

3 Modelos Flexíveis

Em Visão Computacional, uma das principais premissas é processar uma imagem com a finalidade de compreendê-la e retirar dela informações úteis. É utilizada em diversas aplicações atualmente, como vigilância e fiscalização através de câmeras, contagem de pessoas ou veículos, controle de processos industriais e navegação para robôs.

Para isso, então, tornam-se necessários métodos para analisar e identificar, dada uma certa imagem, quais são os elementos presentes ou então apenas distinguir se um elemento buscado se encontra ou não naquela imagem. No entanto, uma barreira é rapidamente encontrada, pois dependendo do elemento em questão, ele pode apresentar diversas formas.

Sigamos o exemplo de uma aplicação para visão computacional que tem como objetivo fazer a contagem de pessoas. Pessoas possuem várias formas devido a seus atributos físicos como altura, peso, cor, podem ser de gêneros diferentes e além disso podem ter idades diferentes, fazendo com que seja possível a distinção entre crianças e adultos, por exemplo. Se considerarmos que a aplicação em questão tenha sido treinada para reconhecer silhuetas frontais de pessoas, caso a câmera não esteja perfeitamente posicionada, ou até mesmo dependendo do modo como a pessoa passe por essa câmera, a pose, postura ou o agrupamento de várias pessoas podem interferir no reconhecimento, resultando em uma aplicação falível.

Os modelos flexíveis podem representar classes de objetos que possuem formas variadas e, portanto, podem ser utilizadas para reconhecer exemplos da classe em uma imagem (8).

3.1 Active Contour Model

Active Contour Model (23), ou *snakes*, é um modelo flexível que tem a capacidade de se deformar para ajustar a certos atributos para o qual foi treinado para detectar. Ele se prende a bordas próximas, localizando um objeto de maneira precisa e pode ser utilizado para uma variada gama de desafios em

Visão Computacional como a detecção de bordas, linhas e contornos subjetivos e rastreamento de movimento.

Para que esse modelo tenha as características de deformação desejadas, três forças de atração são estabelecidas:

1. Intensidade da imagem - atrai o modelo para linhas claras ou escuras;
2. Bordas - atrai o modelo para regiões de alta frequência;
3. Terminações - atrai o modelo para fins de segmentos e cantos.

Apesar de ser um método satisfatório em certos casos, para que alcance resultados ótimos, as forças de atração devem ser definidas e refinadas manualmente para cada aplicação, reduzindo seu uso prático.

3.2

Active Shape Model

O *Active Shape Model* (ASM) é, assim como o *Active Contour Model*, um modelo flexível. Denominado por Cootes de “*Snake* inteligente” (8), tem a capacidade de refinar estimativas de pose, escala e forma de objetos em uma imagem. Ao contrário da *snake* tradicional, onde deve haver uma sintonia fina e manual das forças de atração, o ASM utiliza de um grupo de exemplos para definir suas próprias regras de deformação. É um método bastante utilizado para extração de contornos e pontos de referência de órgãos, ossos e outros componentes em imagens médicas (17) (21) (38) (39), bem como para extrair pontos e traços faciais em imagens de indivíduos, como visto nos trabalhos relacionados.

O funcionamento básico de um ASM começa com o cálculo de uma estimativa inicial de forma para um objeto em uma imagem e a iteração sobre essa estimativa para refinar seus pontos de referência, movendo-os para melhores posições, ou seja, para locais que representem o objeto mais precisamente.

Uma das maneiras de representar a forma de um objeto é através de n pontos de referência, vértices, sendo esses pontos representados na dimensão desejada. Assim, uma forma f bidimensional pode ser definida pelo seguinte vetor de $2n$ elementos:

$$f = (x_1, y_1, x_2, y_2, x_3, y_3, \dots, x_n, y_n)$$

As formas são invariantes a deslocamento, rotação e escala e o estudo estatístico de várias formas que representam um mesmo objeto possibilita a criação de um *Statistical Shape Model* (SSM), um modelo que compreende tais variações. Como exemplo, podemos tomar como objeto de estudo uma mão. A mão é capaz de apresentar diversas formas, os dedos podem estar fechados, abertos, encostados uns nos outros. No entanto, todas elas representam o mesmo objeto, a mão. A Figura 3.1 ilustra tal exemplo.



Figura 3.1: Uma mão com algumas formas de variação. Foto de (10)

Um SSM é capaz de analisar novas formas e aceitar apenas aquelas que são plausíveis com o objeto estudado. Para a criação do SSM é necessário possuir um banco de pontos de referência do objeto com as possíveis mudanças e variações que se deseja detectar.

3.2.1 Statistical Shape Model

Para a formulação do SSM, primeiramente, faz-se a Análise Generalizada de Procrustes¹ (GPA) no banco de pontos de referência, que consiste na superimposição das várias formas. Essa análise consiste em quatro etapas:

1. A escolha de uma forma de referência;
2. A superimposição de todas as formas à forma de referência;
3. O cálculo da forma média gerada pela superimposição de todas as formas e o cálculo da diferença de Procrustes entre a forma média e a de referência;

¹Procrustes é um personagem da mitologia grega, vilão, filho de Poseidon, que ajustava suas vítimas ao tamanho da sua cama. A Análise de Procrustes faz o ajuste um conjunto de dados a um outro através de transformações como rotação, translação e mudança de escala.

4. Caso a diferença entre a média e a de referência ultrapasse um limite pré-definido, diz-se que a forma média é a nova forma de referência e repete o processo.

Ao fim desse processo, as formas estão alinhadas tanto em escala quanto em rotação e translação. Essa análise também gera a forma média representante do banco de dados que será utilizada em uma etapa mais adiante.

Após o GPA, é feito o *Principal Component Analysis* (PCA) para reduzir as dimensões da forma. Isso é importante porque a forma pode ser definida por um número muito grande de pontos de referência e, para que o algoritmo tenha um bom desempenho, a redução desses pontos se torna uma tarefa necessária. O PCA é executado e a escolha do novo número de dimensões é feita de modo que uma porcentagem da variação das formas vistas seja coberta, como por exemplo, 98% do total. A escolha é feita acumulando os autovalores em ordem decrescente até o momento em que a soma ultrapassa a quantidade de variação desejada.

Com a forma média encontrada após o GPA, \bar{x} , e os autovetores associados aos autovalores selecionados ϕ (uma matriz de $2n \times r$), um vetor de deformação b (de r elementos), uma forma, x , é calculada da seguinte maneira:

$$x = \bar{x} + \phi b$$

O vetor b tem a dimensão igual ao número de autovalores escolhidos e quando todos os seus elementos tem valor igual a 0 a deformação é nula, pois a nova forma é igual a forma média.

Para que uma nova forma gerada, x , seja plausível com o SSM, é necessário restringir os possíveis valores dos elementos de b . Isto é, o valor do elemento b_i , não deve ultrapassar um fator pré-estabelecido, geralmente menor ou igual a $\pm 3\sqrt{a_i}$, sendo a_i o autovalor associado ao autovetor da posição i . Esse fator pode ser modificado para tornar o modelo mais ou menos rígido, dependendo da aplicação.

A alteração do valor dos elementos de b dentro do limite estipulado gera deformações na forma média que compreendem as deformações vistas nas formas do banco de dados com o qual o SSM foi treinado. Um exemplo da variação do vetor b e suas deformações de forma para uma mão é exibido na Figura 3.2, onde cada modo corresponde à mudança de um autovetor específico, sendo a imagem da esquerda uma variação de $-3\sqrt{a_{Modo}}$, a imagem da direita $+3\sqrt{a_{Modo}}$ e a imagem do meio demonstra nenhuma variação, um vetor b nulo.

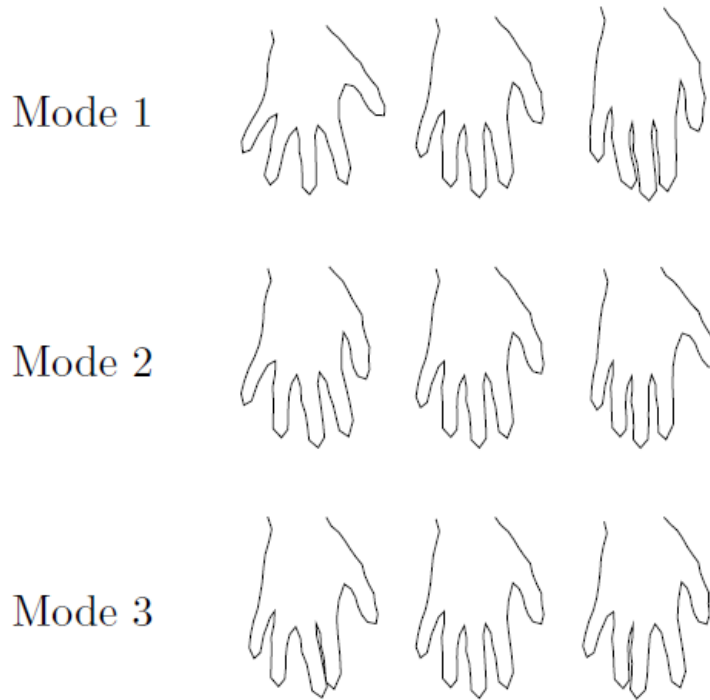


Figura 3.2: A deformação de um SSM para reconhecimento de mãos em 3 autovalores, $-3\sqrt{a_{Modo}}$ na mão da esquerda, 0 no meio e $+3\sqrt{a_{Modo}}$ na mão da direita. Foto de (10)

Dado um SSM, é possível encontrar os valores de b que fazem com que o modelo represente um nova forma não existente nos exemplos de treinamento, com um erro mínimo aceitável, desde que ela seja plausível, ou seja, se encaixe nas restrições definidas. Isso é feito através da fórmula:

$$x = T_{X_t:Y_t:s:\theta}(\bar{x} + \phi b),$$

na qual T representa uma transformação de similaridade, ou seja, um deslocamento de X_t , Y_t , uma rotação de θ e uma escala de s . No entanto, é mais fácil encontrar a transformação que faz com que a nova forma se adeque ao modelo e utilizar a inversa para adequar o modelo à forma.

3.2.2

Análise de luminância dos pontos de referência

Uma vez que existe um modelo deformável e um jeito de moldar esse modelo a novas formas plausíveis, resta descobrir, dada uma nova imagem do objeto estudado, quais são os pontos de referência que representam a forma

do objeto nessa imagem. Esses pontos podem ser encontrados de algumas maneiras, sendo uma delas através da detecção de bordas fortes. Outra maneira é através do estudo estatístico de como a luminância dos *pixels* se comporta localmente para cada ponto de referência do objeto, que será explicado a seguir (9).

Para essa análise, o banco de dados deve possuir não apenas vários exemplos de formas anotados com seus pontos de referência mas também as imagens referentes aos exemplos. A análise do comportamento de *pixels* é feita da seguinte maneira: para cada ponto e para cada imagem do banco de imagens, k *pixels* ortogonais ao ponto são selecionados. Então, para cada ponto é calculado um modelo local através da média da normalização da soma dos valores de luminância de todos os k *pixels* de cada imagem. Em outras palavras, o modelo local informa como os k *pixels* ortogonais ao ponto costumam se comportar em escala de luminância. Os pixels selecionados são ditos ortogonais porque eles devem, no caso ótimo, formar um ângulo reto com a borda da forma. A Figura 3.3 ilustra os pixels que são analisados para cada ponto de referência, indicados de cor branca. Como podemos observar na figura, a maioria dos conjuntos de pixels são ortogonais ou formam um ângulo próximo a 90 graus com a forma amarela no ponto em que a interceptam.

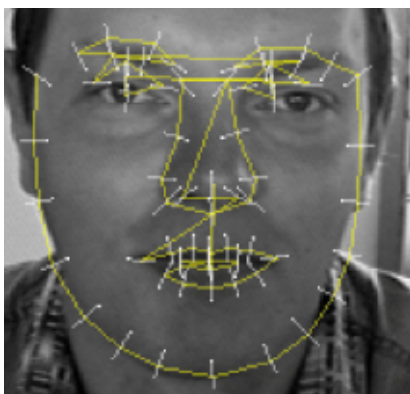


Figura 3.3: Demonstração dos pixels analisados para definição do comportamento de luminância dos pontos de referência. Foto de (30)

Com os modelos locais calculados, dada uma nova imagem com o objeto em questão, os pontos de referência que representam a forma na imagem são encontrados da seguinte maneira:

1. Uma forma inicial é definida para a imagem (etapa muito importante pois com uma forma inicial mal definida o algoritmo não converge para os pontos corretos);

2. Da forma inicial, para cada ponto são extraídos m , $m > k$, pixels ortogonais ao ponto;
3. Para cada possível subconjunto de m de tamanho k com pixels adjacentes, calcula-se a distância de Mahalanobis entre o subconjunto e o modelo local do ponto em questão. A nova posição do ponto é referente à posição que teve o menor resultado da distância de Mahalanobis.
4. Deforma o SSM para que ele represente os novos pontos de referência calculados;
5. Esse processo é feito até que o modelo tenha convergido, ou seja, enquanto a variação da forma estiver ultrapassando um limite mínimo pré-definido.

Um SSM em conjunto com um modo de encontrar pontos de referência em uma imagem configura um *Active Shape Model* (ASM). Assim, quando uma nova imagem é recebida, primeiramente é necessário encontrar os pontos de referência do objeto nessa imagem. Esses pontos são buscados da maneira descrita acima, ou seja, através do estudo do comportamento local de *pixels*. Ao encontrar esses pontos, o modelo é então ajustado para a nova forma, através da transformação de similaridade descrita anteriormente.

Para aumentar a velocidade e capacidade de convergência do algoritmo ASM, a adequação do modelo a uma imagem é feita, na verdade, sobre uma pirâmide de imagens, geralmente de três ou mais níveis. Portanto, nas imagens menores, os pontos são encontrados de maneira mais grosseira e, com o aumento da resolução das imagens, os pontos são sendo refinados para capturar os detalhes. Um exemplo de aplicação do algoritmo é exibido na Figura 3.4.

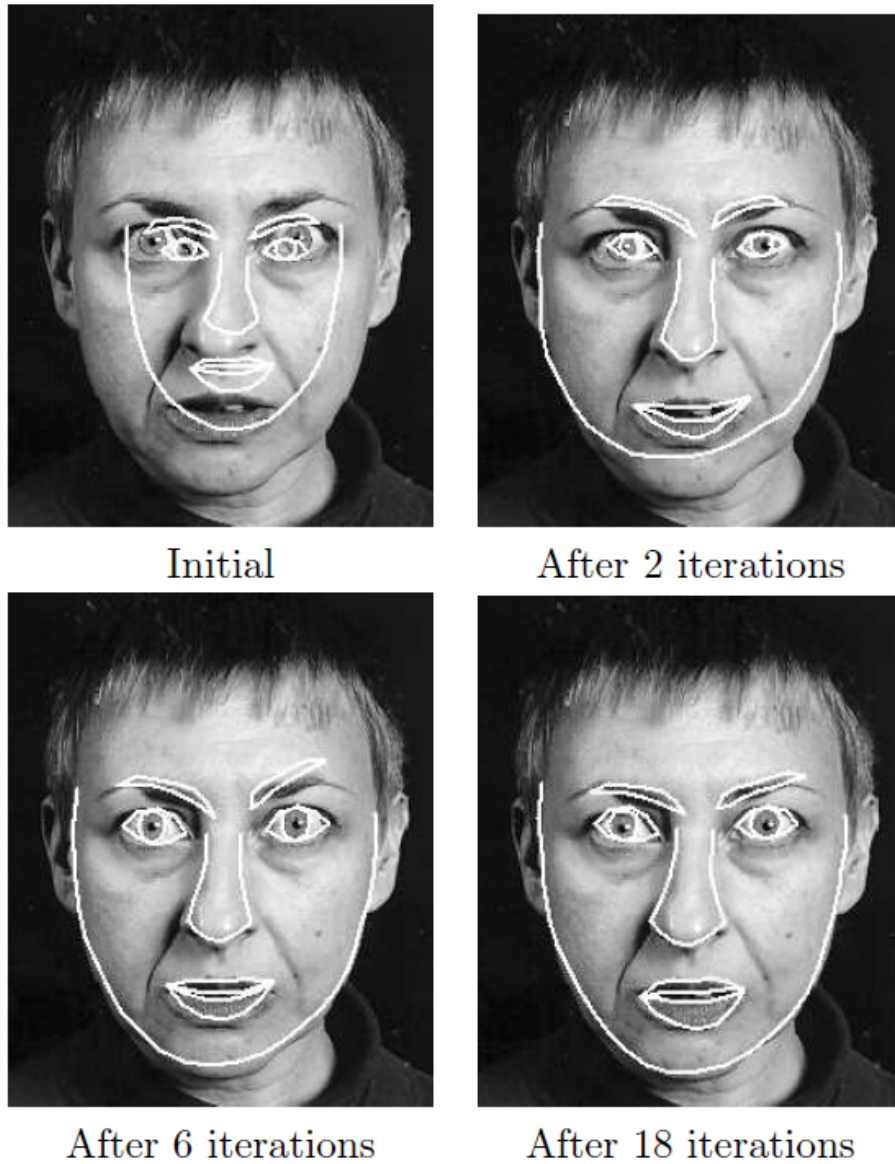


Figura 3.4: Um exemplo da execução do algoritmo ASM, desde a forma inicial até a sua décima oitava iteração. Foto de (10)