faces propriamente dito(extração de feições e busca de faces correspondentes utilizando as feições extraídas).

Após a etapa de normalização de faces, é realizada a etapa de extração de feições, onde é extraída a informação que melhor discrimine uma determinada face, facilitando assim a distinção entre diferentes faces. Usualmente a informação de feições extraída é representada em forma de um vetor n-dimenssional.

Para encontrar faces correspondentes, o vetor de feições extraído da face de entrada é comparado com os vetores de feições das faces armazenadas em uma base de faces. Se os vetores de feições da face de entrada e alguma face da base de faces forem muito parecidas, ou seja, a distância entre os vetores estiver abaixo de um determinado limiar, o sistema terá como saída que ambas as faces pertencem a mesma pessoa. Caso contrário, a face de entrada não é conhecida pelo sistema, e portanto, não pode ser reconhecida (JAIN; LI, 2005).

Os métodos de reconhecimento facial podem ser divididos em dois grupos principais, dependendo do tipo de informação que é utilizada para o reconhecimento, no caso, existem os métodos que utilizam informação 2D e os métodos que utilizam informação 3D. Os métodos 2D utilizam informação de textura e cor providos por câmeras comuns, já os métodos 3D utilizam a informação geométrica da forma da face para o reconhecimento facial. Existem também os métodos híbridos que utilizam tanto informação 2D quanto 3D para o reconhecimento facial (ABATE et al., 2007).

Atualmente capturar informação 2D de uma face é muito simples, pois esse tipo de sensor está amplamente disponível em *smartphones*, *notebooks* e câmeras digitais. E embora já existam avanços significantes, os métodos de reconhecimento de faces 2D ainda tem limitações devido à variação de pose, iluminação, expressões e envelhecimento. Alguns desses problemas como à variação de pose e iluminação são facilmente resolvidos com o uso de informação 3D, provida por um scanner 3D (BOWYER; CHANG; FLYNN, 2005). Alguns dos problemas dos scanners 3D são: os dispositivos com boa qualidade são muito caros, ocupam muito espaço físico e possuem velocidade de aquisição limitada.

Até o ano de 2010, não existia uma alternativa de equipamento no mercado, com preço razoável para adquirir imagens 3D. Porém, em novembro de 2010 a Microsoft em parceria com a Prime Sence revolucionou a forma de jogar vídeo games com o lançamento do sensor de movimentos Kinect para o Xbox 360. Para estimar o movimento do jogador o sensor utiliza informação 2D e 3D da cena juntamente com algoritmos de visão computacional. Além de ter entrado para o livro dos recordes, como equipamento eletrônico mais vendido no período de dois meses na história (WEBB; ASHLEY, 2012), o Kinect foi a primeira alternativa de baixo custo para adquirir imagens 2D e 3D de uma cena em tempo real. A Tabela 1 demonstra alguns dos sensores 3D disponíveis no mercado com seus respectivos preços em dólares, velocidade de aquisição e acurácia. Como pode-se verificar, embora o Kinect seja o dispositivo com o menor preço e com ótima velocidade de aquisição, ele tem a pior acurácia entre os sensores. Isso se da pelo fato de o Kinect ter uma baixa resolução. Ao fazer um comparativo visual entre um quadro adquirido com o Kinect e um quadro adquirido com um scanner 3D de qualidade superior, é visível a diferença de qualidade. A Figura 1.2 demonstra a diferença de resolução entre uma face capturada com o Kinect e um scanner 3D da base de faces 3D FRGC v2(PHILLIPS et al., 2005).

Motivado pelo recente surgimento do Kinect e a possibilidade de utilizar a informação de profundidade para criar um método mais robusto à variação de poses. Este trabalho tem

Dispositivo	Velocidade (seg)	Tempo de Carrregamento	Tamanho (Polegadas^3)	Preço (USD)	Precisão (mm)
3dMD	0.002	10 seg	N/A	>$50k	<0.2
Minolta	2.5	Não	1408	>$50k	0.1
Artec Eva	0.063	Não	160.5	>$20k	0.5
3D3 HDI R1	1.3	Não	N/A	>$10k	>0.3
SwissRanger	0.02	Não	17.53	>$5k	10
DAVID SLS	2.4	Não	N/A	>$2k	0.5
Kinect	0.033	Não	41.25	<$200	1.5-50

Tabela 1.1: Comparação entre alguns scanners 3D disponíveis para venda no mercado (LI et al., 2013).

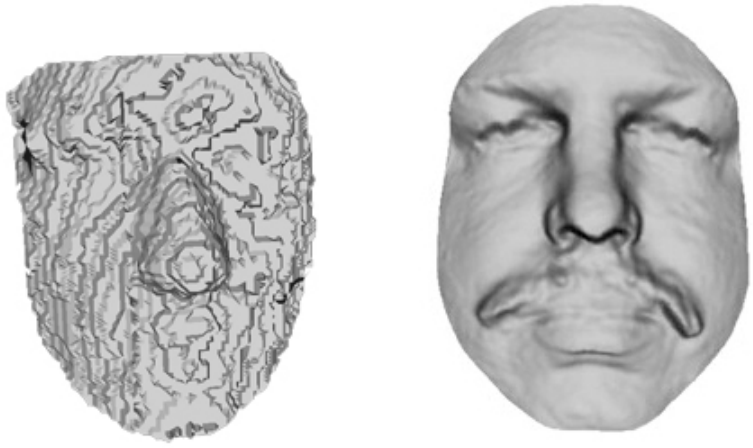


Figura 1.2: Comparação entre uma face capturada pelo Kinect(esquerda) e uma face capturada por um scanner 3D(direita) retirada da base de faces FRGC v2(PHILLIPS et al., 2005).

como objetivo explorar o uso do Kinect para desenvolver um método que utilize as imagens de cor e profundidade capturadas por um Kinect para propósitos de reconhecimento facial, tendo como objetivo principal um método tolerante à variação de poses.

A primeira etapa do método proposto é a detecção de faces nas imagens capturadas pelo Kinect que é feita com o método *Active Apearance Models*(AAM) (ZHOU et al., 2010), o qual além de detectar faces nas imagens, ajusta um conjunto de pontos pré-estipulados na face detectada.

Utilizando os pontos ajustados pelo AAM é realizada uma etapa de recorte das faces nas imagens coloridas e de profundidades. Onde apenas a informação relevante das faces é mantida. Após o recorte, as imagens dessas faces são normalizadas para um tamanho padrão. Posteriormente, a imagem colorida é normalizada para escalas de cinza. E a imagem de profundidade é suavizada com um filtro da mediana e normalizada para valores entre 0 e 255.

Para lidar com o problema de variação de pose a maioria dos métodos 3D utilizam uma normalização de pose para todas as faces, onde normalmente as faces são rotacionadas para ficarem em pose frontal. Posteriormente estes métodos extraem feições dessas faces normalizadas. Entretanto extrair boas feições unicamente das imagens de profundidade

adquiridas por um Kinect é um desafio, dada a baixa resolução e acurácia do sensor. Para lidar com esse problema o método proposto divide o problema de classificação em subproblemas de classificação, onde cada um aprende um modelo de face especialista em um tipo distinto de pose.

Estimar a pose da cabeça em relação a câmera é uma etapa importante para o método proposto. Esta estimativa é feita utilizando a informação de profundidade adquirida pelo kinect. Foi utilizado método de alinhamento de nuvens de pontos *Iterative Closest Point* (BESL; MCKAY., 1992) para realizar a estimativa da pose da cabeça. Embora existam métodos de estimativa de pose em 2D, utilizar a informação 3D para estimar a pose da cabeça traz duas vantagens: precisão e maior velocidade de processamento (FANELLI et al., 2011).

Para aprender os modelos especialistas em pose, o método proposto necessita de um conjunto de imagens de faces de treinamento, as quais devem estar recortadas, normalizadas e com a pose estimada. Com a informação de pose das faces de treinamento, o método proposto separa estas faces em diferentes grupos de pose. Para separar as faces em diferentes grupos de pose, o método proposto aplica o método de agrupamento K-means no "espaço de poses", onde serão encontrados K agrupamentos de pose. Para cada agrupamento e suas imagens correspondentes é treinado um modelo de faces Fisherfaces (BELHUMEUR; HESPANHA; KRIEGMAN, 1997) utilizando as informações de cor e profundidades providas pelo Kinect. Este método foi nomeado de K-fisherfaces e é a principal contribuição deste trabalho.

A validação do método proposto foi realizada utilizando duas bases de faces diferentes, as quais foram gravadas pelos autores desse trabalho. A primeira base é composta de um conjunto de imagens de diferentes pessoas em 4 poses e expressões padronizadas. A segunda base é composta de um conjunto de vídeos de pessoas movimentando a cabeça livremente na cena. Foram realizados quatro experimentos, onde os resultados obtidos com o método em ambas as bases foi superior aos outros métodos testados.

1.1 Objetivos

Como objetivo geral deste trabalho foi definida a elaboração e descrição de uma técnica de reconhecimento facial, que utilize as imagens de cor e profundidade capturadas por um Kinect a qual deve ser tolerante a variação de poses. Para isso, é necessário que a técnica aqui apresentada possua as seguintes características:

- tolerância à variação de pose;
- tolerância à variação de expressões faciais;
- capacidade de detectar faces nas imagens fornecidas e segmentar apenas a informação relevante;
- capacidade de estimar a pose da cabeça na cena utilizando a informação de profundidade;
- capacidade de identificar a pessoa presente na imagem;

Para atingir esse objetivo estas tarefas foram subdivididas entre os seguintes objetivos específicos:

1. Desenvolvimento de um método capaz de localizar, segmentar e normalizar faces presentes em imagens coloridas e estimar a pose da face em relação à câmera utilizando a imagem de profundidades.

2. Desenvolvimento um método capaz de aprender um modelo de faces que seja robusto à variação de poses.

3. Validar, analisar e comparar o desempenho do método proposto utilizando imagens reais.

1.2 Organização do trabalho

Este trabalho está estruturado em 6 capítulos, como descrito a seguir. No Capítulo 2 são descritos brevemente os principais trabalhos relacionados ao problema que propoem-se a solucionar. Esses trabalhos compõem o estado da arte de reconhecimento de faces utilizando o Kinect. No capítulo 3 é realizada uma breve revisão dos conceitos teóricos de visão computacional e reconhecimento de padrões necessária para a compreensão do método desenvolvido. O Capítulo 4 destina-se à descrição detalhada do método de reconhecimento facial proposto neste trabalho. O Capítulo 5 apresenta e discute alguns resultados obtidos com a metodologia proposta. Finalmente, no Capítulo 6, são apresentadas as conclusões dos estudos e experimentos obtidas neste trabalho. Também neste capítulo, são realizadas algumas propostas e sugestões para trabalhos futuros.

2 ESTADO DA ARTE

Existem inúmeros métodos de detecção 2D e 3D para reconhecimento facial e também já foram realizadas inúmeras *surveys* comparando estes métodos 2D e 3D (ABATE et al., 2007; SCHEENSTRA; RUIFROK; VELTKAMP, 2005a; JAIN; LI, 2005). Neste capítulo serão abordados apenas os métodos mais recentes que compõem o estado da arte de métodos de reconhecimento facial que utilizam imagens adquiridas por um Kinect. Nas sub-seções a seguir, são apresentados os métodos em questão.

2.1 An RGB-D Database Using Microsoft Kinect for Windows for Face Detection

O trabalho realizado por Hg e Jasek em 2012 (HG et al., 2012) foi o primeiro método a utilizar imagens gravadas por um kinect para fins de reconhecimento facial. Para realizar os testes os autores gravaram uma base de imagens de faces utilizando um kinect, a qual contém 31 pessoas em 17 poses e expressões diferentes. Esse método utiliza apenas o mapa de profundidades do kinect para realizar o reconhecimento facial, ou seja, os autores ignoram a informação de cor que é provida pelo Kinect.

Nesse método, os autores assumem que a pessoa que contém a face a ser localizada, é o objeto mais próximo da câmera. Portanto, a região do objeto mais próxima da câmera é selecionada para ser a região de busca da face. Após segmentar uma pessoa da cena como sendo o objeto mais próximo, o método aplica um filtro da média com uma janela de tamanho 13×13 para preencher possíveis buracos oriundos de falta de informação na imagem de profundidades.

Após a etapa de preenchimento dos buracos na imagem de profundidade, essa imagem é então utilizada para detectar possíveis candidatos de face, para realizar esta etapa é utilizado o método de análise de curvaturas HK-Classification (COLOMBO; CUSANO; SCHETTINI, 2006), sendo H é descrito por:

$$H(x,y) = \frac{(1+f_y^2)f_{xx} - 2f_x f_y f_{xy} + (1+f_x^2)f_{yy}}{2(1+fx^2+f_y^2)^{3/2}}, \tag{2.1}$$

e o K por:

$$K(x,y) = \frac{f_{xx}f_{xy} - f_{xy}^2}{2(1+fx^2+f_y^2)^2}, \tag{2.2}$$

onde f é a imagem de profundidades e f_x, f_y, f_{xx}, f_{xy}, f_x^2 e f_y^2 são as derivadas de primeira e segunda ordens correspondentes na posição (x, y). O método *HK-Classification* é utilizado para determinar o tipo de superfície que um determinado pixel da imagem de profundidades pertence.

O tipo de superfície que um determinado pixel de profundidade pertence será utilizado para determinar se o pixel é ou não pertencente a um provável candidato de nariz ou olhos. Para este propósito, as duas medidas de curvatura apresentadas anteriormente são limiarizadas. O limiar para determinar se um pixel pertence ao nariz é $K > T_{K_{nose}}$ e $H > T_{H_{nose}}$ onde $T_{K_{nose}}$ e $T_{H_{nose}}$ são definidos a priori. De forma similar, para determinar se um pixel pertence aos olhos, são utilizados os limiares $K > T_{K_{eye}}$ e $H > T_{H_{eye}}$ onde $T_{K_{eye}}$ e $T_{H_{eye}}$ são definidos a priori. Ao analisar a imagem de profundidades limiarizada gerada por estes limiares, pode-se notar os possíveis candidatos a olhos e nariz. Para determinar as regiões de possíveis candidatos a olhos e nariz é utilizada a relação 4×4 dos vizinhos de cada pixel para conectar pixels vizinhos pertencentes ao mesmo candidato de nariz ou olhos. A Figura 2.1(c) demonstra uma face em uma imagem de profundidades com os possíveis candidatos de olhos e nariz detectados. Como pode-se verificar o método detecta inúmeros candidatos, criando a necessidade de selecionar um conjunto de candidatos que melhor se ajustem a face.

A seleção do melhor conjunto de candidatos de olhos e nariz é feita unindo os possíveis candidatos em um conjunto triangular, onde cada vértice do triangulo será um elemento da face(os dois olhos e o nariz). São impostas as seguintes restrições: os olhos devem estar acima do nariz e o triângulo deve ser aproximadamente equilátero, a Figura 2.1(d) demonstra um exemplo de um candidato de olhos e nariz selecionados.

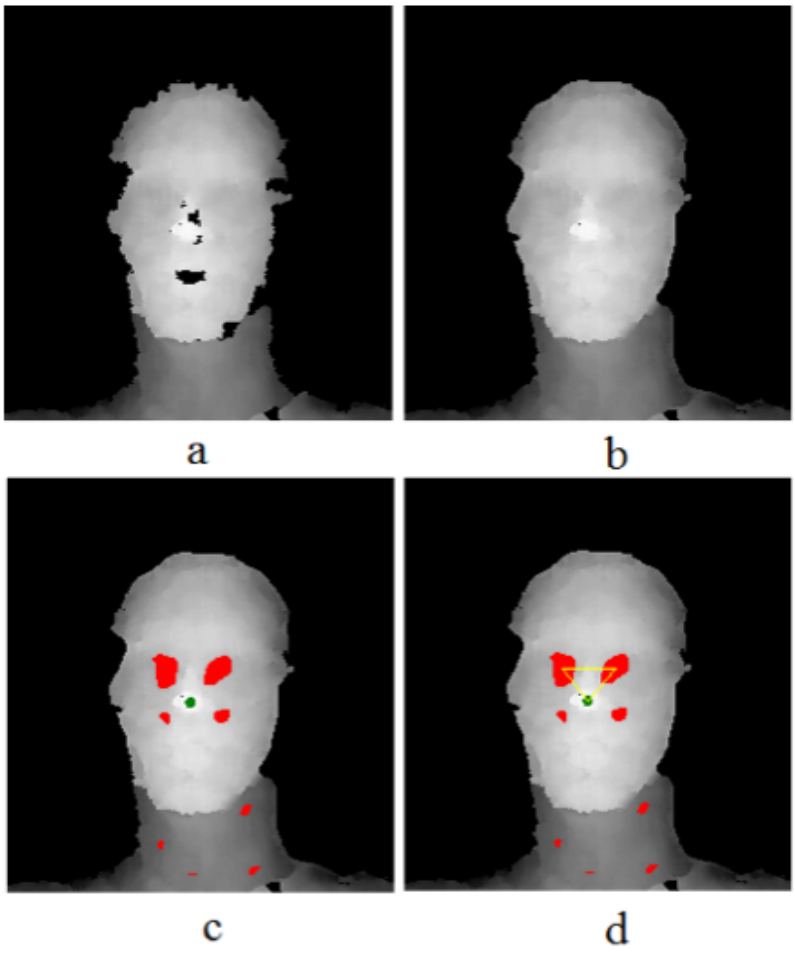


Figura 2.1: Processo de ajuste do triângulo na face proposto por (HG et al., 2012).

O melhor conjunto de candidatos em forma de triângulo encontrado é utilizado para

normalizar as faces para a posição frontal, isto é feito aplicando uma translação e duas rotações as quais mapeiam o nariz para o centro da imagem e mapeiam os outros pixels correspondentemente.

Com as faces normalizadas o método utiliza PCA (*Principal Component Analisys*) para selecionar as principais componentes de um conjunto de faces de treinamento. A matriz de projeção encontrada com a PCA é considerada o modelo de faces do método e é utilizada posteriormente para classificar as faces.

Um problema desse método, é a utilização dos limiares para selecionar os candidatos de nariz e olhos, isto torna o método pouco genérico, pois se a base de faces for trocada, provavelmente vai ser necessário reajustar os limiares. Outro problema são as restrições, primeiro a face deve ser o objeto mais próximo da câmera, se duas pessoas estiverem na cena, o método irá detectar apenas uma. Em seguida os olhos devem estar sempre acima do nariz, o que impede que exista grande rotação no eixo Z da face, alem de que, caso a pessoa estiver com um dos olhos oclusos, o método não vai ser capaz de localizar os candidatos corretamente, pois espera-se encontrar os dois olhos para ajustar o triângulo. O método também ignora a informação descritiva que a imagem colorida pode fornecer.

2.2 Using Kinect for face recognition under varying poses, expressions, illumination and disguise

No método de Li e Mian (LI et al., 2013), a face é representada como uma nuvem de pontos 6D $(XYZ - RGB)$ onde os pixels das imagens colorida e de profundidade foram retificadas a priori. Para realizar a detecção de faces o método necessita que uma pessoa localize manualmente um ponto na ponta do nariz das faces presentes nas imagens.

Dado o ponto da ponta do nariz que foi localizado por uma pessoa, o método recorta a face da nuvem de pontos utilizando uma esfera de 8cm de raio, pontos da nuvem que estiverem fora da esfera são descartados, já os pontos que estiverem dentro são considerados como pertencentes a face localizada manualmente.

As faces recortadas tem então suas poses corrigidas, para realizar esta etapa é utilizado *Iterative Closest Point*(ICP) (BESL; MCKAY., 1992) para alinhar a nuvem de pontos da face de entrada com uma face em pose padrão. Como as imagens do Kinect são muito ruidosas é utilizado para o alinhamento como face padrão uma face criada a partir da base de imagens de scanner 3D FRGC (PHILLIPS et al., 2005).

Após o alinhamento das faces, o método espelha um dos lados da face para completar o outro lado com informação nas regiões onde a face não tem informação disponível, segundo os autores o efeito causado pela assimetria da face é menor que o efeito causado por identidades diferentes (LI et al., 2013). Finalmente a nuvem de pontos da face resultante é suavizada e tem o tamanho reajustado.

A Figura 2.2 demonstra os passos de recorte, alinhamento com ICP, preenchimento utilizando a informação de simetria da face e o resultado final da nuvem de pontos da face suavizada.

Antes de extrair feições das nuvens de pontos das faces, o método converte o espaço de cores das coordenadas R, G, B da nuvem de pontos, para isto é utilizada a técnica de conversão de espaço de cores *Discriminant Color Space(DCS)* (SCHEENSTRA; RUIFROK; VELTKAMP, 2005b). Segundo os autores deste método o espaço de cores RGB é ruim para o reconhecimento facial dada a correlação intra-componentes (LI et al., 2013). O espaço de cores DCS procura um conjunto de combinações para as componentes R, GeB de forma que maximize a separabilidade entre classes, uma idéia similar ao LDA.

Para extrair feições das nuvens de pontos o método utiliza a técnica *Sparse Representation Classifier* (WRIGHT et al., 2009) onde a informação de profundidade e cor são ambas utilizadas na extração de feições. No método é utilizado o algoritmo SRC multi-modal para o reconhecimento facial. Mais especificamente, SRC é aplicado separadamente nas informação de cor e profundidade. Como a imagem a cores tem três canais, os canais são primeiramente transformados em um vetor que une os três canais.

O problema desse método é a necessidade de que a informação da ponta do nariz seja informada por uma pessoa. Uma pessoa consegue detectar a ponta do nariz de faces de forma muito mais eficiente do que os detectores de ponta de nariz atuais, acarretando que, o método acaba não recebendo falsas detecções de nariz. E isso acarreta que o método vai ter um resultado provavelmente melhor, dada a intervenção humana para a detecção.

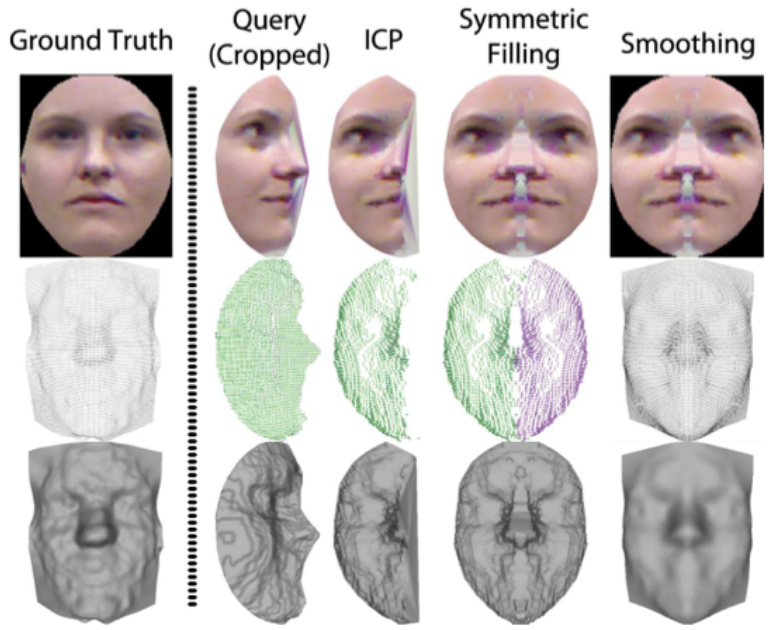


Figura 2.2: Processo de recorte, alinhamento, espelhamento e suavização da face proposta por (LI et al., 2013).

2.3 On RGB-D Face Recognition using Kinect

Em (GOSWAMI et al., 2013) os autores propõem um método que utiliza ambas as informações 2D e 3D providas pelo Kinect. Para detectar a face nas imagens os autores utilizam o método Viola Jones (VIOLA; JONES, 2001) nas imagens coloridas, a região do retângulo da face detectada pelo Viola Jones é utilizada para recortar a face na imagem colorida, a mesma região é recortada na imagem de profundidades. A Figura 2.3(a) demonstra um exemplo de uma face recortada na imagem colorida e a Figura 2.3(d) a mesma face recortada na imagem de profundidades.

Após o recorte, o método computa quatro mapas de entropia correspondentes à imagem colorida e de profundidade, variando os tamanhos das correspondências. Também é calculado um mapa de saliências visuais na imagem colorida. As Figuras 2.3(b) e (e) demonstram os mapas de entropia da imagem colorida e de profundidades respectivamente. A Figura 2.3(e) demonstra o mapa de saliências da imagem colorida.

Entropia é definida como uma medida de incerteza em uma variável aleatória (RRNYI.,

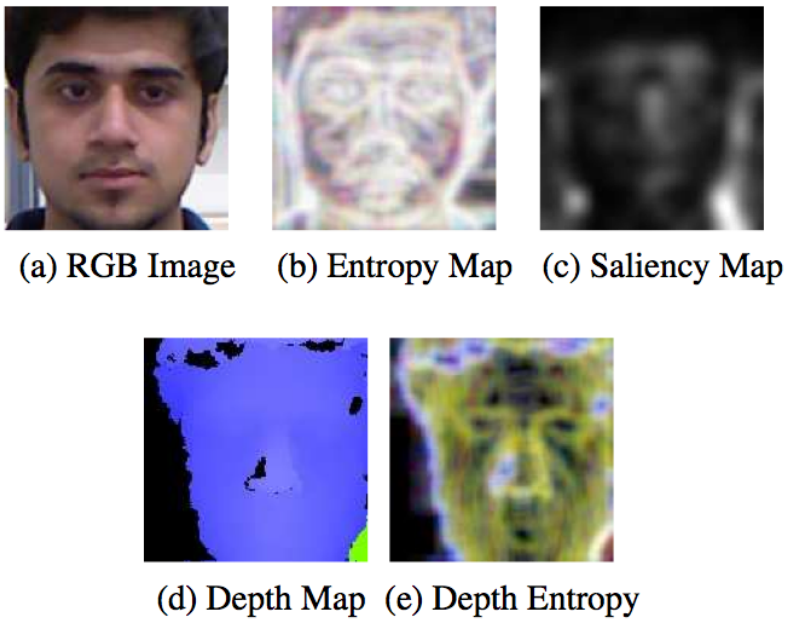


Figura 2.3: Mapas de entropia e de saliências calculados por (GOSWAMI et al., 2013).

1961). A entropia H de uma variável aleatória x é $H(x) = -\sum_{i=1}^{n} p(x_i) log_b p(x_i)$, onde $p(x_i)$ é valor da função de densidade probabilidade de x_i. Sendo a imagem de entrada denotada como um par de funções de intensidade, $[Irgb(x, y), I_d(x, y)]$, onde $Irgb(x, y)$ é a imagem rgb e $Id(x, y)$ é a imagem de profundidades, cada imagem de tamanho $M \times N$. Sendo ambas definidas com o mesmo conjunto de pontos (x, y), Z, de tal modo que, $x \in [1, M]$ e $y \in [1, M]$. Sendo que $H(I_j)$ denota o mapa de entropia da imagem I_j. Aqui, I_j pode ser tanto a imagem de cores quanto profundidades ou uma pequena parte destas imagens. Duas imagens de patches são extraídas de ambas $Irgb$ e Id. Dois patches P_1, de tamanho $\frac{M}{2} \times \frac{N}{2}$ centralizado em $[\frac{M}{2} \times \frac{N}{2}]$ e P_2, de tamanho $\frac{3M}{4} \times \frac{3N}{4}$ centralizado em $[\frac{M}{2} \times \frac{N}{2}]$, são extraídos de I_{rgb}; De forma similar, dois patches P_3 e P_4 são extraídos de I_d. Quatro mapas de entropia E_1, E_2, E_3, E_4 são calculados para os patches P_1, P_2, P_3, P_4 utilizando a equação:

$$E_i = H(P_i), onde, i \in [1, 4] \tag{2.3}$$

Além do mapa de entropia, também é calculado o mapa de saliências da imagem RGB. Saliências visuais estão relacionadas com a capacidade de uma determinada região de uma imagem em atrair a atenção visual de um espectador (GOSWAMI et al., 2013). A distribuição de atenção visual em uma imagem inteira é determinada como o mapa de saliências de uma imagem. Sendo a imagem representada como $I(x, y)$. O mapa de saliência pode ser denotado como uma função de intensidades $S(.)$, a qual mapeia pixels individuais para um valor de intensidade proporcional a saliência deste determinado pixel. A Figura 2.3(c) demonstra um exemplo do mapa de saliências calculado de uma imagem colorida. Existem várias técnicas para calcular o mapa de saliências de uma imagem (ITTI; KOCH; NIEBUR., 1998). Dado que, os métodos de saliência visual foram desenvolvidos para imagens coloridas e não para imagens de profundidades, tais técnicas tendem a retornar resultados irregulares para uma imagem de profundidades. O método extrai o mapa de saliências S_1 da imagem colorida I_{rgb} utilizando a seguinte equação:

$$S_i(x, y) = S(I_{rgb}(x, y) \forall (x \in [1, M], y \in [1, N])) \tag{2.4}$$

A extração de feições é feita utilizando *Histogram of Oriented Gradients*(HOG) (DALAL; TRIGGS., 2005), um descritor HOG produz um histograma de uma dada imagem na qual os pixels são guardados conforme a magnitude e direção de seus gradientes. É um descritor robusto com um tamanho fixo para as feições e vem sendo utilizado com sucesso em diversas aplicações de detecção e reconhecimento (GOSWAMI et al., 2013). HOG é aplicado em todos os mapas de entropia e de saliência. Um vetor com a feições HOG extraídas das imagens de entropia e saliências é então gerado concatenando todas as feições em um único vetor de feições. Este vetor será utilizado, então, para realizar a classificação de faces. Como classificador o método utiliza *Random Decision Forests* (HO., 1995) que é um classificador que lida muito bem com problemas de multi-classes.

Um problema deste método é que ele utiliza o detector de faces Viola Jones, o qual apenas informa o retângulo da face, isto ocasiona que fragmentos do fundo não são corretamente eliminados neste método e estes artefatos podem interferir na qualidade final de classificação do método.

3 FUNDAMENTOS TEÓRICOS

3.1 Visão Computacional

Em visão computacional tenta-se descrever o mundo o qual nós vemos em uma ou mais imagens e reconstruir suas propriedades, como forma, iluminação, e distribuição de cores. Os seres humanos e animais, com seus sistemas de visão, realizam esta tarefa com grande facilidade. Enquanto algoritmos de visão computacional ainda são propensos a erros ou ineficientes (SZELISKI, 2011). Alguns exemplos de algoritmos de visão computacional que são amplamente utilizados no mundo real hoje em dia incluem: reconhecimento de caracteres, inspeção de qualidade na indústria, reconhecimento de objetos, reconstrução 3D de cenas, processamento de imagens médicas, segurança de veículos, contagem de veículos, reconhecimento facial, entre outros.

3.1.1 Formação de uma Imagem

Quando modela-se o processo de formação de uma imagem colorida, descreve-se como feições geométricas 3D no mundo são projetadas em 2D em uma imagem. Entretanto, imagens não são compostas de feições 2D. Em vez disto elas são formadas de valores discretos de intensidade de cor. A Figura 3.1 exemplifica como funciona o processo de formação de uma imagem 2D, para produzir uma imagem 2D é indispensável ter um emissor de fonte de luz, a luz é emitida e então refletida na superfície de um objeto 3D do mundo. Um porção desta luz refletida é direcionada para à câmera. Uma vez que a luz da cena chega na câmera, ela precisa passar pelas lentes da câmera, as quais podem corrigir ou distorcer os raios de luz. Finalmente, após passar pelas lentes a luz é captada pelo sensor (CCD) da câmera(SZELISKI, 2011). A Modelagem mais simples de um modelo de câmera é o modelo *pinhole*, onde as coordenadas de um ponto em 3D no espaço $X = (X, Y, Z, 1)^T$ e sua correspondente projeção sobre o plano da imagem $x = (x, y, 1)^T$, ambos representados em coordenadas homogêneas, estão relacionados pela equação de projeção:

$$\lambda x = PX, \tag{3.1}$$

onde λ é um fator de escala desconhecido proporcional a profundidade de X em relação à câmera e P é a matriz 3×4 de projeção de câmera, que pode ser fatorada como:

$$P = K[R|t], \tag{3.2}$$

onde:

$$K = \begin{bmatrix} f & s & x_0 \\ 0 & f & y_0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{3.3}$$

A matriz de calibração de câmera K mapeia coordenadas métricas em coordenadas de imagem (pixels). K contém os parâmetros intrínsecos da câmera, onde f representa a distância focal da câmera, $[x_0, y_0]^\top$ são as coordenadas do ponto principal da câmera, que representa as coordenadas da imagem onde ocorre a intersecção do eixo ótico e o plano da imagem, s é referido como o fator de inclinação e diz respeito a formatos não retangulares de pixels. A matriz 3×4 de parâmetros externos $[R|t]$ representa a orientação e a posição da câmera. R é uma matriz de rotação e t é um vetor de translação (HARTLEY; ZISSERMAN, 2004).

Usualmente as câmeras digitais modernas fazem um pré-processamento para remover ruído e melhorar a qualidade das imagens adquiridas e depois armazenam as imagens am algum modelo de cor. Essencialmente um modelo de cor é uma especificação de um sistema de coordenadas tridimensionais e um subespaço dentro deste sistema onde cada cor é representada por um único ponto. Os modelos de cor mais conhecidos e utilizados em visão computacional são o RGB, YIQ e o HSI (GONZALEZ; WOODS, 2006).

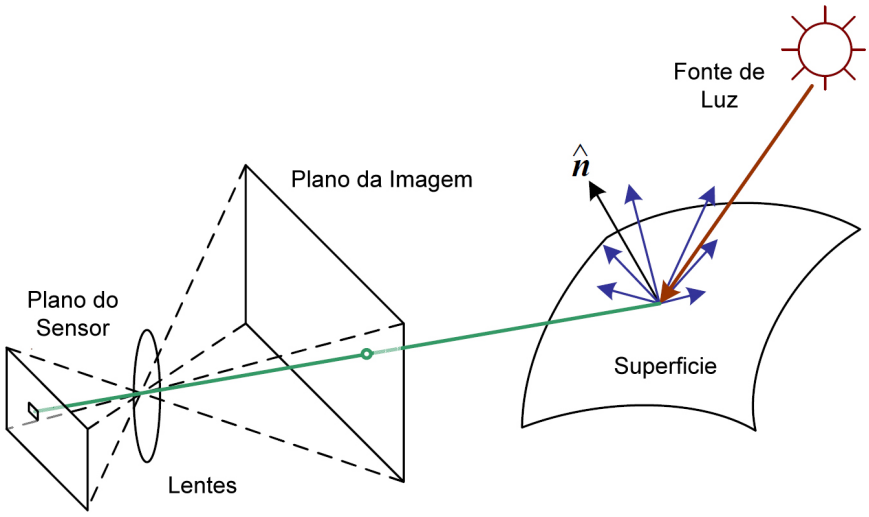


Figura 3.1: Simplificação de um modelo que demonstra a formação de uma imagem 2D(cores) (SZELISKI, 2011).

3.1.2 Formação de uma Imagem de Profundidade

Ao contrário de uma imagem colorida que informa a intensidade de cor dos objetos na cena, uma imagem de profundidade informa em cada pixel a distância entre à câmera e os objetos da cena. Existem muitas utilidades práticas em obter a profundidade de uma cena, nelas estão inclusas, maior facilidade em segmentar dos objetos da cena, remoção de fundo, cálculo do tamanho dos objetos, entre outras. Entretanto, recuperar a posição de pontos 3D na cena e estimar sua distância utilizando apenas restrições geométricas em uma única imagem não é possível. Como seres humanos, nós confiamos em nosso conhecimento semântico do mundo para realizar tal inferência, mas essa capacidade pode ser facilmente enganada(SZELISKI, 2011).

Atualmente existem várias técnicas propostas na literatura para recuperar a profundidade de uma cena, algumas delas são, estrutura pelo movimento, correspondência estéreo e luz estruturada. Em estrutura pelo movimento, tenta-se recuperar a profundidade da cena utilizando um conjunto de feições correspondentes adquiridas em diferentes posições de câmera utilizando uma triangulação para recuperar a profundidade. Em correspondência estéreo o processo é muito similar, porém em vez de utilizar uma única câmera e movimentá-la, utilizam-se duas câmeras paralelas devidamente calibradas. Já em luz estruturada projeta-se um padrão na cena e com as deformações deste padrão calculam-se as profundidades da cena(SZELISKI, 2011), o Kinect utiliza luz estruturada para estimar a profundidade da cena, a seguir será detalhado o funcionamento do Kinect.

3.1.2.1 Microsoft Kinect

O Kinect foi lançado em Novembro de 2010 como um sensor de movimentos para o vídeo game Xbox 360 da Microsoft e logo se transformou em um sucesso de vendas. O sucesso foi tanto que o Kinect entrou para o Guiness Book como o dispositivo eletrônico mais rapidamente vendido, depois de vender 8 milhões de unidades.

No início de 2012, a Microsoft lançou uma versão do Kinect para PC chamada Kinect for Windows. Esta nova versão é muito parecida com o modelo do XBox 360 mas com uma diferença, ele possui a opção de modo próximo(*near mode*), neste modo é possível captar distâncias de objetos mais próximos da câmera do Kinect. Juntamente com o lançamento do Kinect for Windows a Microsoft também anunciou o Kinect SDK[1] o qual contém os drivers necessários e bibliotecas para acessar o Kinect nas linguagens C++, C# e Visual Basic. Embora também seja possível acessar o Kinect do Xbox em um PC utilizando o SDK, este modelo não suporta o modo próximo. Existe também a biblioteca OpenNI [2] que dá suporte a vários sensores RGB-D em Windows, Linux e OSX.

O Hardware Kinect é composto por uma câmera de cores RGB com um resolução máxima de 1280×960 pixels, uma câmera de infravermelho com um resolução máxima de 640×480 pixels, um projetor de feixes de infravermelho, um array de 4 microfones distribuídos horizontalmente e um motor de movimento vertical para as câmeras. A Figura 3.2 demonstra como estão dispostos estes componentes no Kinect.

O ângulo de visão horizontal do Kinect é de 57 graus e o vertical 43 graus. O alcance de profundidade em modo próximo é de 400mm e em modo padrão é de 800mm. A Figura 3.3 demonstra um quadro colorido e um de profundidade adquiridos com o Kinect em modo próximo. Devido a natureza da luz infravermelha o Kinect funciona apenas em locais cobertos.

Para estimar a profundidade da cena o Kinect projeta um padrão estruturado de infravermelho utilizando o projetor de infravermelho, este padrão é captado pela câmera de infravermelho e as deformações do padrão projetado são utilizadas para estimar a profundidade de cada pixel utilizando uma triangulação(KHOSHELHAM, 2011). Os inventores (FREEDMAN et al., 2010) descrevem a medida da profundidade como um processo de triangulação. A fonte de lasers emite um único feixe que é dividido em múltiplos feixes por uma grade de difração para criar um padrão constante de manchas projetadas na cena. Este padrão é então capturado pela câmera de infravermelho e é correlacionado com um padrão de referência. O padrão de referência é obtido capturando um plano a uma distância conhecida do sensor, e é armazenado na memória do sensor. Quando o padrão de manchas é projetado em um objeto o qual a distância do sensor seja menor ou maior que

[1]Disponível para download em: http://www.microsoft.com/en-us/Kinectforwindows/
[2]Disponível para download em: http://www.openni.org/

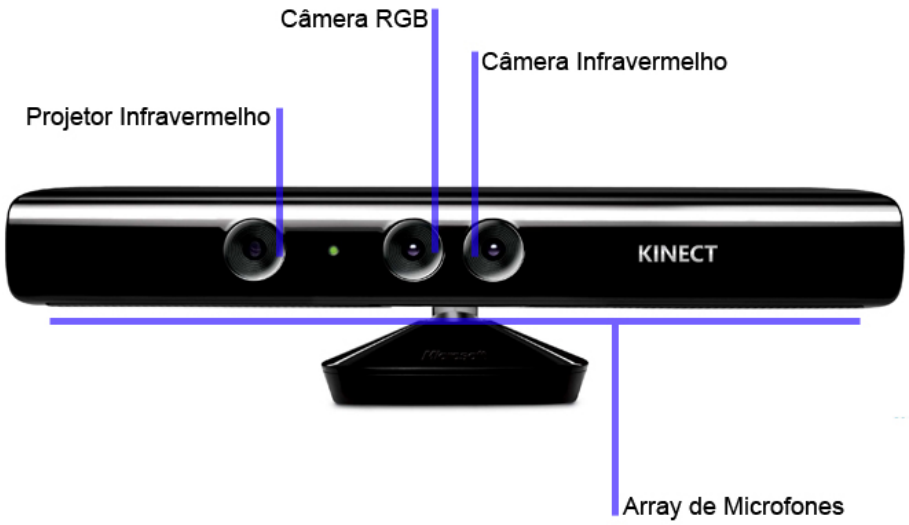


Figura 3.2: Esta imagem demonstra como estão dispostos os componentes do Kinect em seu invólucro.

Figura 3.3: Exemplo de um frame RGB(esquerda) e de um frame de profundidade(direita) capturados com o Kinect.

a do plano de referência, a posição das manchas na imagem de infravermelho irão ser deslocadas em direção da linha base entre o projetor de lasers e o centro de perspectiva da câmera de infravermelho. Estes deslocamentos são medidos para todas as manchas com um simples procedimento de correlação, o que gera uma imagem de disparidades. Para cada pixel a distância do sensor com a cena pode ser recuperada da imagem de disparidades. A figura 3.4 demonstra um frame capturado do padrão de manchas da câmera de infravermelho do Kinect (KHOSHELHAM, 2011).

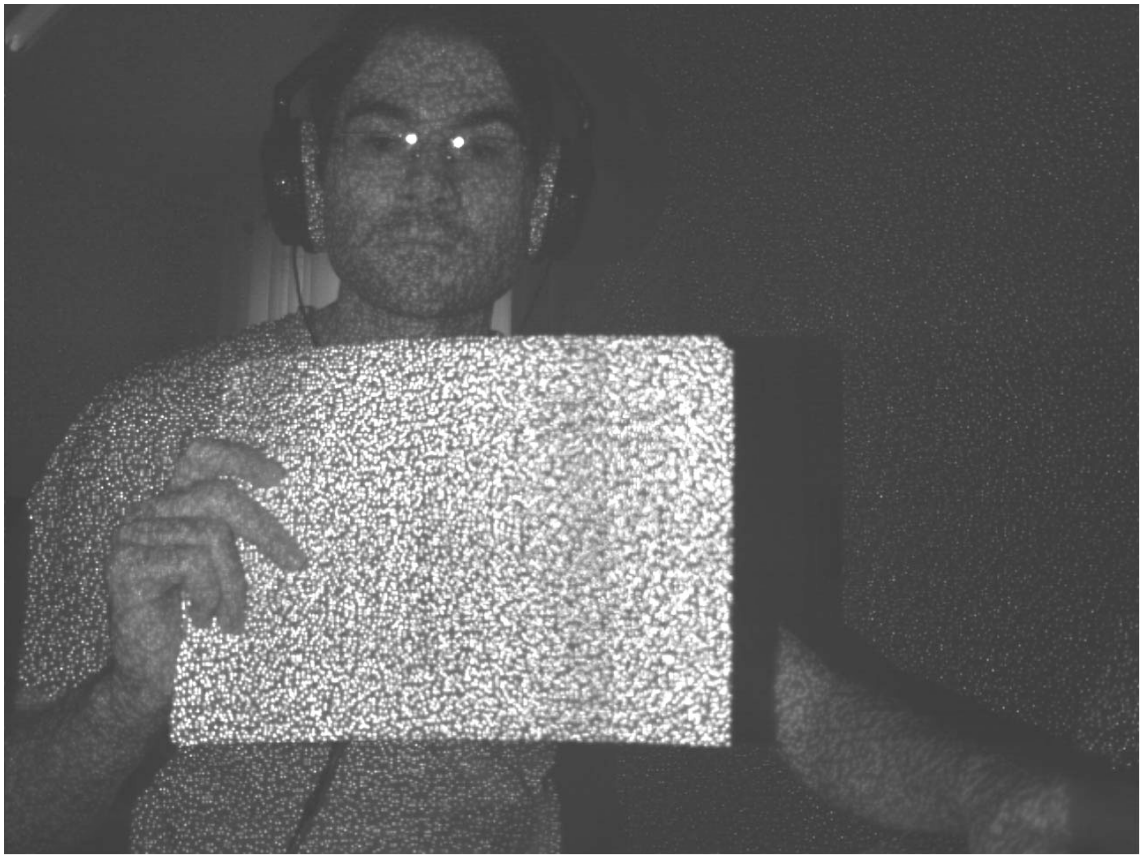


Figura 3.4: Padrão de manchas projetado pelo emissor de infravermelho do Kinect capturado pela câmera de infravermelho.

A Figura 3.5 ilustra a relação entre a distância de um ponto k de um objeto e o sensor em relação a um plano de referência e a disparidade d medida. Para expressar as coordenadas 3D dos pontos do objeto é considerado o centro de perspectiva da câmera de infravermelho como a origem do sistema de coordenadas. O eixo Z é ortogonal ao plano de imagem em relação ao objeto, o eixo X é perpendicular ao eixo Z na direção da linha de base b entre o centro ótico da câmera infravermelha e o projetor de lasers, e o eixo Y é ortogonal ao X e Z gerando um sistema de coordenadas da mão direita. Suponha que um objeto está no plano de referência a uma distância Z_o do sensor, e uma mancha sobre o objeto é capturada sobre o plano de imagem da câmera de infravermelho. Se o objeto é deslocado para mais perto(ou mais longe) do sensor, a posição da mancha capturada sobre o plano da imagem vai ser deslocada na direção do eixo X. Isto é medido no espaço de imagem como uma disparidade d correspondente a um ponto k no espaço do objeto. E a partir da semelhança de triângulos temos:

$$\frac{D}{b} = \frac{Z_o - Z_k}{Z_o}, \tag{3.4}$$

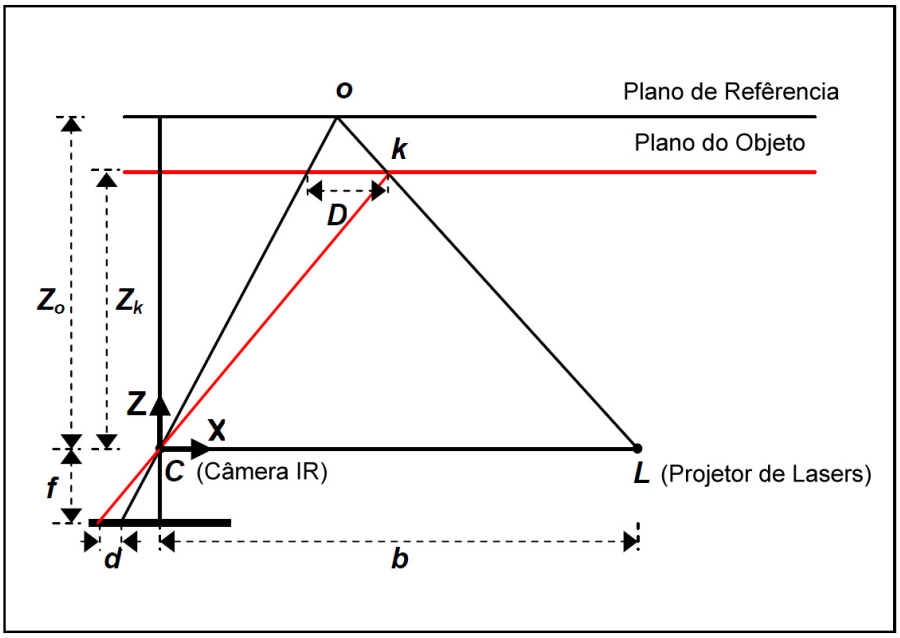


Figura 3.5: Representação esquemática da relação entre a profundidade e a disparidade (KHOSHELHAM, 2011).

e:

$$\frac{d}{f} = \frac{D}{Z_k}, \tag{3.5}$$

onde Z_k denota a distância(profundidade) do ponto k no espaço do objeto, b é o tamanho da linha base(distância entre à câmera de infravermelho até o projetor de lasers.), f é a distância focal da câmera de infravermelho, D é deslocamento do ponto k no espaço do objeto, e d é a disparidade observada no espaço da imagem. Substituindo oD de (3.5) em (3.4) e expressando Z_k em termos de outras variáveis:

$$Z_k = \frac{Z_o}{1 + \frac{Z_o}{fb}d} \tag{3.6}$$

A equação (3.6) é o modelo matemático básico para a derivação da profundidade a partir da disparidade observada onde os parâmetros constantes Z_o, f, e b podem ser determinados por calibração (KHOSHELHAM, 2011).

3.1.3 Filtragem, Realce e Suavização de Imagens

O principal objetivo das técnicas de realce de imagens é processar uma imagem de modo que a imagem resultante seja mais adequada que a imagem original para uma aplicação em específico. Desta afirmativa decorrem duas importantes conclusões (FILHO; NETO, 1999):

1. A interpretação de que o resultado é mais adequado, ou não, normalmente é subjetiva e depende de conhecimento prévio do observador a respeito das imagens analisadas.

2. As técnicas de realce de imagens são por natureza orientadas a um problema que se deseja resolver. Logo, não existem técnicas capazes de resolver 100% dos problemas que uma imagem digital possa apresentar, como também nem sempre uma

técnica que produz bons resultados para imagens biomédicas adquiridas através de um tomógrafo computadorizado apresentará desempenho satisfatório se aplicada a uma imagem contendo uma impressão digital, por exemplo.

Os métodos de filtragem de imagens são normalmente classificados em duas categorias: as técnicas de filtragem espacial e as técnicas de filtragem no domínio da freqüência. Os métodos que trabalham no domínio espacial operam diretamente sobre a matriz de pixels que é a imagem digitalizada, normalmente utilizando operações de convolução com máscaras. Os métodos que atuam no domínio da freqüência se baseiam na modificação da transformada de Fourier da imagem. A seguir são apresentados os fundamentos e alguns exemplos de técnicas de Filtragem, realce e suavização de imagens no domínio espacial.

3.1.3.1 Vizinhança de um Pixel e Convolução de Máscaras

Uma imagem digital pode ser definida pela função $f(x, y)$ discretizada tanto espacialmente quanto em amplitude. Portanto, uma imagem digital pode ser vista como uma matriz cujas linhas e colunas identificam um ponto na imagem, cujo valor corresponde ao nível de cinza da imagem naquele ponto. Para referir-se a um pixel em particular, serão utilizadas letras minúsculas, tais como p.

Um pixel p, de coordenadas (x, y), tem 4 vizinhos horizontais e verticais, cujas coordenadas são $(x + 1, y)$, $(x - 1, y)$, $(x, y + 1)$ e $(x, y - 1)$. Estes pixels formam a chamada "4-vizinhança"de p, que são chamados de $N_4(p)$. Os quatro vizinhos diagonais de p são os pixels de coordenadas $(x - 1, y - 1)$, $(x - 1, y + 1)$, $(x + 1, y - 1)$ e $(x + 1, y + 1)$, que constituem o conjunto $N_d(p)$. A "8-vizinhança"de p é definida como a união dos dois conjuntos:

$$N_8(p) = N_4(p) \cup N_d(p). \tag{3.7}$$

A Figura 3.6 demonstra estes tipos de vizinhança (GONZALEZ; WOODS, 2006).

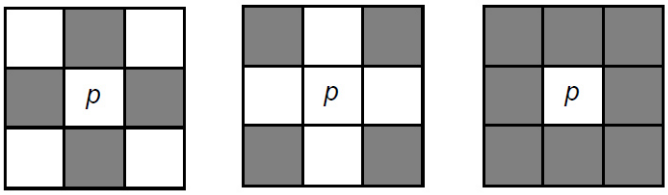


Figura 3.6: Conceitos de 4-vizinhança, vizinhança diagonal e 8-vizinhança (GONZALEZ; WOODS, 2006).

As técnicas de filtragem no domínio espacial são aquelas que atuam diretamente sobre a matriz de pixels que é a imagem digitalizada. Logo, as funções de processamento de imagens no domínio espacial podem ser expressas como:

$$g(x, y) = T[f(x, y)], \tag{3.8}$$

onde: $g(x, y)$ é a imagem depois de ser processada, f(x,y) é a imagem original e T é um operador aplicado em f, definido em uma certa vizinhança de (x,y). Além disso, o operador T pode também operar sobre um conjunto de imagens de entrada.

A vizinhança normalmente definida ao redor de (x, y) é a 8-vizinhança do pixel de referência, o que equivale a uma região 3×3 na qual o pixel central é o de referência, como indica a Figura 3.7. O centro dessa região ou sub-imagem é movido pixel a pixel,

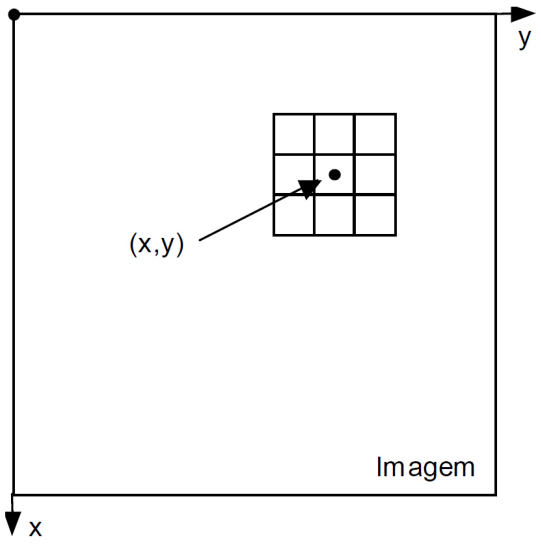


Figura 3.7: Exemplo de uma mascara de convolução sendo aplicada em um determinado pixel de uma imagem (GONZALEZ; WOODS, 2006).

iniciando no canto superior esquerdo da imagem e aplicando a cada localidade o operador T para calcular o valor de g naquele ponto.

Dada a sub-imagem de uma imagem:

Z_1	Z_2	Z_3
Z_4	Z_5	Z_6
Z_7	Z_8	Z_9

onde: $Z_1, \cdots, Z_9$ são os valores de tons de cinza de cada pixel. E dada uma máscara 3×3 de coeficientes genéricos $W_1, \cdots, W_9$:

W_1	W_2	W_3
W_4	W_5	W_6
W_7	W_8	W_9

A máscara acima percorrerá a imagem, desde o seu canto superior esquerdo até seu canto inferior direito. A cada posição relativa da máscara sobre a imagem, o pixel central da sub-imagem em questão será substituído, em uma matriz denominada 'imagem-destino', por um valor:

$$Z = \sum_{i=1}^{9} W_i.Z_i \tag{3.9}$$

As operações de convolução com máscaras são amplamente utilizadas no processamento de imagens. Uma seleção apropriada dos coeficientes $W_1, \cdots, W_9$ torna possível uma grande variedade de operações úteis, tais como redução de ruído, afinamento e detecção de características da imagem (FILHO; NETO, 1999).

3.1.3.2 Remoção de Ruído - Filtro da Média

O filtro da média basicamente calcula a média dos pixels na vizinhança de um pixel p e substitui o valor antigo de p pela média da vizinhança definida. A forma mais simples

de implementar um filtro com tais características é construir uma máscara 3×3 com todos seus coeficientes iguais a 1, dividindo o resultado da convolução por um fator de normalização, neste caso igual a 9. Um filtro com esta característica é denominado filtro da média. A Figura 3.8(a) demonstra a máscara resultante, enquanto as Figura 3.8(b) e 3.8(c) ilustram o mesmo conceito, aplicado a máscaras com dimensões maiores. Na escolha do tamanho da máscara deve-se ter em mente que quanto maior a máscara, maior o grau de borramento da imagem resultante. A figura 3.9(b), 3.9(c) e 3.9(d) demonstra alguns exemplos de máscaras de filtragem pela média de diferentes dimensões aplicadas na Figura 3.9(a). As figuras Figura 3.9(f) e 3.9(h) mostram exemplos da aplicação do filtro da média para remoção de ruídos em imagens em escalas de cinza.

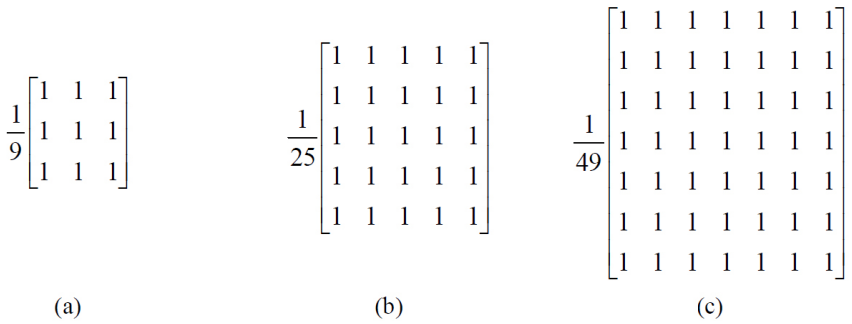


Figura 3.8: Exemplos de máscaras do filtro da média espacial (FILHO; NETO, 1999).

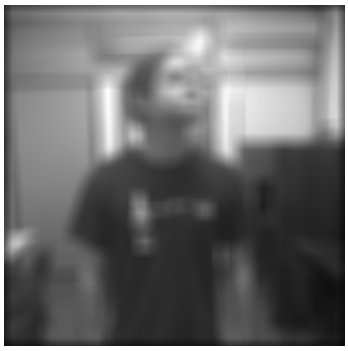
3.1.3.3 Remoção de Ruído - Filtro da Mediana

Uma das principais limitações do filtro da média é sua incapacidade de preservar bordas e detalhes finos na imagem e isto é ruim quando o objetivo é a remoção de ruído da imagem. Para contornar este problema, pode-se utilizar o filtro da mediana. Nesta técnica, o nível de cinza do pixel central da janela é substituído pela mediana dos pixels situados em sua vizinhança. Este método não-linear apresenta desempenho particularmente bom em situações nas quais a imagem é contaminada por ruído impulsivo (sal-e-pimenta), como pode-se verificar na Figura 3.10(f). Já para situações em que o ruído é do tipo gaussiano, seu desempenho é apenas satisfatório, comparável ao do filtro da média como é mostrado na Figura 3.10(h). A mediana m de um conjunto de n elementos é o valor tal que metade dos n elementos do conjunto situem-se abaixo de m e a outra metade acima de m. Quando n é ímpar, a mediana é o próprio elemento central do conjunto ordenado. Nos casos em que n é par, a mediana é calculada pela média aritmética dos dois elementos mais próximos do centro (GONZALEZ; WOODS, 2006).

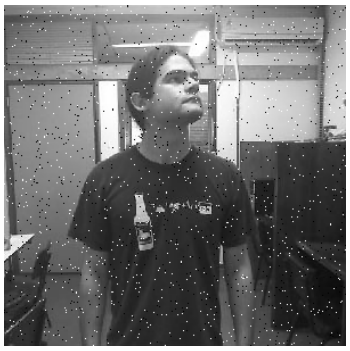
(a) Imagem original

(b) Máscara 3×3

(c) Máscara 5×5



(d) Máscara 15×15



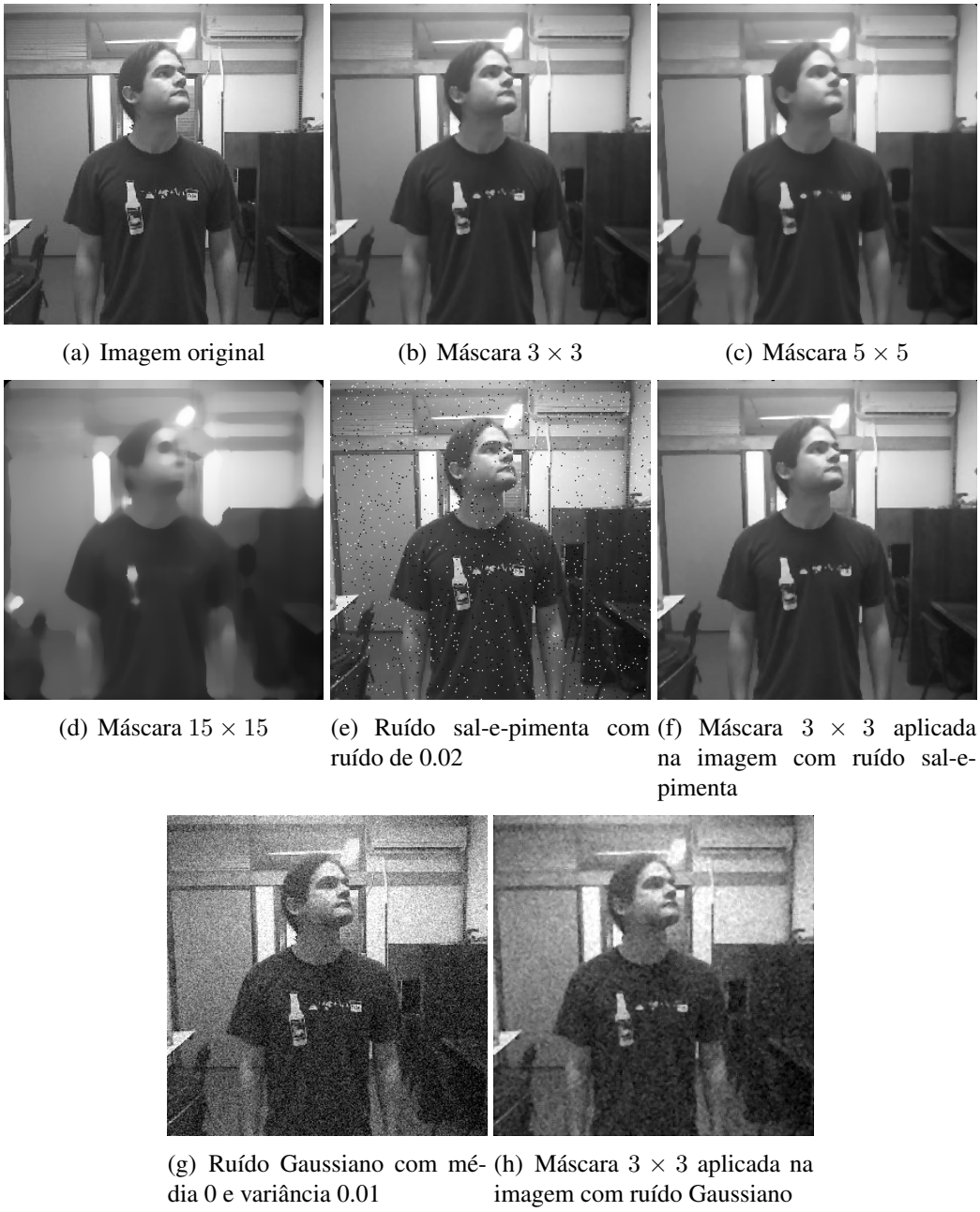
(e) Ruído sal-e-pimenta com ruído de 0.02

(f) Máscara 3×3 aplicada na imágem com ruído sal-e-pimenta

(g) Ruído Gaussiano com média 0 e variância 0.01

(h) Máscara 3×3 aplicada na imagem com ruído Gaussiano

Figura 3.9: Demonstração dos resultados de um filtro da média em uma imagem 2D em escalas de cinza. As sub-figuras (b),(c) e (d) são resultados aplicando a máscara na sub-figura (b). Já a sub-figura (f), é o resultado aplicando a máscara em (e). E a sub-figura (h), é o resultado aplicando a máscara em (g).



(a) Imagem original (b) Máscara 3×3 (c) Máscara 5×5

(d) Máscara 15×15 (e) Ruído sal-e-pimenta com ruído de 0.02 (f) Máscara 3×3 aplicada na imagem com ruído sal-e-pimenta

(g) Ruído Gaussiano com média 0 e variância 0.01 (h) Máscara 3×3 aplicada na imagem com ruído Gaussiano

Figura 3.10: Demonstração dos resultados de um filtro da mediana em uma imagem 2D em escalas de cinza. As sub-figuras (b),(c) e (d) são resultados aplicando a máscara na sub-figura (b). Já a sub-figura (f), é o resultado aplicando a máscara em (e). E a sub-figura (h), é o resultado aplicando a máscara em (g).

3.1.3.4 Detecção de Bordas

Define-se uma borda (*edge*) como sendo a fronteira entre duas regiões cujos níveis de cinza predominantes são razoavelmente diferentes. Pratt (PRATT, 1991) define uma borda de luminosidade como uma descontinuidade na luminosidade de uma imagem. Analogamente, pode-se definir borda de textura ou borda de cor, em imagens onde as informações de textura ou cor, respectivamente, são as mais importantes. Serão abordadas somente de bordas de luminosidade, às quais serão denominadas simplesmente bordas.

Para a detecção e realce de bordas, aplicam-se habitualmente filtros espaciais lineares de dois tipos: (a) baseados no gradiente da função de luminosidade, $I(x, y)$, da imagem,

e (b) baseados no laplaciano de $I(x, y)$.

Tanto o gradiente quanto o laplaciano costumam ser aproximados por máscaras de convolução ou operadores 3×3. Exemplos destas máscaras são os operadores de Sobel e Prewitt, mostrados na Figura 3.11.

	Vertical	Horizontal
Sobel	$\frac{1}{4}\begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$	$\frac{1}{4}\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$
Prewitt	$\frac{1}{3}\begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}$	$\frac{1}{3}\begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$

Figura 3.11: Máscaras de convolução de Sobel e Prewitt (FILHO; NETO, 1999).

As Figura 3.12(b) e 3.12(c) mostram os resultados da aplicação dos operadores de Prewitt e Sobel a uma imagem monocromática. Os resultados obtidos com a aplicação dos operadores verticais e horizontais foram combinados por meio de uma operação lógica OR.

O operador laplaciano é definido como:

$$\bigtriangledown^2 f(x, y) = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}, \tag{3.10}$$

e que pode ser aproximado pelas máscara:

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}. \tag{3.11}$$

A figura 3.12(c) mostra os resultados obtidos com o operador laplaciano, embora o laplaciano seja insensível à rotação, e portanto capaz de realçar ou detectar bordas em qualquer direção, seu uso é restrito devido à sua grande suscetibilidade a ruído (FILHO; NETO, 1999)..

3.1.4 *Active Appearance Models*

O Modelo de Aparência Ativa (AAM - Active Appearance Model) é um modelo estatístico baseado no formato dos objetos. O AAM é composto por um modelo estatístico do formato e da aparência em tons de cinza do objeto de interesse que pode generalizar quase qualquer objeto válido. Casar uma imagem implica em encontrar os parâmetros do modelo que minimizem a diferença entre a imagem e o modelo sintetizado projetado na imagem (COOTES; EDWARDS; TAYLOR, 1998).

Assumindo que uma forma s é descrita por N feições de pontos, $s = [x_1, y_1, x_2, y_2, \cdots, x_N, y_N]$ na imagem, uma forma é representada por um AAM com uma forma média s_0 mais uma combinação linear de n bases da forma $\{s_i\}$:

$$s(\mathbf{p}) = s_0 + \sum_{i=1}^{n} p_i s_i, \tag{3.12}$$

(a) Imagem original (b) Máscara de Prewitt

(c) Máscara de Sobel (d) Laplaciano

Figura 3.12: Demonstração de alguns dos métodos de detecção de bordas, (a) imagem original, (b) Convolução com o operador de Prewitt, (c) Convolução com o operador de Sobel e (d) Convolução com o operador Laplaciano.

onde $\mathbf{p} = [p_1, p_2, \cdots, p_n]$ são os parâmetros da forma. Normalmente a forma média s_0 e as bases da forma $\{s_i\}$ são aprendidas aplicando PCA nas formas de treinamento. A Figura 3.13(a) demonstra alguns exemplos das bases de forma. Para considerar a transformação global de uma forma, as bases das formas $\{s_i\}$ são expandidas para incluir quatro bases adicionais representando translação, escala e rotação (ZHOU et al., 2010).

A aparência A do AAM é definida como a parte da imagem associada pela forma média S_0. Similar à forma, a aparência A é representada pela aparência média A_0 mais uma combinação linear de m bases de aparência $\{A_i\}$:

$$A = A_0 + \sum_{i_m}^{m} \lambda_i A_i, \tag{3.13}$$

onde os coeficientes $\{\lambda_i\}$ são os parâmetros de aparência. A aparência média A_0 e as bases de aparência A_i são aprendidas aplicando PCA nas imagens de treinamento com forma normalizada, A Figura 3.13(b) demonstra alguns exemplos das bases de aparência(ZHOU et al., 2010).

Para localizar uma forma em uma imagem observada I, AAM tenta achar um conjunto ótimo de parâmetros de forma $\mathbf{p}$ e de parâmetros de aparência λ os quais minimizem a diferença entre a aparência deformada $I(\mathbf{W}(\mathbf{p}))$ e a aparência sintetizada A_λ:

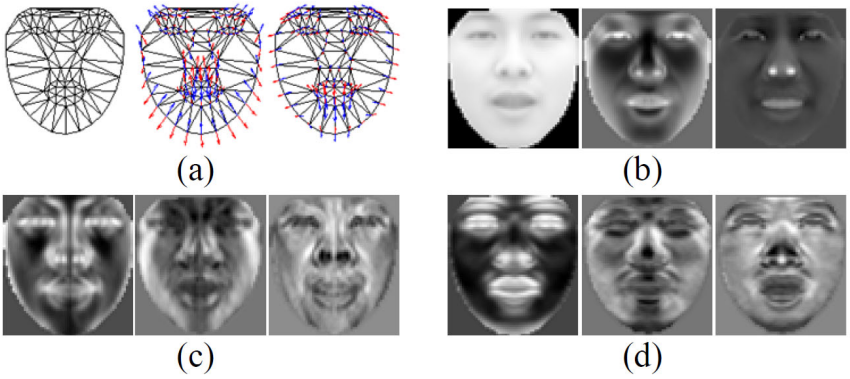


Figura 3.13: Modelo AAM,(a) forma média s_0 e as duas primeiras bases de forma aprendidas com PCA. (b) até (d) são a aparência média e as duas primeiras bases de aparência (ZHOU et al., 2010).

$$\begin{aligned} E_a(\mathbf{p},\lambda) &= \|A_\lambda - I(\mathbf{W}(\mathbf{p}))\|_2 \\ &= \sum_{x \epsilon s_0} [A_0(x) + \sum_{i=1}^{m} \lambda_i A_i(x) - I(\mathbf{W}(x;\mathbf{p}))]^2 \end{aligned} \quad , \tag{3.14}$$

onde $\mathbf{W}(x;\mathbf{p})$ é uma função de deformação definida para mapear cada pixel x nas coordenadas do modelo para as coordenadas correspondentes nos pontos da imagem.

Normalmente $\mathbf{W}(x;\mathbf{p})$ é uma deformação afim por partes definida pelo par de formas s_0 e $s(\mathbf{p})$: para cada triângulo(vide Figura 3.13(a)) em s_0 existe um triângulo correspondente em $s(\mathbf{p})$ e cada par de triângulos define-se uma deformação afim. A função de custo (3.14) pode ser eficientemente minimizada utilizando a inversa da técnica compositional parameter update (MATTHEWS; BAKER, 2004).

3.1.5 Nuvens de Pontos

Nuvens de Pontos(Point Clouds) representam um formato básico de entrada para alguns sistemas de percepção 3D, e prevêem uma representação discreta porém significativa do mundo ao redor. Sem qualquer perda de generalidade, as coordenadas $\{x_i, y_i, z_i\}$ de qualquer ponto $p_i \in P$ são dados com respeito a um sistema de coordenadas fixo, usualmente tendo sua origem no dispositivo utilizado para aquisição da informação. Isto significa que cada ponto p_i representa a distância sobre os três eixos de coordenadas definidos a partir do ponto de vista do dispositivo até o ponto da superfície que foi amostrado (RUSU, 2009).

3.1.6 *Iterative Closest Point*

O algoritimo Iterative Closest Point(ICP) calcula iterativamente a transformação T ótima(i.e. rotação e translação) entre duas nuvens de pontos. O ICP é basicamente composto de três passos principais, isto é (i) busca de correspondências entre as nuvens de pontos, (ii) busca pela transformação rígida dados os pontos correspondentes, (iii) Aplica a transformação e repete o processo. Os passos são repetidos ciclicamente até obter convergência, i.e. a transformação calculada não muda significativamente (BESL; MCKAY., 1992).

O ICP começa com uma estimativa inicial para a transformação T entre as duas nuvens de pontos. Assumindo que se quer alinhar a nuvem $B \in \mathbb{R}^{3\times n}$ com a nuvem $A \in \mathbb{R}^{3\times k}$. A correspondência consiste em procurar para cada ponto de $T.B$ o ponto mais próximo em A. Existem inúmeros métodos para procurar o vizinho mais próximo como, busca exaustiva, e *kd-trees*. Procurar uma transformação entre pontos correspondentes normalmente envolve minimizar a soma ponto a ponto das distâncias euclidianas. Considerando o caso de informação 3D, existem algumas alternativas de funções de custo. (CHEN; MEDIONI, 1991) propõem uma distância ponto a ponto que minimiza o erro baseado na normal da superfície de uma leitura.

3.2 Reconhecimento de Padrões

A área de reconhecimento de padrões tem como foco a descoberta automática de padrões em dados através do uso de algoritmos de computadores e com o uso destes padrões realizar ações como classificar os dados em diferentes categorias ou classes. Dado um padrão representado em um vetor $\mathbf{x} = (x_1, x_2, \cdots, x_n)^T$, onde cada componente x_i são medidas ou características de um dado objeto. O objetivo é relacionar este vetor $\mathbf{x}$ com uma das C classes ou grupos, denotados por $w_1, w_2, \cdots, w_C$. Um vetor $\mathbf{x}$ pertence a uma classe w_i caso sua variável classificatória c seja $c = i$ onde $i \in \{1, 2, \cdots, C\}$ (BISHOP, 2006).

Exemplos de aplicações de reconhecimento de padrões são: reconhecimento de fala, reconhecimento de caracteres, reconhecimento de objetos, previsão do tempo, diagnóstico de doenças, reconhecimento de digitais, seqüenciamento do DNA, entre outras (RICHARD O. DUDA PETER E. HART, 2000)

Existem dois grupos principais de aprendizado de padrões: aprendizado supervisionado e aprendizado não supervisionado. No aprendizado supervisionado, um professor fornece um rótulo para cada classe ou custo para cada padrão no conjunto de treinamento, e busca-se reduzir a soma dos custos para estes padrões. Já no aprendizado não supervisionado não existe um professor explícito, e o sistema busca por aglomeramentos(*clusters*) ou "agrupamentos naturais"dos padrões de entrada (RICHARD O. DUDA PETER E. HART, 2000).

Neste capítulo, são abordados alguns dos conceitos básicos das técnicas de reconhecimento de padrões. Dentre os assuntos tratados, serão vistas técnicas de redução de dimensionalidade e *clustering*.

3.2.1 Classificador de Bayes - Regra de Decisão de Bayes

A teoria de decisão Bayesiana é uma abordagem estatística fundamental para o problema de classificação de padrões (RICHARD O. DUDA PETER E. HART, 2000). O classificador Bayesiano ingênuo, chamado assim por assumir que as características são independentes entre si, apresenta resultados bastante competitivos em relação aos outros classificadores (FRIEDMAN; GEIGER; GOLDSZMIDT, 1997).

Considerando $p(\omega_i|x)$ como sendo a probabilidade de um dado padrão $x \in \mathbb{R}^n$ pertencer à classe $\omega_i, i = 1, 2, ..., c$, que pode ser definida pelo teorema de Bayes:

$$p(\omega_i|x) = \frac{p(x|\omega_i)p(\omega_i)}{p(x)}. \tag{3.15}$$

A fórmula de Bayes indica que, ao observar os valores de x, é possível converter a probabilidade a priori $p(\omega_i)$ para uma probabilidade a posteriori $p(\omega_i|x)$(probabilidade

de uma amostra pertencer a classe ω_j dado o vetor de características x). $p(x|\omega_i)$ é a probabilidade de ω_i com respeito a x (um termo escolhido para indicar que, considerando o restante constante, a categoria ω_i para qual a probabilidade $p(x|\omega_i)$ for maior é a mais provável de ser a categoria correta). O produto da probabilidade e probabilidade a priori é o mais importante para se determinar a probabilidade a posteriori, uma vez que o fator $p(x)$ pode ser visto como apenas um fator para garantir que a soma das probabilidades a posteriori seja igual a 1 (RICHARD O. DUDA PETER E. HART, 2000).

Um classificador Bayesiano decide se uma amostra x pertence à uma classe ω_j se:

$$p(\omega_i|x) > p(\omega_j|x), i, j = 1, 2, \cdots, c, i \neq j, \tag{3.16}$$

que pode ser reescrita da seguinte forma:

$$p(x|\omega_i)p(\omega_i) > p(x|\omega_j)p(\omega_j), i, j = 1, 2, \cdots, c, i \neq j. \tag{3.17}$$

Os valores da probabilidade de $p(\omega_i)$ podem ser facilmente obtidos através do cálculo do histograma de classes, por exemplo. No entanto, o problema mais complexo é calcular a função de densidade de probabilidade $p(x|\omega_i)$, uma vez que estão disponíveis apenas as informações dos conjuntos de padrões e seus respectivos rótulos. Uma função bastante utilizada para modelar tal problema é a Gaussiana ou normal. Dessa forma, assume-se que as funções de densidade de probabilidade são gaussianas e que é possível estimar seus parâmetros através das amostras da base de dados.

A densidade gaussiana de n dimensões do padrão da classe ω_i pode ser descrita como:

$$p(x|\omega_i) = \frac{1}{(2\pi)^{n/2}\left|C_i\right|^{1/2}} exp\left[-\frac{1}{2}(x-\mu_i)^T C_i^{-1}(x-\mu_i)\right], \tag{3.18}$$

onde μ e Ci representam, respectivamente, a média e a matriz de covariância da classe wi. Tais parâmetros são obtidos da seguinte forma:

$$\mu_i = \frac{1}{N_i}\sum_{x\in\omega_i} x, \tag{3.19}$$

$$C_i = \frac{1}{N_i}\sum_{X\in\omega_i}(xx^T - \mu_i\mu_i^T), \tag{3.20}$$

onde N_i representa o número de amostras da classe w_i.

3.2.2 K-means

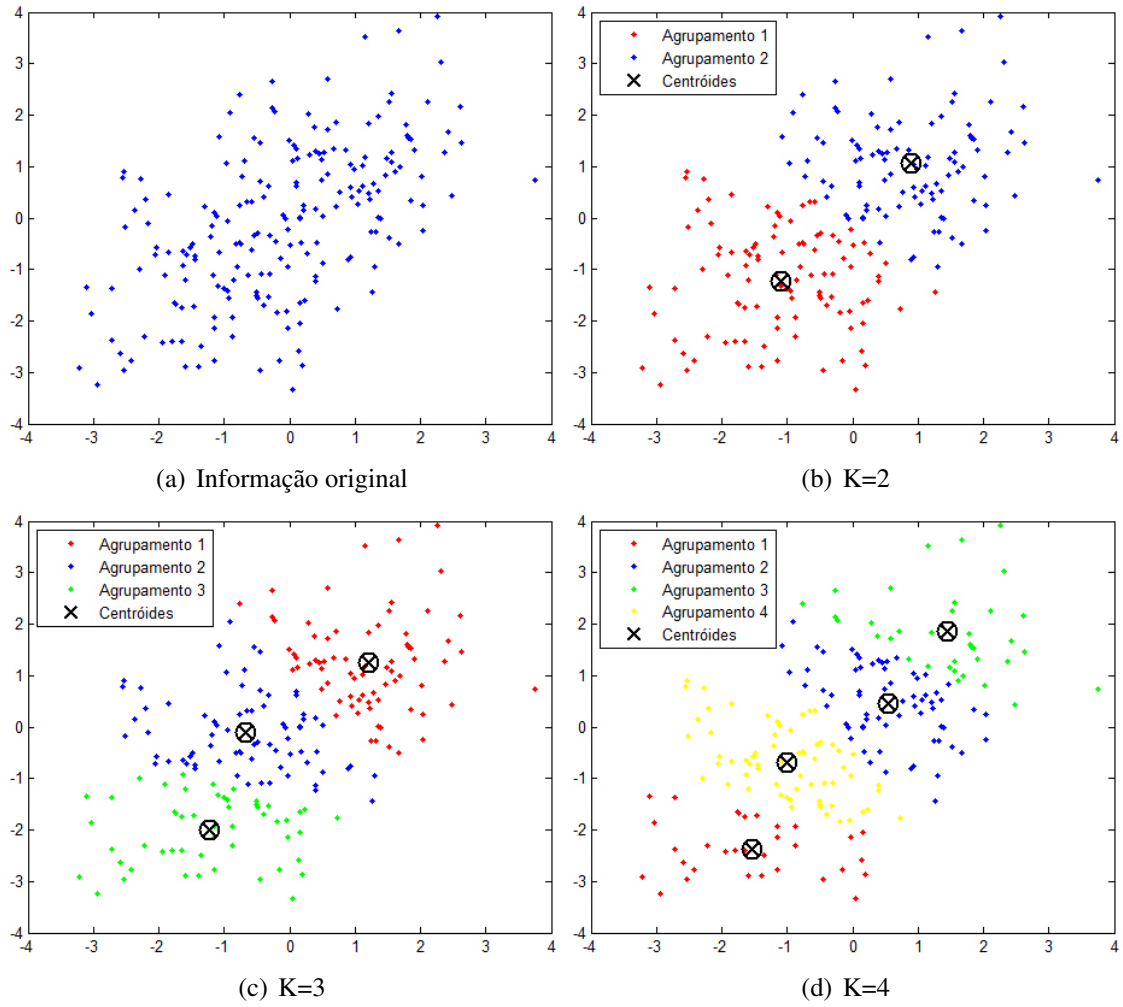
O K-means é um método de classificação não supervisionado, onde se considera o problema de encontrar grupos ou agrupamentos(clusters) de pontos de dados em um espaço multidimensional. Dado um conjunto de pontos $x_1, x_2, \cdots, x_N$ de N observações dimensional. O objetivo é particionar os dados em um número de K agrupamentos, onde K é um número dado a priori. Intuitivamente pode-se pensar em um aglomerado como um grupo de pontos comprimidos onde a distância entre estes pontos é pequena comparada com as distâncias dos pontos fora deste aglomerado. Podem-se formalizar esta idéia introduzindo o conjunto de vetores D-dimensional μ_k, onde $k \in \{1, 2, \cdots, K\}$, em que μ_k é um protótipo associado com o $k - esimo$ aglomerado. Como pode-se notar rapidamente, pode-se pensar em μ_k como sendo o centróide de cada aglomerado. O objetivo é

então associar os pontos dados aos devidos agrupamentos, assim como encontrar o conjunto de vetores μ_k, de tal forma que a soma dos quadrados das distâncias de cada ponto com o vetor μ_k mais próximo seja mínima (BISHOP, 2006). A Figura 3.14 demonstra alguns exemplos de K e seus resultados utilizando um conjunto de pontos.

Para cada ponto x_n, é associado um conjunto binário indicador r_{nk}, onde $K = 1, 2, \cdots, K$ que descreve qual dos K agrupamentos está associado ao ponto x_n, assim o ponto x_n é associado ao aglomerado K de tal forma que $r_{nk} = 1$, e $r_{nj} = 0$ para $j \neq k$. Isto é conhecido como o esquema de código $1 - de - K$. Com isto pode-se definir uma função objetiva,

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \left\| x_n - \mu_k \right\|^2, \tag{3.21}$$

a qual representa a soma dos quadrados das distâncias de cada ponto para seu vetor associado μ_k. O objetivo é encontrar valores para $\{r_{nk}\}$ e $\{\mu_k\}$ que minimizem J. Isto pode ser feito utilizando um processo iterativo (BISHOP, 2006).



(a) Informação original (b) K=2

(c) K=3 (d) K=4

Figura 3.14: Exemplos de agrupamentos encontrados utilizando o K-means variando os valores de K.

3.2.3 *Principal Component Analysis* (eigenfaces)

Principal Component Analysis, ou PCA, é uma técnica que é amplamente usada em aplicações como redução de dimensionalidade, compressão de imagens com perdas, extração de feições e visualização de informação. A PCA é também conhecida como a transformada de Karhunen-Loève. PCA pode ser definida como uma projeção ortogonal dos dados em um espaço linear de dimensão reduzida, conhecido como subespaço principal, para isso a variância dos dados projetados é maximizada (BISHOP, 2006).

Considerando um conjunto de N dados observados $\{X_n\}$ onde $n \in \{1, 2, \cdots, N\}$, e x_n tem dimensionalidade D. O objetivo é projetar os dados em um espaço com dimensionalidade $M < D$ enquanto maximiza-se a variância dos dados projetados (BISHOP, 2006). O novo vetor de feições $y_k \in R^M$ é definido pela seguinte transformação linear:

$$y_k = W^T x_k, \qquad k = 1, 2, \cdots, M, \tag{3.22}$$

onde $W \in R^{N \times M}$ é uma matriz com colunas ortonormais.

Se a matriz de covariância dos dados for definida como

$$S = \sum_{i=1}^{N} (x_i - \mu)(x_i - \mu)^T, \tag{3.23}$$

onde μ é a média de X que é definida por

$$\mu = \frac{1}{N} \sum_{i=1}^{N} x_i. \tag{3.24}$$

Ao aplicar a transformação linear W^T, o espalhamento dos vetores de características transformados $\{y_1, y_2, \cdots y_N\}$ é$W^T SW$.

No PCA, a projeção ótima W_{opt} é escolhida para maximizar o determinante da matriz total de espalhamento dos exemplos projetados. W_{opt} é dada por:

$$W_{opt} = \arg\max_{W} \left| W^T SW \right| = [w_1 w_2 \cdots w_m] \tag{3.25}$$

onde $\{w_i i = 1, 2, \cdots, m\}$ é o conjunto de autovetores de n dimensões de S correspondendo aos m maiores autovalores (RICHARD O. DUDA PETER E. HART, 2000).

A escolha do número de autovetores depende da representatividade desejada. A representatividade é obtida por meio da soma dos autovalores em ordem decrescente, sendo que para se formar o novo espaço de características deve-se utilizar os autovetores equivalentes. A Figura 3.15 demonstra um exemplo 2D de projeção em um espaço 1D usando PCA.

3.2.3.1 Eigenfaces

O método eigenfaces(TURK; PENTLAND, 1991), foi o primeiro método de reconhecimento facial que utiliza PCA para encontrar as componentes principais de uma face e reduzir a dimensionalidade com a matriz de projeção W encontrada. O eigenfaces representa cada dimensão do vetor X como sendo um pixel da imagem da face.

Após reduzir a dimensionalidade de uma imagem de face, é necessário definir uma métrica que informe quão diferente uma face é de outra. Uma métrica possível é a distância Euclidiana(RICHARD O. DUDA PETER E. HART, 2000), dada por

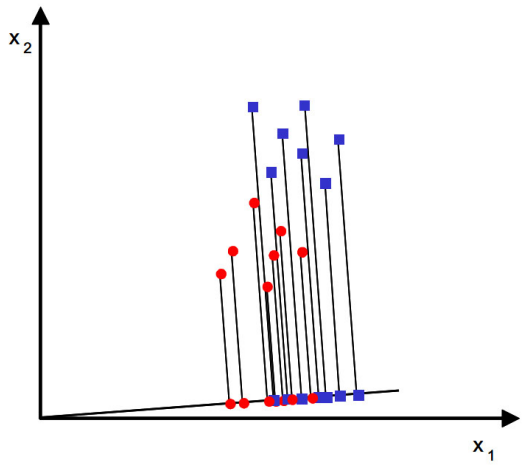


Figura 3.15: Exemplo de uma projeção PCA em um espaço de dimensão menor utilizando dados sintéticos.

$$D(A,B) = \sqrt{\sum_{i=1}^{d}(a_i - b_i)^2} \tag{3.26}$$

onde D é a distância Euclidiana, A e B são os vetores entre os quais quer se medir a distância e d é o número de dimensões dos vetores a e b.

3.2.4 *Linear Discriminant Analysis - Fisher Linear Discriminant* (fisherfaces)

PCA é uma técnica não-supervisionada e, como tal, não inclui informações do rótulo dos dados. A *Linear Discriminant Analysis* (LDA) e o discriminante linear de Fisher(*Fisher Linear Discriminant* FLD) relacionado são métodos utilizados nas estatísticas, reconhecimento de padrões e aprendizado de máquina para encontrar uma combinação linear de recursos que caracterize ou separe duas ou mais classes de objetos ou eventos. A combinação resultante pode ser utilizada como um classificador linear, ou, mais geralmente, para a redução de dimensionalidade antes da classificação posterior. O FLD é um exemplo de método específico de classe, no sentido que tenta "modelar"o espalhamento dos dados de forma que a classificação se torne mais confiável. Esse método seleciona a transformação linear W de forma que a razão entre o espalhamento inter-classes e o intra-classes seja maximizado(BELHUMEUR; HESPANHA; KRIEGMAN, 1997). O espalhamento inter-classes é definido como:

$$S_B = \sum_{i=1}^{C} N_i(\mu_i - \mu)(\mu_i - \mu)^T, \tag{3.27}$$

e o espalhamento intra-classes é definido como:

$$S_W = \sum_{i=1}^{C} \sum_{x_k \in X_i} (x_k - \mu_i)(x_k - \mu_i)^T, \tag{3.28}$$

onde μ é a vetor médio entre todas as classes, μ_i é o vetor médio da classe X_i e N_i é o número de exemplos da classe X_i.

Se S_W for uma matriz não singular, a projeção ótima W_{opt} é escolhida como a matriz com colunas ortonormais que maximiza a taxa do determinante da matriz de espalhamento entre classes dos exemplos projetados e também que minimiza a taxa do determinante da matriz de espalhamento intra-classe dos exemplos projetados.

$$W_{opt} = \arg\max_{W} \frac{|W^T S_B W|}{|W^T S_W W|} = [w_1 w_2 \cdots w_m] \tag{3.29}$$

A Figura 3.16 demonstra um exemplo 2D de projeção em um espaço 1D usando LDA, como pode-se observar a LDA separa melhor as classes comparado com a PCA(ver Figura 3.15).

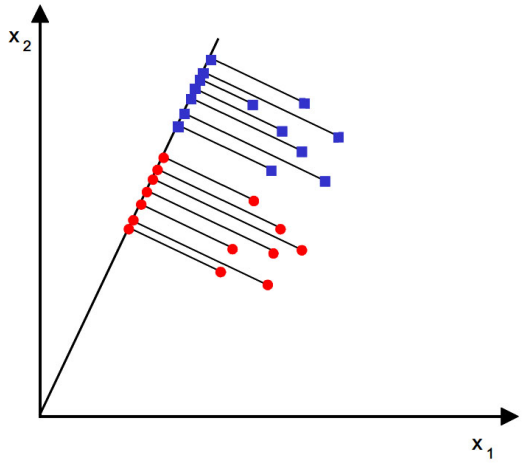


Figura 3.16: Exemplo de uma projeção LDA em um espaço de dimensão menor utilizando dados sintéticos.

3.2.4.1 Fisherfaces

Fisherfaces (BELHUMEUR; HESPANHA; KRIEGMAN, 1997) é outro método para reconhecimento facial de pessoas por meio de imagens. De forma similar ao eigenfaces, a técnica fisherfaces também se baseia na redução de dimensionalidade do espaço de características. A projeção ótima neste caso é obtida por meio da análise discriminante linear de Fisher(FLD - Fisher's Linear Discriminant). O fisherfaces tem um resultado melhor que o eigenfaces quando há variação de iluminação e expressões, isso se deve ao fato de que o fisherfaces encontra a projeção W de forma que a razão entre o espalhamento inter-classes e o intra-classes seja maximizada.

4 RECONHECIMENTO FACIAL TOLERANTE À VARIAÇÃO DE POSE UTILIZANDO UMA CÂMERA RGB-D DE BAIXO CUSTO

4.1 Visão Geral

Este trabalho tem como objetivo principal propor um método de reconhecimento facial que seja tolerante à variação de poses. O método de reconhecimento facial Fisherfaces (BELHUMEUR; HESPANHA; KRIEGMAN, 1997) tem ótimos resultados na classificação de faces quando não há variação de pose. Entretanto quando há uma grande variação de poses o Fisherfaces tem uma queda significativa nos resultados de classificação. Para lidar com este problema propõem-se a utilização da informação de pose de cada face de treinamento para separar o problema de classificação de faces em K sub-problemas de classificação baseados na informação de pose em comum das faces de treinamento. Para realizar a separação de poses o método utiliza o K-means na informação de pose das faces utilizadas para treinamento. Uma vez que foram encontrados os K agrupamentos de pose, o método seleciona as Ψ faces mais próximas de cada agrupamento e com a imagem das Ψ faces selecionadas treina um modelo do Fisherfaces para cada um dos K agrupamentos encontrados. O método proposto foi denominado K-Fisherfaces, pois dado um conjunto de faces de treinamento o método treina K modelos de faces baseados no Fisherfaces, onde cada modelo idealmente corresponde a um agrupamento de pose relativamente diferente.

Para capturar as imagens de faces foi utilizado um Kinect (câmera RGB-D), a vantagem em utilizar o Kinect em vez de uma câmera comum é que o Kinect além de fornecer uma imagem em cores fornece também uma imagem de profundidades que indica a distância entre à câmera e os objetos na cena. A informação de cor fornecida pelo Kinect é utilizada para localizar faces nas imagens coloridas fornecidas pelo Kinect. Com a informação do detector de faces é proposto um novo método de segmentação de faces nas imagens a cores e de profundidades. A face segmentada é então normalizada para um tamanho fixo e depois convertida para escalas de cinza. Esta nova imagem é então utilizada para treinar o modelo de faces proposto que é depois utilizado para o reconhecimento de novas faces de teste. Para o método proposto funcionar ainda é necessário estimar a pose da cabeça em relação à câmera, embora existam vários métodos de estimativa de pose de cabeça 2D estes métodos não funcionam em tempo real e/ou não tem uma boa precisão (FANELLI et al., 2011), portanto optou-se em utilizar a informação de profundidade fornecida pelo Kinect para estimar a pose da cabeça.

Este capítulo está estruturado da seguinte maneira: a seção 4.2 detalha como o método pré-processa as imagens que serão utilizadas posteriormente nas etapas de treinamento e

classificação. A seção 4.3 explica como o método proposto aprende um modelo de faces tolerante à variação de poses a partir de um conjunto de imagens de treinamento e como este modelo pode ser utilizado para classificar novas faces.

4.2 Pré-processamento das Imagens

Esta seção explica como as imagens são processadas antes de serem utilizadas para treinamento ou testes. Nesta etapa é utilizado um detector de faces na informação de cor para localizar uma face na imagem, se uma face foi localizada ela é então recortada para descartar o fundo e manter apenas a informação relevante da face. Após o recorte a face é normalizada para um tamanho padronizado e a imagem a cores é convertida para escalas de cinza e a imagem de profundidades para valores entre 0 e 255. Após esta etapa a face tem sua pose em relação à câmera estimada utilizando a informação de profundidade provida pelo Kinect. A Figura 4.1 exemplifica em forma de diagrama cada passo do pré-processamento das imagens proposto.

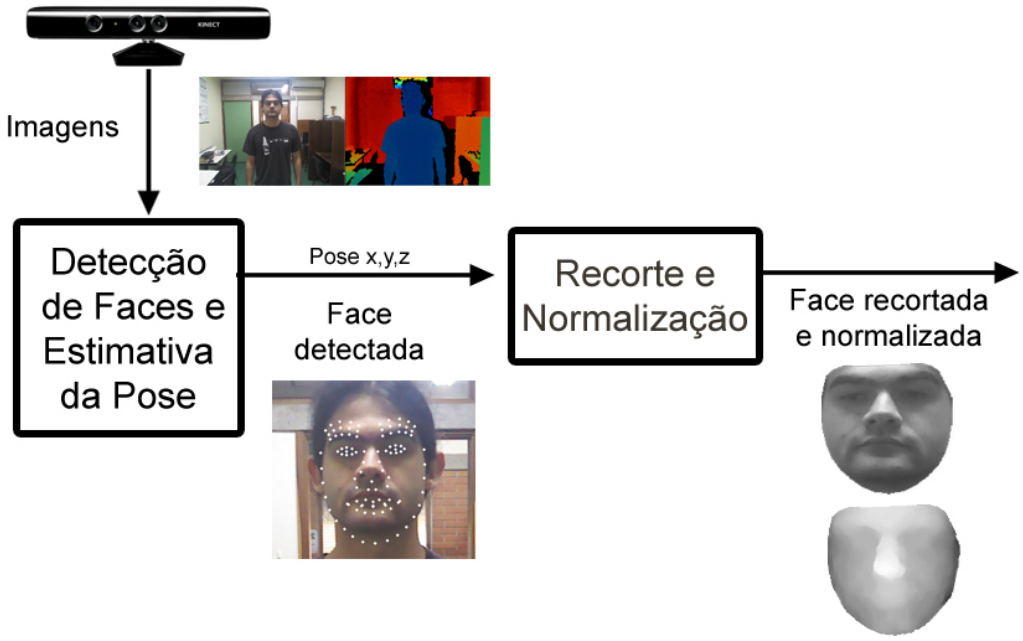


Figura 4.1: Pré-processamento das imagens adquiridas com o Kinect. Primeiro uma face é localizada nas imagens e tem sua pose estimada. A face é então recortada e normalizada nas imagens colorida e de profundidades.

4.2.1 Detecção de Faces e Estimativa de Pose

O primeiro passo de um sistema de reconhecimento facial é a detecção de faces nas imagens (JAIN; LI, 2005), caso na imagem não existirem faces presentes, o sistema não tem motivos para passar para a próxima etapa. Para localizar faces nas imagens foi utilizado o método de detecção de faces baseado em *Active Appearance Models* (ZHOU et al., 2010). Este método foi escolhido por dois motivos, o primeiro é que o mesmo vem implementado de forma otimizada no SDK do Kinect [1] e segundo é que métodos baseados

[1] Disponível em http://www.microsoft.com/en-us/Kinectforwindows/

em Active Apperance, pela sua natureza, já ajustam um conjunto de pontos na face localizada, onde cada ponto tem um identificador único e posição nas faces localizadas. Um exemplo de resultado do detector de faces utilizado é mostrado na Figura 4.2, os pontos em branco(landmarks) são os pontos ajustados pelo AAM na face detectada.

Figura 4.2: Exemplo de uma face detectada utilizando AAM. Os pontos em branco são os pontos que foram ajustados à face.

A informação de pose da cabeça de uma face em relação à câmera pode ser uma informação útil para o reconhecimento facial. De fato, o método proposto aqui nesta dissertação é dependente dessa informação para funcionar. A estimativa de pose é feita utilizando a implementação que também vem disponível no SDK do Kinect.

O SDK do Kinect utiliza a técnica de alinhamento de nuvens de pontos Iterative Closest Point (ICP)(seção 3.1.6) para estimar a pose da cabeça de uma face em relação à câmera. Como resultado o método do Kinect retorna as coordenadas de rotação$(yaw, pitch, roll)$ e translação(x, y, z) da cabeça em reação à câmera.

4.2.2 Segmentação e Normalização das Faces

Alguns dos pontos $\mathbf{p}$ ajustados na face pelo detector de faces baseado em AAM são utilizados para realizar a seleção e recorte da área de interesse da face nas imagens de cores e profundidades. Utilizando os pontos ajustados nas bordas da face é gerado um polígono. Cada ponto ajustado pelo AAM na face tem um identificador único e mesma localização de região na face. A Figura 4.3 demonstra o polígono e os pontos que formam este polígono nessa etapa.

Com a região do polígono definida o método cria uma imagem binária onde os pontos dentro do polígono são marcados com 1 (pixels da área de interesse) e pontos fora do polígono são marcados com 0 (pixels sem interesse que serão descartados). A Figura 4.4 demonstra a imagem binária resultante dessa etapa. Para desenhar o polígono foi utilizada a função do *matlab* "poly2mask" [2] que desenha um polígono binário a partir de um conjunto de pontos em uma imagem. Portanto foram passados para esta função apenas os pontos pertencentes as borda da face detectada.

Utilizando a imagem binária, são excluídos pixels que não pertencem à face na imagem original, para isto os pixels na imagem original que estão desligados na imagem

[2]Mais detalhes em: http://www.mathworks.com/help/images/ref/poly2mask.html

Figura 4.3: Exemplo do polígono, em vermelho, formado pelos pontos externos ajustados em uma face pelo AAM.

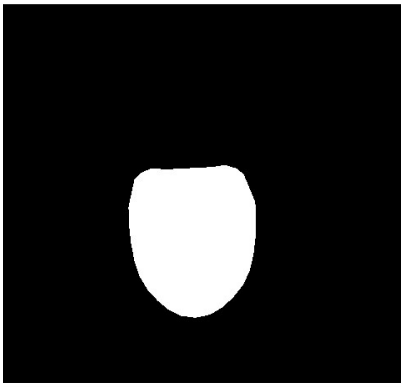


Figura 4.4: Imagem binária que informa a região de interesse da face.

binária (valor igual a 0) são substituídos com o valor $RGB = [0, 0, 0]$, ou seja,

$$I(x, y) \begin{cases} [0, 0, 0] & \text{Se } IB(x, y) = 0 \\ I(x, y) & \text{Se } IB(x, y) \neq 0, \end{cases} \tag{4.1}$$

onde I é a imagem colorida original e IB é a imagem binária, a Figura 4.5 demonstra o resultado desta etapa depois de ser aplicada na imagem colorida original.

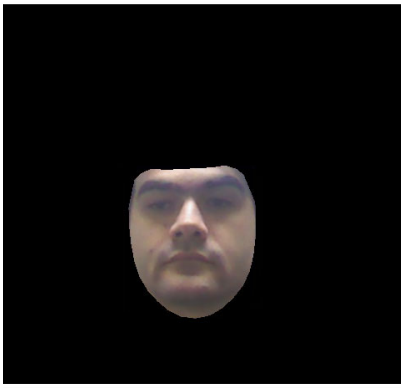


Figura 4.5: Remoção dos pixels não pertencentes à face na imagem colorida.

Após remover todos os pixels não pertencentes à face da imagem colorida boa parte da imagem vai conter pixels que não tem nenhum valor(pixels preenchidos com $RGB = [0, 0, 0]$), para eliminar estes pixels é recortado o menor retângulo possível da região que tem informação relevante da face. O retângulo é definido como não tendo rotação e é definido por dois pontos, A e B, onde o ponto A é o ponto superior à esquerda do retângulo e o ponto B é o ponto inferior à direita do retângulo. A Figura 4.6 demonstra o retângulo definido pelos dois pontos. Para encontrar os valores de (x, y) de cada um dos pontos utiliza-se a imagem binária IB. Os valores de (x, y) de A são definidos pelos menores valores de x e y ligados na imagem IB, ou seja:

$$A = (x_{min}, y_{min}), \tag{4.2}$$

onde x_{min} é definido por $\arg\min_X IB(x, y) = 1$ e y_{min} é definido por $\arg\min_y IB(x, y) = 1$. Já os valores de (x, y) de B são definidos pelos maiores valores de x e y ligados na imagem IB, ou seja:

$$B = (x_{max}, y_{max}), \tag{4.3}$$

onde x_{max} é definido por $\arg\max_X IB(x, y) = 1$ e y_{max} é definido por $\arg\max_y IB(x, y) = 1$.

O retângulo definido é utilizado para gerar uma nova imagem I_R usando o recorte da região interna do retângulo na imagem RGB que contém apenas os pixels relevantes da face. Após ser recortada, a imagem I_R é interpolada para um tamanho fixo padronizado de $t \times t$ pixels onde t é o novo tamanho da imagem, a interpolação é feita utilizando interpolação bi-cúbica. A Figura 4.7 demonstra a face da imagem após ser interpolada. A última etapa é converter a face para escalas de cinza gerando a imagem I_{gray} como é mostrado na Figura 4.8. Para converter a imagem para escalas de cinza foi utilizada a função do *matlab* "rgb2gray" [3].

[3]Mais detalhes em: http://www.mathworks.com/help/images/ref/rgb2gray.html

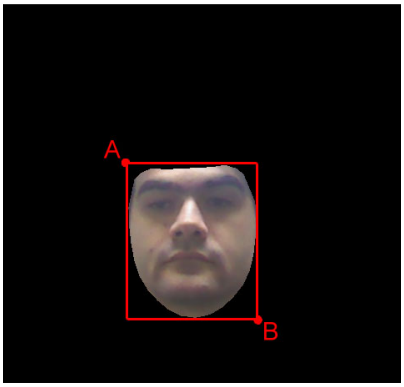


Figura 4.6: Retângulo de recorte definido pelos pontos A e B.

Figura 4.7: Face após ser recortada e normalizada para um tamanho padrão de $t \times t$ pixels.

Figura 4.8: Resultado da conversão para escalas de cinza.

A imagem em escalas de cinza é transformada em um vetor coluna $1 \times t^2$, onde cada posição do vetor segue sequêncialmente os pixels da imagem de escalas de cinza, sendo a imagem em escalas de definida pela matriz:

$$I_{gray} = \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,t} \\ a_{2,1} & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ a_{t,1} & \cdots & \cdots & a_{t,t} \end{bmatrix}, \tag{4.4}$$

onde $a_{i,j}$ são os pixels da imagem em escala de cinza. O vetor coluna resultante desta matriz é definido como:

$$\mathbf{f_{RGB}} = [a_{1,1}\, a_{1,2}\, \cdots\, a_{1,t}\, a_{2,1}\, a_{2,2}\, \cdots\, a_{2,t}\, \cdots\, a_{t,1}\, a_{t,2}\, \cdots\, a_{t,t}]^T, \tag{4.5}$$

onde $\mathbf{f_{RGB}}$ é o resultado final da imagem de uma face devidamente recortada, normalizada e rasterizada em forma de um vetor coluna. Assim como a imagem colorida, a face da de profundidades também é recortada da imagem original. Para encontrar a face na imagem de profundidades foram utilizados os pontos ajustados na face da imagem colorida pelo AAM. Como os espaços das câmeras do Kinect (colorida e de profundidades) não são iguais, ou seja, as posições (x,y) de cada pixel em uma imagem não é exatamente correspondente entre as duas câmeras. Para solucionar este problema é necessário encontrar uma transformação que transforme os pixels do espaço de uma câmera para o espaço da outra. No caso foi utilizado uma função de retificação de pontos entre os espaços das câmeras que está disponível no SDK do Kinect. Essa função foi utilizada para transformar os pontos ajustados do AAM na face da imagem colorida para o espaço da imagem de profundidades. Tendo os pontos AAM no espaço da câmera de profundidades, os mesmos são utilizados para recortar a face da imagem de profundidade de forma semelhante à imagem colorida.

Após o recorte da face a imagem de profundidades tem seus valores normalizados para valores entre $[0255]$ utilizando a seguinte equação:

$$D(x,y) = 255 - \frac{D(x,y) - [\mathbf{min}(D > 0) - 1]}{\mathbf{max}(D)} * 255 \tag{4.6}$$

onde D é a imagem recortada da face. A Figura 4.9 demonstra o resultado da face recortada e normalizada.

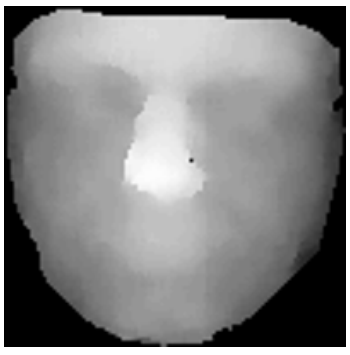


Figura 4.9: Resultado do recorte e normalização de uma face da imagem de profundidades.

Após o recorte, um filtro da mediana de 7×7 é aplicado para suavizar e remover ruído da imagem de profundidades. A Figura 4.10 mostra o resultado final da imagem de profundidades depois de ser recortada, normalizada e suavizada.

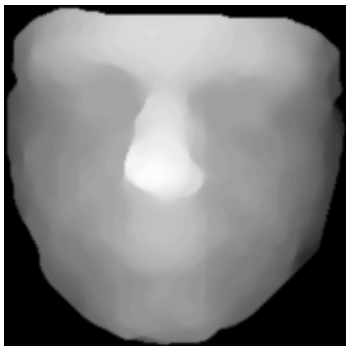


Figura 4.10: Resultado do recorte de ma face da imagem de profundidades.

Após o processamento a face da imagem de profundidades é convertida para um vetor coluna, da mesma forma que a imagem colorida:

$$\mathbf{f_D} = [a_{1,1}\ a_{1,2}\ \cdots\ a_{1,t}\ a_{2,1}\ a_{2,2}\ \cdots\ a_{2,t}\ \cdots\ a_{t,1}\ a_{t,2}\ \cdots\ a_{t,t}]^T, \tag{4.7}$$

Finalmente a os vetores das imagens colorida e de profundidades são concatenados para formar um único vetor de face

$$\mathbf{f} = \begin{bmatrix} \mathbf{f_{RGB}} \\ \mathbf{f_D} \end{bmatrix} \tag{4.8}$$

o vetor $\mathbf{f}$ será utilizado posteriormente para treinamento e classificação.

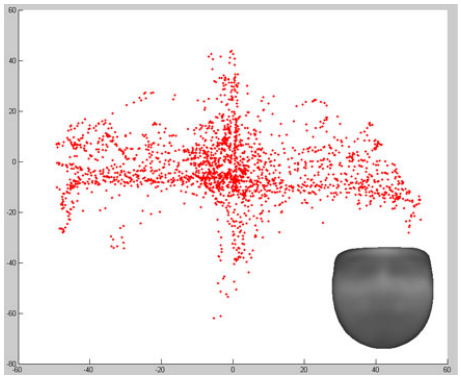
4.3 K-Fisherfaces

Para lidar com o problema de variação de poses, propõem-se separar o problema de classificação de faces em K sub-problemas de classificação. Para isto, um conjunto de faces é separado em diferentes grupos de poses, onde cada grupo tem seu próprio espaço de dimensão reduzida que é encontrado utilizando LDA, mais especificamente, utiliza-se o método Fisherfaces, o qual é utilizado para encontrar a matriz de projeção para cada grupo especifico de pose. Para determinar os K grupos de pose é utilizado o método K-means.

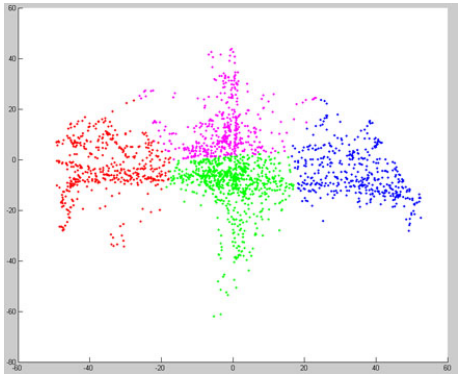
Dado um conjunto de faces em diferentes poses, pode-se calcular a imagem média deste conjunto, utilizando suas imagens. A Figura 4.11(a) exemplifica uma imagem média de um conjunto de faces em diferentes poses. Nesta figura também é apresentado o espaço das poses estimadas deste conjunto de faces, onde cada ponto é uma face do conjunto. Para apresentar melhor os dados, são apresentados apenas os eixos do yaw (eixo x) e pitch (eixo y) de cada pose. O eixo roll(eixo z) da pose foi ignorado apenas para apresentar melhor o gráfico. Como pode-se observar a imagem média de todas as faces ficou bastante borrada e a mesma não descreve coerentemente nenhuma pose.

Uma alternativa interessante para melhorar a qualidade da imagem média, é separar este conjunto de faces em sub-conjuntos. Para realizar esta separação pode ser utilizado o método K-means com a informação da pose estimada de cada face. A Figura 4.11(b) demonstra o mesmo espaço de poses da Figura 4.11(a) dividido em 4 sub-conjuntos($K = 4$)

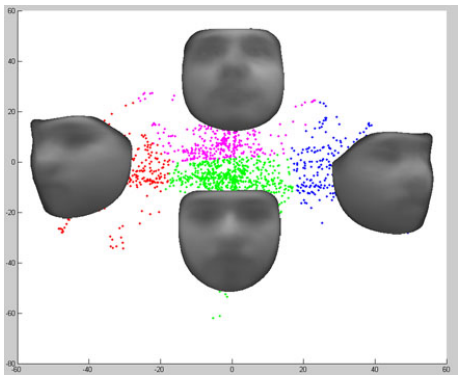
de pose, onde cada cor representa um grupo de pose(sub-espaço) diferente. Se para cada K encontrado for calculada uma imagem média apenas das faces pertencentes ao mesmo, pode-se verificar que a mesma irá descrever melhor uma determinada pose. A Figura 4.11(c) demonstra as imagens médias de cada K encontrado, como pode-se verificar, as imagens médias ficaram menos borradas e descrevem melhor diferentes poses.



(a) Espaço de poses de um conjunto de faces de treinamento e a face média correspondente das imagens RGB



(b) O mesmo espaço de poses da Figura 4.11(a) separado em 4 ($K = 4$) grupos diferentes de poses. Cada cor corresponde a um grupo de pose distinto



(c) Imagens médias dos grupos de poses demonstrados na figura Figura 4.11(b)

Figura 4.11: Exemplo de separação de um conjunto de faces em grupos de pose e suas imagens médias.

A seguir é explicado detalhadamente como é feito o treinamento do modelo de faces do K-Fisherfaces aqui proposto e como o modelo treinado é utilizado para realizar a classificação de novas faces.

4.3.1 Treinamento do Modelo de Faces K-Fisherfaces

Para lidar com o problema de variação de poses propõe-se dividir o problema de classificação em sub-problemas de classificação baseados na informação de pose da cabeça, ou seja, divide-se o conjunto de faces de treinamento em K diferentes agrupamentos que tenham pose em comum e para cada agrupamento de pose treina-se um modelo do Fisherfaces(i.e encontra-se a matriz de projeção W).

Para o método proposto treinar um modelo de faces é necessário um conjunto de n imagens de faces vetorizadas $F = \{\mathbf{f}_1, \mathbf{f}_2, \cdots, \mathbf{f}_n\}$ de C classes(pessoas) diferentes, as quais devem estar devidamente identificadas $ID = \{id_1, id_2, \cdots, id_n\}$ onde $id \in [1, C]$ e suas respectivas poses estimadas $P = \{\mathbf{p}_1, \mathbf{p}_2, \cdots, \mathbf{p}_n\}$, onde id_i identifica a classe e $\mathbf{p}_i$ a pose correspondentes à face $\mathbf{f}_i$.

Para separar o conjunto de faces de treinamento em K agrupamentos de poses, utiliza-se o método k-means(descrito na seção 3.2.2) no espaço de poses P das faces. Cada centróide μ_k encontrado pelo K-means irá representar um agrupamento diferente de pose. Utilizando a informação de pose $\mathbf{p}$ de cada face, pode-se separar as imagens $\mathbf{f}$ em K diferentes agrupamentos minimizando a distância euclidiana entre a pose de cada face e os centróides μ_k, ou seja:

$$K_i = \arg\min_k \|\mu_k - \mathbf{p}_i\| . \tag{4.9}$$

Entretanto este método de separação tem um problema, alguns agrupamentos podem ficar sem a informação de algumas das classes caso a classe não tenha informação perto o suficiente do centróide μ_k do agrupamento em questão. Para garantir que todos os agrupamentos tenham informação de todas as classes optou-se utilizar uma abordagem um pouco diferente. Em vez de verificar a qual centróide cada face $\mathbf{f}$ pertence, o método proposto seleciona as Ψ faces $\mathbf{f}$ com pose mais próximas de cada centróide μ_k de cada classe, com isto garante-se que cada agrupamento de pose vai ter a mesma quantidade de informação de todas as classes. Para realizar esta etapa primeiro separa-se as poses das imagens de treinamento por classes:

$$\delta_c = P \in [ID = c], \tag{4.10}$$

onde $c \in [1, C]$. Para selecionar as Ψ poses mais próximas de cada centróide, calcula-se a distância de todas as imagens de cada classe com cada um dos centróides:

$$dist(\mu_k, \delta_{c,i}) = \|\mu_k - \delta_{c,i}\| , \tag{4.11}$$

onde $\delta_{c,i}$ é cada uma das poses selecionadas da classe δ_c. Finalmente cada vetor de distâncias $dist(\mu_k, \delta_{c,i})$ é ordenado de forma crescente e são selecionadas as Ψ primeiras posições do vetor ordenado. Os índices das poses selecionadas de cada classe são armazenados em um vetor auxiliar $indices(\mu_k)$, onde são armazenados os índices das Ψ poses mais próximas de cada classe ao centróide μ_k. Com os índices das faces mais próximas são selecionadas as imagens que serão utilizadas para o treinamento de cada um dos K modelos de pose

$$\Omega(k) = F \in indices(k). \tag{4.12}$$

Para cada $\Omega(k)$ utiliza-se LDA para reduzir a dimensionalidade, onde busca-se encontrar uma matriz de projeção W ótima. Assim como em (BELHUMEUR; HESPANHA; KRIEGMAN, 1997) as imagens tiveram sua dimensionalidade reduzida utilizando primeiro PCA, isto é feito para garantir que matriz S_w não seja singular. Para a PCA de cada $\Omega(k)$ o primeiro passo é calcular a matriz de covariância dos dados, que é definida como

$$S(k) = \sum_{i=1}^{\Psi * C} (\Omega(k)_i - \theta_{pca}(k))(\Omega(k)_i - \theta_{pca}(k))^T, \tag{4.13}$$

onde $\theta_{pca}(k)$ é o vetor médio de $\Omega(k)$. A seguir, são calculados os auto-vetores e auto-valores de cada $S(k)$ e são selecionados os V auto-vetores associados com os V maiores auto-valores de $S(k)$, dando origem à matriz de projeção $W_{pca}(k)$. A matriz de projeção $W_{pca}(k)$ é utilizada para converter $\Omega(k)$ para o espaço reduzido da PCA:

$$\Omega_{pca}(k) = W_{pca}(k)^T[\Omega(k) - \theta_{pca}(k)]. \tag{4.14}$$

Após a redução de dimensionalidade em $\Omega(k)$, são calculados o espalhamento inter-classes de cada agrupamento que é definido como:

$$S_B(k) = \sum_{i=1}^{C} \Psi(\theta_i - \theta)(\theta_i - \theta)^T, \tag{4.15}$$

e o espalhamento intra-classes de cada agrupamento que é definido como:

$$S_W(k) = \sum_{i=1}^{C} \sum_{\Omega_{pca}(k)_j \in C_i} (\Omega_{pca}(k)_j - \theta_i)(\Omega_{pca}(k)_j - \theta_i)^T, \tag{4.16}$$

onde θ é a vetor médio entre todas as classes de $\Omega_{pca}(k)$ e θ_i é o vetor médio da classe C_i.

Se S_W for uma matriz não singular, a projeção ótima W_{opt} é escolhida como a matriz com colunas ortonormais que maximiza a taxa do determinante da matriz de espalhamento entre classes dos exemplos projetados e também que minimiza a taxa do determinante da matriz de espalhamento intra-classe dos exemplos projetados

$$W_{fisher}(k) = \underset{W_{fisher}(k)}{\arg\max} \frac{\left|W_{fisher}(k)^T S_B W_{fisher}(k)\right|}{|W_{fisher}(k)^T S_W W_{fisher}(k)|} = [w_1 w_2 \cdots w_m]. \tag{4.17}$$

As faces selecionadas para o treinamento são projetadas no novo espaço de faces e armazenadas para futura classificação:

$$\Phi(k) = W_{fisher}(k)^T \Omega_{pca}(k). \tag{4.18}$$

4.3.2 Classificação das Faces

Para verificar se uma nova face $\mathbf{f}$ está ou não presente em $\Phi(k)$(base de faces), é necessário primeiro converter $\mathbf{f}$ para o espaço com dimencionalidade reduzida, para isso é necessário selecionar o modelo de pose mais próximo da face, ou seja seleciona-se o centróide μ_k com a distância euclidiana mais próxima da pose $\mathbf{p}$ relacionada com $\mathbf{f}$:

$$m = \underset{k}{\arg\min} \left\|\mu_k - \mathbf{p}\right\|, \tag{4.19}$$

onde m é o índice do centróide mais próximo, a seguir, a imagem da face é convertida para o espaço grupo de pose selecionado. Portanto, primeiro $\mathbf{f}$ é converida para o espaço da PCA:

$$\mathbf{f}_{pca} = W_{pca}(m)^T[\mathbf{f} - \theta_{pca}(m)], \tag{4.20}$$

e posteriormente para o espaço do fisherfaces:

$$\mathbf{f}_{fisher} = W_{fisher}(m)^T\mathbf{f}_{pca}, \tag{4.21}$$

para verificar a face mais próxima na base de faces, é minimizada a distância euclidiana entre $\Phi(k)$ e $\mathbf{f}_{fisher}$:

$$\varepsilon_S = \|\mathbf{f}_{fisher} - \Phi(m)_S\|\,, \tag{4.22}$$

se a distância ε_S entre as duas faces estiver abaixo de um dado limiar Θ_ε, $\mathbf{f}$ pertence a classe de S .

5 RESULTADOS EXPERIMENTAIS

Para realizar os experimentos foram gravadas duas bases de faces, uma das bases é gravada em um cenário controlado e a outra em um cenário sem restrições. A seção 5.1 detalha como cada uma das bases de faces foi gravada. Na seção 5.2 explica como as bases foram pré-processadas e organizadas para os experimentos.

Nos experimentos o método proposto é comparado com os métodos clássicos eigenfaces (TURK; PENTLAND, 1991) e fisherfaces (BELHUMEUR; HESPANHA; KRIEGMAN, 1997) utilizando as bases gravadas em quatro cenários de testes. Primeiro é avaliado o desempenho dos métodos quando não existe variação de pose ou expressão, em seguida é avaliado como é o desempenho dos métodos quando existe variação de expressão e pose, no terceiro experimento verifica-se o desempenho dos métodos quando apenas uma expressão é utilizada para treinamento e no quarto experimento os métodos são avaliados em um cenário onde existe uma grande variação de pose utilizando a base sem restrições. Estes experimentos e seus resultados são descritos na seção 5.3. Na seção 5.3.5 o método proposto é comparado com o estado da arte.

5.1 Bases de Faces Gravadas

5.1.1 Base de Faces com Restrições

Esta base é chamada de "base de faces com restrições"pois ela foi gravada em um cenário mais controlado. Durante a gravação foram impostas algumas restrições de pose e expressões para cada participante. Uma pessoa auxiliou durante a gravação para garantir que o participante estivesse realizando as poses e expressões corretamente.

A base é composta por imagens de 54 pessoas diferentes de ambos os sexos, para cada pessoa foram gravadas 4 poses padronizadas diferentes de cabeça: olhando para o Kinect, olhando para à esquerda, olhando para cima e olhando para à direita. Para cada uma das poses foram gravadas 4 expressões faciais diferentes: Sorrindo, expressão neutra, com a boca aberta e expressão de bravo. Para cada expressão foram registrados 20 frames com a supervisão de uma pessoa. Cada uma das pessoas gravadas tem um total de $4 \times 4 \times 20 = 320$ frames registrados. A Figura 5.2 demonstra algumas das pessoas gravadas e suas respectivas poses e expressões registradas. Todos os frames registrados foram processados para conter as informações de detecção de faces AAM e estimativa de pose da cabeça.

Para auxiliar na gravação da base o Kinect foi posicionado em frente a uma parede sem janelas e foram colados 3 marcadores em volta do Kinect, cada marcador possui um número diferente. Os marcadores foram posicionados a 55 centímetros de distância do Kinect, a Figura 5.1 exemplifica como foram dispostos os marcadores na parede. A ideia

de utilizar os marcadores foi retirada de (HG et al., 2012). O objetivo dos marcadores é servir de referência para o participante na hora de mudar de pose.

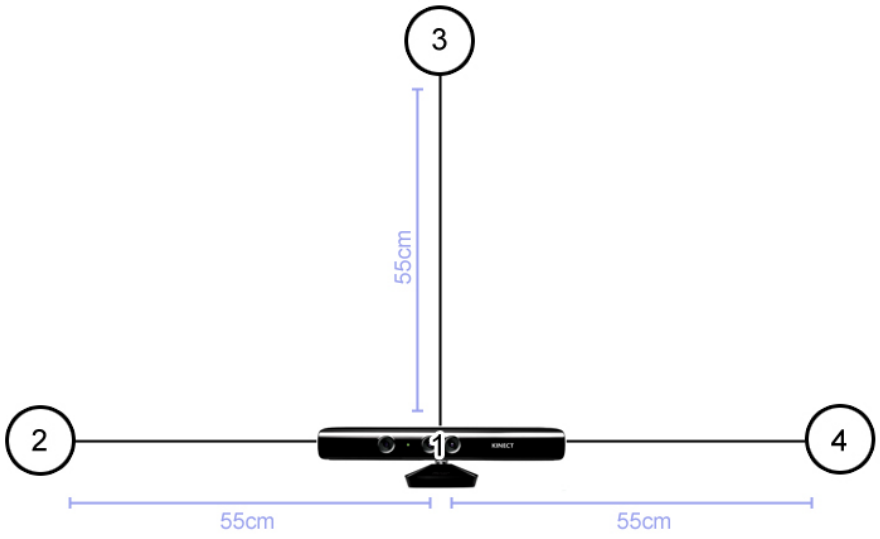


Figura 5.1: Marcadores que foram colados em volta do Kinect na parede, o Kinect é considerado como o primeiro marcador. A distância entre cada marcador e o Kinect é de 55cm.

Figura 5.2: Exemplo de algumas das pessoas gravadas em diferentes poses e expressões, estas imagens fazem parte da base de dados com restrições descrita na seção 5.1.1.

5.1.2 Base de Faces Sem Restrições

Nesta base as faces foram adquiridas em um cenário mais flexível, onde os frames utilizados para treinar e classificar não seguem regras rígidas de aquisição. Não foi solicitado ao participante que olhasse para os marcadores, em vez disto, foi solicitado apenas que o participante movimentasse a sua cabeça livremente em frente à câmera. Foram adquiridos todos os frames de um período de tempo não regular, ou seja, os vídeos de diferentes participantes tem tempo e quantidade de frames diferentes. Esta base tem uma variação de poses muito maior que na base descrita anteriormente e estas poses não seguem um padrão como na base anterior. Outro diferencial desta base é que foram gravadas duas tomadas de vídeos para cada participante, uma tomada para treinamento e uma tomada para classificação. Nas tomadas de treinamento os participantes ficam em uma posição parada na cena e movimentam a cabeça livremente em diferentes poses. Já nas tomadas de classificação, foi solicitado aos participantes que viessem caminhando em direção da câmera e que parassem em um certo ponto do cenário, após parar de caminhar o participante movimenta a sua cabeça livremente em frente à câmera. Para esta base foram gravadas tomadas de 11 participantes diferentes. A Figura 5.3 demonstra alguns frames das tomadas de treinamento de alguns dos participantes e a Figura 5.4 demonstra alguns frames das tomadas de classificação dos mesmos participantes. É interessante observar a diferença de localização na cena entre os vídeos de treinamento e de classificação, além de existirem frames borrados em virtude do movimento rápido da cabeça dos participantes.

5.2 Pré-processamento das Bases Adquiridas

Nesta seção serão descritos os passos realizados para preparar as imagens das bases de dados para os experimentos. Inicialmente detecta-se faces em todas as imagens utilizando AAM e estima-se suas relativas poses utilizando ICP. Para este trabalho foi utilizada a implementação do AAM e ICP que está disponível no SDK do Kinect.

Após as faces serem detectadas e terem suas poses estimadas, é feita a etapa de recorte e normalização das faces descrita na seção 4.2. Todas as faces detectadas nas imagens coloridas foram normalizadas para terem tamanho de 200×200 pixels e as imagens de profundidade foram normalizadas para terem tamanho de 100×100 pixels.

Para os experimentos as bases foram divididas em dois grupos, o grupo de treinamento e o grupo de classificação. Como a base com restrições não tem uma gravação para cada grupo foram selecionados os 5 primeiros frames de cada expressão de cada pose para treinamento e os últimos 5 frames de cada expressão de cada pose para classificação, deixando um intervalo de 10 frames entre os frames de treinamento e classificação. Como na base de vídeos foram gravados vídeos para treinamento e classificação cada grupo recebeu seus respectivos frames gravados. Na base sem restrições, quadros que não tiveram uma face detectada são descartados e não são contabilizados nos resultados dos métodos avaliados.

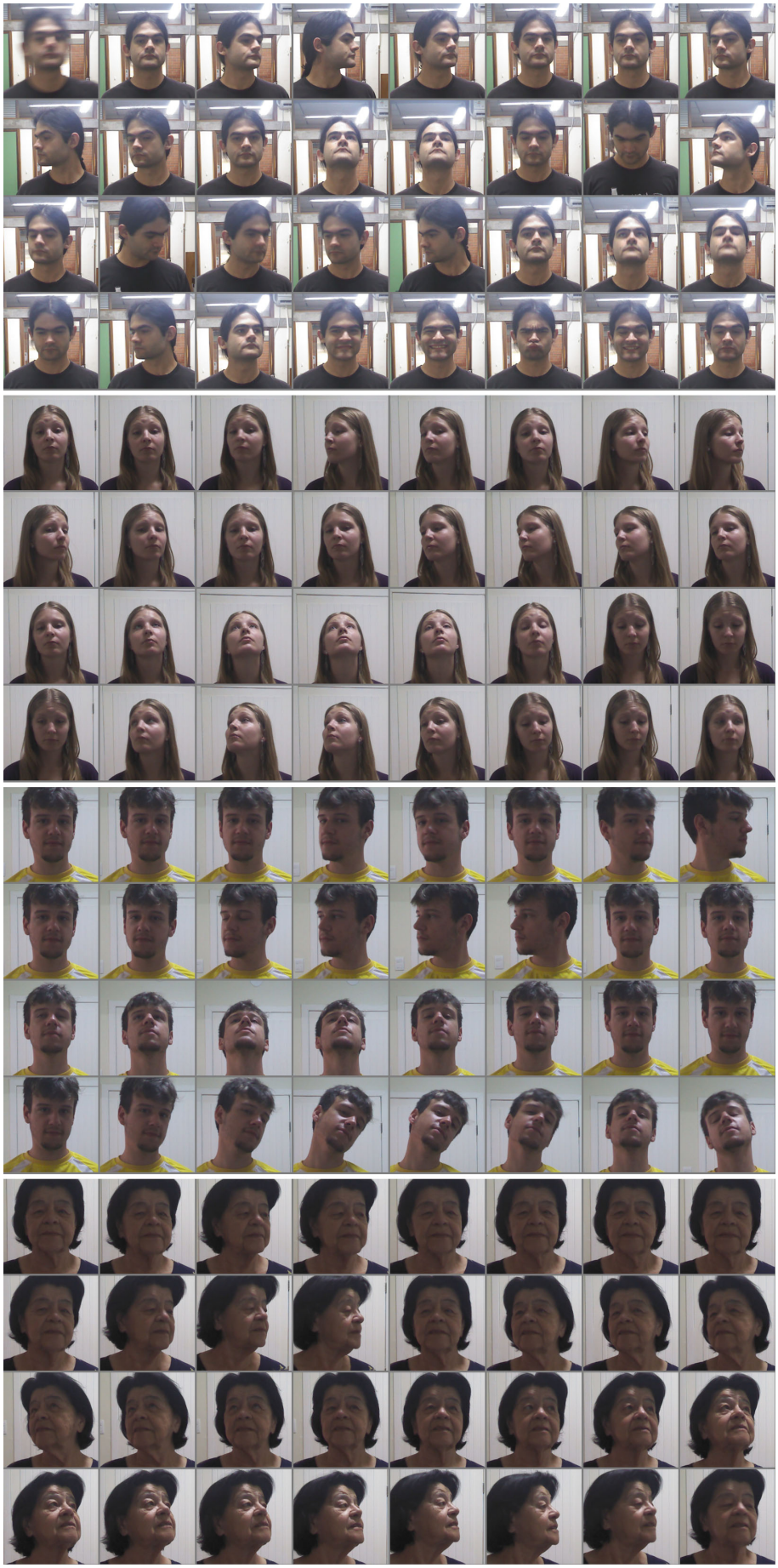


Figura 5.3: Exemplo de algumas das tomadas gravadas para treinamento.

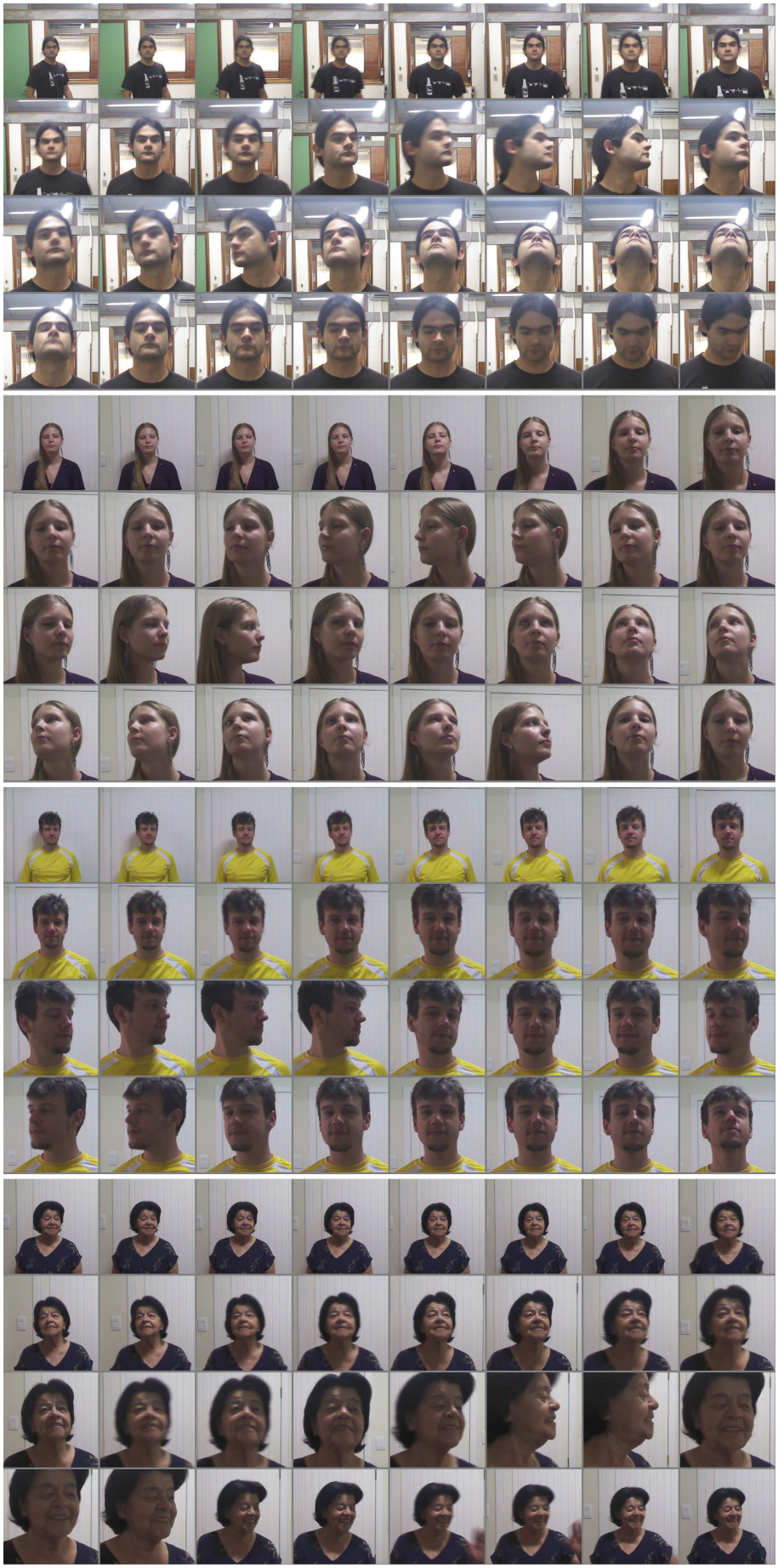


Figura 5.4: Exemplo de algumas das tomadas gravadas para a testar a classificação.

5.3 Testes Comparativos Realizados

Nesta seção serão avaliados comparativamente os resultados do método proposto e dos métodos fisherfaces e eigenfaces. Para verificar o desempenho dos métodos foram realizados quatro experimentos utilizando as duas bases de dados gravadas.

Os três primeiros experimentos utilizam a base de faces com restrições em três cenários de testes diferentes. O primeiro cenário avalia os resultados utilizando apenas faces em pose frontal e expressão neutra tanto para treinamento quanto para classificação. O segundo cenário avalia os resultados utilizando todas as poses e todas as expressões tanto para treinar quanto para classificar. Já o terceiro cenário avalia os resultados utilizando apenas as expressões neutras em todas as poses para treinar e todas as expressões de todas as poses para testar a classificação.

No quarto experimento é avaliado o desempenho do eigenfaces e do fisherfaces utilizando a base sem restrições, onde o cenário não é controlado como na base com restrições e aproxima-se mais de uma aplicação real onde não existe intervenção humana para selecionar as imagens. Para padronizar os experimentos o eigenfaces foi treinado variando o número de componentes principais entre $[20\ 30\ 40\ 60\ 80]$. Já para o fisherfaces foram utilizadas sempre as 100 primeiras componentes principais.

5.3.1 Teste 1: Faces em pose Frontal e Expressão neutra

O primeiro experimento realizado tem como objetivo testar o desempenho utilizando apenas as faces em pose frontal e com expressão neutra. Para o experimento foi utilizada a base com restrições, foram selecionadas manualmente apenas as faces em posição frontal(i.e. olhando para o Kinect) e em expressão neutra. A Tabela 5.1 na coluna "Teste 1"demonstra os resultados do fisherfaces e eigenfaces neste experimento. O número que acompanha o eigenfaces na coluna "Método"indica a quantidade de componentes principais que foram utilizadas. Como pode-se verificar o fisherfaces tem um resultado um pouco melhor que o eigenfaces, algo já esperado e que já foi comprovado em (BELHUMEUR; HESPANHA; KRIEGMAN, 1997). Neste experimento o método proposto não foi testado pois não existe variação de pose.

Método	Teste 1	Teste 2	Teste 3	Teste 4
Método Proposto	-	**99.09%**	**85.20%**	**79.24%**
Fisherfaces	**99.25%**	99.05%	84.60%	67.27%
Eigenfaces 20	97.03%	97.52%	78.54%	65.15%
Eigenfaces 30	97.03%	98.14%	80.11%	69.69%
Eigenfaces 40	97.40%	98.24%	80.62%	70.55%
Eigenfaces 60	97.77%	98.26%	80.94%	71.81%
Eigenfaces 80	97.77%	98.35%	81.15%	71.52%

Tabela 5.1: Resultados dos testes realizados, para o fisherfaces foram utilizadas as 100 principais componentes e para o eigenfaces foram utilizadas as principais componentes variando em 20, 30, 40, 60 e 80.

5.3.2 Teste 2: Variação de Pose e Expressão

Neste experimento será avaliado o resultado dos métodos testados quando há variação de pose e expressão. Para o experimento são utilizadas todas as 4 poses e as suas 4 expressões gravadas na base com restrições. No experimento são utilizadas todas as poses

e expressões tanto para treinar os métodos quanto para testar. A Tabela 5.1 demonstra os resultados dos métodos testados neste experimento na coluna "Teste 2".

Os valores de K e Ψ do método proposto foram encontrados empiricamente, no caso $\Psi = 30$ e $K = 19$. Como podes-se observar os métodos testados continuam a ter um bom resultado, isso se da ao fato de estarem sendo utilizadas todas as expressões para o treinamento. No próximo experimento pode-se verificar que ao remover algumas expressões do treinamento reduz o resultado de classificação.

5.3.3 Teste 3: Apenas Faces com Expressão Neutra para Treinar o Modelo

Este experimento visa verificar como será o resultado dos métodos testados quando são utilizas faces em apenas uma expressão para treinamento, no caso deste experimento foram utilizadas apenas as faces em expressão neutra de todas as poses da base com restrições para o treinamento. A coluna "Teste 3"da Tabela 5.1 demonstra os resultados dos métodos testados neste experimento .

Os valores de K e Ψ do método proposto foram encontrados empiricamente, no caso $\Psi = 10$ e $K = 4$. Como pode-se verificar, o resultado do método proposto foi um pouco melhor que o fisherfaces, entretanto a diferença não é grande, o motivo disto acontecer é que foram utilizadas todas as poses para o treinamento neste experimento e as poses utilizadas para o treiamento e a classificação são semelhantes. No próximo experimento será utilizada a base sem restrições, onde existe uma grande variação de poses e as mesmas não estão padronizadas.

5.3.4 Teste 4: Variação de Pose em um Ambiente não Controlado

Neste experimento é utilizada a base sem restrições, serão utilizados todos os quadros dos vídeos de treinamento (Figura 5.3) para treinar os modelos e todos os frames dos vídeos de classificação (Figura 5.3) para testar o desempenho do método proposto. A Tabela **??** demonstra os resultados do método proposto neste experimento na coluna "Teste 4".

Os valores de K e Ψ do método proposto foram encontrados empiricamente, no caso $\Psi = 17$ e $K = 11$. Como pode-se verificar neste experimento, o método proposto tem um desempenho superior aos métodos tradicionais do eigenfaces e fisherfaces quando há uma grande variação de poses na base de treinamento e classificação. Conclui-se que isto se deve ao fato de o método proposto separar o problema de classificação em sub-problemas de classificação, cada um especialistas em um tipo pose, o que melhora consideravelmente o desempenho de classificação quando há um grande variação de pose.

5.3.5 Comparativo com o Estado da Arte

Dado o recente surgimento do Kinect ainda não existe uma base de faces padrão sendo utilizada pelos métodos do estado da arte, pelo conhecimento dos autores cada método utiliza a sua própria base de dados e ainda não existem experimentos padronizados para fazer um comparativo real de qual dos métodos do estado da arte obtém os melhores resultados e em quais condições. Infelizmente nenhum dos métodos do estado da arte fornecem seus códigos abertamente para realizar experimentos comparativos com bases de testes gravadas. Dadas estas condições, será adotada a mesma metodologia de comparação utilizada em (LI et al., 2013) onde os autores apenas apresentam os resultados de classificação de cada método e suas características em uma tabela comparativa, claramente esta não é a melhor abordagem para comparar os resultados dos métodos pois cada

método foi avaliado em uma base e com características distintas. A Tabela 5.3.5 demonstra um comparativo entre os resultados do método proposto e os métodos do estado da arte, na coluna "Resultados"o valor após a taxa de acerto significa o tipo de informação utilizado para treinar o modelo, por exemplo (2D +3D) significa que foram utilizadas as informações 2D e 3D fornecidas pelo Kinect. É interessante ressaltar que os métodos comparados utilizam bases semelhantes à base com restrições que gravamos, e pelo conhecimento do autor, nenhum método utiliza uma base semelhante à base sem restrições gravada. Por este motivo serão mostrados na tabela apenas os resultados dos testes "3"e "4"realizados. O método proposto tem resultados competitivos com o estado da arte dos métodos que utilizam o Kinect para reconhecimento facial.

Método	Base de Dados (número de pessoas)	Condições	Resultados
Método Proposto	Guardian Dataset (54)	Pose Expressão	99.09% (2D+3D) 85.20% (2D+3D)
(LI et al., 2013)	CurtinFaces (52)	Pose Expressão Iluminação Óculos Escuro	88.7% (3D) 91.1% (2D) 96.7% (2D+3D)
(GOSWAMI et al., 2013)	IIIT-D RGB-D (106)	Pose Iluminação Expressão	87.2% (2D) 91.6% (2D+3D)
(HG et al., 2012)	VAP RGB-D Face data set (31)	Pose Expressão	82.34% (3D)

Tabela 5.2: Comparativo entre os resultados do método proposto e os métodos do estado da arte.

6 CONCLUSÕES

A principal motivação deste trabalho foi o recente surgimento do Kinect que possibilita a aquisição de imagens 2D e 3D da cena. Dadas essa nova possibilidade de aquisição de imagens, o objetivo principal é a elaboração de uma técnica de reconhecimento facial, que a partir de um conjunto de imagens coloridas e de profundidades de faces humanas, seja capaz de aprender um modelo de faces robusto à variação de pose.

Tendo em vista este objetivo principal, foram elaborados três objetivos específicos, que em conjunto constituem a solução proposta para o problema. O primeiro objetivo específico foi o desenvolvimento de um método capaz de localizar, segmentar e normalizar faces presentes em imagens coloridas e estimar a pose da face em relação à câmera utilizando a imagem de profundidades. O segundo objetivo específico foi o desenvolvimento de um método capaz de aprender um modelo de faces que seja robusto à variação de poses. Já o terceiro objetivo específico foi validar, analisar e comparar o desempenho do método proposto utilizando imagens reais adquiridas por um Kinect.

Para detectar faces nas imagens foi utilizado um detector de faces baseado em *Active Appearance models*, o qual ajusta um conjunto de pontos na face detectada. Com o conjunto de pontos ajustado em uma face foi proposta uma nova abordagem de recorte de faces que mantém apenas a informação relevante da face na imagem final, além disto, a imagem recortada é padronizada para um tamanho padrão e convertida para escalas de cinza. Com a informação de profundidade foi utilizado o método de aproximação de nuvens de pontos 3D *Iterative Closest Point* para estimar a pose da cabeça em relação à câmera.

Foi proposto neste trabalho um novo método de reconhecimento facial que foi chamado de K-Fisherfaces que é mais tolerante à variação de poses que os métodos tradicionais. Para treinar este modelo foi utilizado o método K-means para descobrir diferentes sub-conjuntos de faces, cada qual especialista em um tipo diferente de pose. Para garantir que cada sub-conjunto de pose tenha informação de todas as classes, propõem-se a abordagem de selecionar as Ψ faces mais próximas de cada classe ao centróide relacionado a este sub-conjunto de pose.

Para avaliar o desempenho do método proposto foram gravadas duas bases de faces distintas, a primeira base, chamada de base com restrições, é composta por um conjunto de galerias de imagens de 54 pessoas que foram gravadas em 4 poses padronizadas cada uma com 4 expressões, já a segunda base, chamada de base sem restrições, é composta de vídeos de 11 pessoas movimentando suas cabeças em frente à câmera livremente, na segunda base foram gravados dois vídeos em momentos diferentes, um vídeo para treinamento e outro para classificação. Foram realizados 4 experimentos onde 3 comparam o método proposto com métodos clássicos.

O primeiro experimento verifica o desempenho dos métodos eigenfaces e fisherfaces

quando não há variação de expressões ou pose. Já o segundo experimento avalia o desempenho do método proposto e dos métodos clássicos quando existe variação de pose e expressões, porém neste experimento foram utilizadas todas as expressões e poses para o treinamento. No terceiro experimento foi avaliado o desempenho do método proposto e clássicos quando apenas um tipo de expressão é utilizado no treinamento. Já o quarto experimento avalia o desempenho dos métodos testados quando há uma grande variação de pose em um ambiente não controlado.

O método proposto obteve um resultado superior aos métodos clássicos do eigenfaces e fisherfaces, principalmente no último experimento, onde o método proposto obteve um resultado bastante superior aos resultados do eigenfaces e fisherfaces. Este resultado superior se dá ao fato do método proposto separar o problema de classificação em subproblemas de classificação, cada um especialista em um tipo de pose.

Ao comparar o método proposto com o estado da arte, foi possível verificar que o mesmo tem um desempenho muito próximo aos resultados dos métodos do estado da arte, infelizmente não foi possível comparar o método com os métodos do estado da arte utilizando as mesma base de dados.

6.1 Trabalhos Futuros

Como trabalhos futuros, pretende-se verificar se o método proposto é capaz de funcionar em tempo real, para isto pretende-se implementar o método em C++. Após a implementação pretende-se verificar os tempo necessário para treinamento e classificação do método proposto.

Outro ponto a explorar é o uso da informação temporal para melhorar os resultados de classificação de um indivíduo que está sendo identificado em uma sequência de vídeo. Um método como o *Hidden Markov Model* pode ser utilizado para explorar as consistências temporais no reconhecimento facial.

O uso de técnicas de calibração de cores para normalizar as imagens coloridas a fim de melhorar a acurácia do método proposto quando há variação de iluminação parece uma alternativa interessante para pesquisa.

Outro objetivo futuro é comparar o método proposto com os métodos do estado da arte, utilizando uma base padrão para treinar os seus respectivos modelos.

Nos experimentos pretende-se implementar rodadas de teste que alterne as faces de treinamento e classificação para ter resultados mais concisos. E também testar os métodos utilizado pessoas que não estejam no conjunto de treinamento.

Finalmente, pretende-se testar o Kinect 2.0 que foi lançado com o Xbox One ano passado, o novo hardware contêm câmeras com qualidade superior ao Kinect 1.0.

REFERÊNCIAS

ABATE, A. F. et al. 2D and 3D face recognition: a survey. **Pattern Recognition Letters**, [S.l.], v.28, n.14, p.1885 – 1906, 2007. Image: Information and Control.

BELHUMEUR, P. N.; HESPANHA, J. a. P.; KRIEGMAN, D. J. Eigenfaces vs. Fisherfaces: recognition using class specific linear projection. In: IEEE TRANS. PATTERN ANAL. MACH. INTELL. **Proceedings...** [S.l.: s.n.], 1997. v.19, n.7, p.711–720.

BESL, P.; MCKAY., N. A Method for Registration of 3-D Shapes. In: IEEE TRANS. PATTERN ANALYSIS AND MACHINE INTELLIGENCE. **Proceedings...** [S.l.: s.n.], 1992. v.v.14, p.239–256.

BISHOP, C. M. **Pattern Recognition and Machine Learning**. New York: Springer Science+Business Media, LLC, 2006.

BOWYER, K. W.; CHANG, K.; FLYNN, P. A survey of approaches and challenges in 3D and multi-modal 3D + 2D face recognition. **Computer Vision and Image Understanding**, [S.l.], p.210–227, 2005.

CHEN, Y.; MEDIONI, G. Object modeling by registration of multiple range images. **IEEE International Conference on Robotics and Automation**, [S.l.], 1991.

COLOMBO, A.; CUSANO, C.; SCHETTINI, R. 3D face detection using curvature analysis. **Pattern Recognition**, [S.l.], v.39, n.3, p.444 – 455, 2006.

COOTES, T.; EDWARDS, G.; TAYLOR, C. Active appearance models. In: COMPUTER VISION A ECCVA98. **Proceedings...** [S.l.: s.n.], 1998.

DALAL, N.; TRIGGS., B. Histograms of oriented gradients for human detection. In: CVPR. **Proceedings...** [S.l.: s.n.], 2005. p.886893.

FANELLI, G. et al. Real Time Head Pose Estimation from Consumer Depth Cameras. **Pattern Recognition, Lecture Notes in Computer Science**, [S.l.], v.6835, p.101–110, 2011.

FILHO, O. M.; NETO, H. V. **Processamento Digital de Imagens**. 1.ed. Rio de Janeiro: Brasport, 1999.

FREEDMAN, B. et al. Depth mapping using projected patterns. **Prime Sense Ltd**, [S.l.], p.29–31, 2010.

FRIEDMAN, N.; GEIGER, D.; GOLDSZMIDT, M. **Bayesian network classifiers. In**: machine learning. [S.l.: s.n.], 1997. 131-163p.

GONZALEZ, R. C.; WOODS, R. E. **Digital Image Processing (3rd Edition)**. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 2006.

GOSWAMI, G. et al. On RGB-D Face Recognition using Kinect. In: INTERNATIONAL CONFERENCE ON BIOMETRICS: THEORY, APPLICATIONS AND SYSTEMS. **Proceedings...** [S.l.: s.n.], 2013.

HARTLEY, R. I.; ZISSERMAN, A. **Multiple View Geometry in Computer Vision**. 2.ed. New York: Cambridge University Press, ISBN: 0521540518, 2004.

HG, R. et al. An RGB-D Database Using Microsoft's Kinect for Windows for Face Detection. In: SIGNAL IMAGE TECHNOLOGY AND INTERNET BASED SYSTEMS (SITIS), 2012 EIGHTH INTERNATIONAL CONFERENCE ON. **Proceedings...** [S.l.: s.n.], 2012. p.42–46.

HO., T. K. Random decision forests. In: ICDAR. **Proceedings...** [S.l.: s.n.], 1995. p.278282.

ITTI, L.; KOCH, C.; NIEBUR., E. A model of saliency-based visual attention for rapid scene analysis. **IEEE TPAMI**, [S.l.], p.12541259, 1998.

JAIN, A. K.; LI, S. Z. **Handbook of Face Recognition**. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2005.

KHOSHELHAM, K. ACCURACY ANALYSIS OF KINECT DEPTH DATA. In: INTERNATIONAL ARCHIVES OF THE PHOTOGRAMMETRY, REMOTE SENSING AND SPATIAL INFORMATION SCIENCES. **Proceedings...** [S.l.: s.n.], 2011. p.29–31.

LI, B. et al. Using Kinect for face recognition under varying poses, expressions, illumination and disguise. In: APPLICATIONS OF COMPUTER VISION (WACV), 2013 IEEE WORKSHOP ON. **Proceedings...** [S.l.: s.n.], 2013. p.186–192.

MATTHEWS, I.; BAKER, S. Active appearance models revisited. In: IJCV. **Proceedings...** [S.l.: s.n.], 2004.

PHILLIPS, P. et al. Overview of the Face Recognition Grand Challenge. In: IEEE CONF. ON COMPUTER VISION AND PATTERN RECOGNITION (CVPR), SAN DIEGO, CA. **Proceedings...** [S.l.: s.n.], 2005.

PRATT, W. K. **Digital Image Processing**. 2.ed. New York: Wiley Interscience, 1991.

RICHARD O. DUDA PETER E. HART, D. G. S. **Pattern Classification**. New York: 2ed. Willey, 2000.

RRNYI., A. On measures of entropy and information. **BSMSP**, [S.l.], p.547561, 1961.

RUSU, R. B. **Semantic 3D Object Maps for Everyday Manipulation in Human Living Environments**. 2009. 284p. phd — Tecnische Universitatet Muenchen, Munich, Germany.

SCHEENSTRA, A.; RUIFROK, A.; VELTKAMP, R. C. A Survey of 3D Face Recognition Methods. **In Lecture Notes in Computer Science, SpringerVerlag**, [S.l.], p.891–899, 2005.

SCHEENSTRA, A.; RUIFROK, A.; VELTKAMP, R. C. A Survey of 3D Face Recognition Methods. **In Lecture Notes in Computer Science, SpringerVerlag**, [S.l.], p.891–899, 2005.

SZELISKI, R. **Computer vision algorithms and applications**. London; New York: Springer, 2011.

TURK, M.; PENTLAND, A. Eigenfaces for Recognition. **J. Cognitive Neuroscience**, [S.l.], v.3, n.1, p.71–86, 1991.

VIOLA, P.; JONES, M. Rapid Object Detection using a Boosted Cascade of Simple Features. In: CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION (CVPR). **Proceedings...** [S.l.: s.n.], 2001. p.511–518.

WEBB, J.; ASHLEY, J. **Beginning Kinect Programming with the Microsoft Kinect SDK**. 1st.ed. New York: Apress, 2012.

WRIGHT, J. et al. Robust Face Recognition via Sparse Representation. In: PATTERN ANALYSIS AND MACHINE INTELLIGENCE, IEEE TRANSACTIONS ON. **Proceedings...** [S.l.: s.n.], 2009. p.210–227.

ZHOU ingcai et al. AAM based face tracking with temporal matching and face segmentation. In: COMPUTER VISION AND PATTERN RECOGNITION (CVPR), 2010 IEEE CONFERENCE ON. **Proceedings...** [S.l.: s.n.], 2010. v.701-708.