3
FERNS

Os FERNS são estruturas não-hierárquicas usadas na classificação de retalhos (patches). Os patches são pequenos pedaços da imagem. Cada FERN se baseia em um pequeno conjunto de testes binários para calcular a probabilidade de um patch pertencer a cada uma das classes existentes. Özuysal et al descrevem no artigo [18] um classificador baseado nos FERNS com o objetivo de fazer a correspondência (matching) entre os pontos característicos (keypoints) extraídos de uma imagem de treinamento e aqueles extraídos a partir de imagens obtidas, em tempo real, de diversos pontos de vista e sob diferentes condições de iluminação. A abordagem descrita no artigo [18] conta com uma fase de treinamento offline. Um exemplo, utilizando FERNS, da correspondência entre keypoints extraídos de duas imagens diferentes pode ser visto na figura 1. Este capítulo foi baseado principalmente nos trabalhos [17] e [18].

Figura 1: Matching de um objeto planar utilizando FERNS. Imagem retirada de Özuysal et al. [18].

Como mostrado em [17], modelar o problema de correspondência como um problema de classificação Naïve Bayesian produz algoritmos simples, robustos e com uma demanda computacional muito menor do que métodos propostos anteriormente. Uma descrição do problema modelado como um problema de classificação é apresentado nas seções 3.1 e 3.2. O restante do capítulo é

organizado da seguinte maneira: a seção 3.3 apresenta uma explicação geral da fase de treinamento do processo, enquanto as seções seguintes apresentam detalhes da implementação desenvolvida por Lepetit e Fua [17]. Na seção 3.4 é descrito o processo de obtenção do conjunto de treinamento, seguido, na seção 3.5, pela detecção dos pontos característicos e finalizando, na seção 3.6, com o método de seleção desses keypoints.

3.1. O problema de matching visto como um problema de classificação

Essa abordagem conta com uma fase de treinamento offline durante a qual, diversas vistas dos retalhos da imagem (image patches) centrados nos keypoints, extraídos de uma imagem de treinamento, são sintetizadas e utilizadas no treinamento do classificador. Considera-se o conjunto de todas as possíveis aparências de um retalho centrado em um keypoint como uma classe. Dessa forma, o objetivo do classificador é, a partir de um retalho associado a um keypoint, detectado em uma imagem em tempo real, definir a classe mais provável a qual esse keypoint pertence. Com isso, é possível fazer a correspondência entre os keypoints extraídos da imagem de treinamento e os keypoints detectados nas imagens obtidas em tempo real baseando-se na classificação de patches. As figuras 2 (a) e 2 (b) ilustram, respectivamente, o problema descrito acima e o conjunto de possíveis aparências de uma classe. Detalhes do método utilizado para construção das vistas de um patch centrado em um keypoint é descrito na seção 3.4

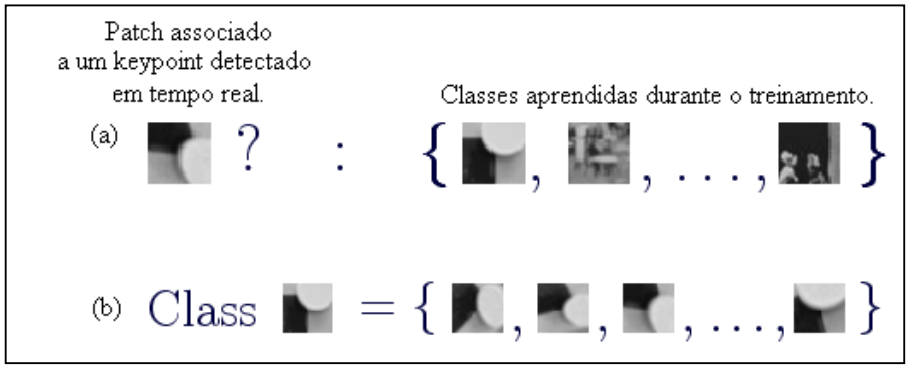

Figura 2: (a) Problema de classificação de um retalho adquirido em tempo real. (b) Conjunto de possíveis aparências de uma classe.

Para a classificação dos retalhos são utilizadas estruturas não-hierárquicas denominadas FERNS. Cada FERN é formado por um conjunto de testes binários e retorna a probabilidade de um patch pertencer a cada uma das classes aprendidas durante o treinamento. Na prática, um FERN sozinho não é suficiente para fazer essa discriminação quando há um grande número de classes envolvido. No entanto, usando-se diversos FERNS e combinando suas respostas é possível obter bons resultados na classificação. O conjunto desses FERNS treinados gera um classificador rápido e robusto, como mostrado em [18].

3.2. Classificação Naïve Bayesian

Como mencionado anteriormente, considera-se o conjunto de todas as possíveis aparências de um retalho (patch) centrado em um keypoint como uma classe. Dado o patch associado a um keypoint detectado em uma imagem, deseja-se definir a classe mais provável à qual esse keypoint pertence. Considerando c_i, i = 1, ... , H o conjunto de classes, deseja-se encontrar

$$\hat{c_i} = argmax_{c_i} P(C = c_i|patch),$$

onde C é uma variável aleatória que representa a classe.

Sendo f_j, j = 1, ... , N o conjunto de características binárias que serão calculadas sobre o patch a ser classificado, temos:

$$\hat{c_i} = argmax_{c_i} P\big(C = c_i\big|f_1,f_2,\dots,f_N\big). \quad (1)$$

O teorema de Bayes fornece

$$P\big(C = c_i\big|f_1,f_2,\dots,f_N\big) = \frac{P(f_1,f_2,\dots,f_N|C = c_i)P(C = c_i)}{P(f_1,f_2,\dots,f_N)}.$$

Uma vez que o denominador pode ser visto como sendo simplesmente um fator de escala, pois independe da classe e assumindo a probabilidade a priori P(C) como sendo uniforme, o problema se reduz a encontrar

$$\hat{c}_i = argmax_{c_i} P(f_1, f_2, \ldots, f_N | C = c_i).$$

O valor de cada característica binária f_j depende apenas das intensidades das posições $d_{j,1}$ e $d_{j,2}$ de dois pixels localizados no patch a ser classificado. Isto é,

$$f_j = \begin{cases} 1 \; se \; I(d_{j,i}) < I(d_{j,2}) \\ 0 \quad caso \; contrário \end{cases},$$

onde I representa o patch. Uma vez que essas características são extremamente simples, é necessário uma grande quantidade destas para uma classificação correta. Por conseguinte, torna-se inviável a representação completa da probabilidade conjunta dada pela equação (1), pois seria necessário calcular e armazenar 2^N entradas para cada classe. Uma maneira de comprimir essa representação seria assumir completa independência entre as características, isto é,

$$P(f_1, f_2, \ldots, f_N | C = c_i) = \prod_{j=1}^{N} P(f_j | C = c_i).$$

No entanto, essa representação ignora completamente a relação entre as características. Para tornar o problema tratável levando em conta essas dependências, o conjunto de características é particionado em M grupos de tamanho $S = {}^{N}/_{M}$. Esses grupos são denominados FERNS e para cada FERN calcula-se a probabilidade conjunta para as características. Dessa forma, a probabilidade condicional pode ser escrita por

$$P(f_1, f_2, \ldots, f_N | C = c_i) = \prod_{k=1}^{M} P(F_k | C = c_i),$$

onde $F_k = \{f_{\sigma(k,1)}, f_{\sigma(k,2)}, \ldots, f_{\sigma(k,S)}\}$, k = 1, ..., M representa O k-éssimo FERN e σ(k, j) é uma função de permutação aleatória variando de 1 a N. Desta forma, segue-se uma abordagem Semi-Naïve Bayesian para modelar apenas algumas das dependências entre as caracterticas [19]. Desta maneira, o problema

pode ser facilmente tratável, uma vez que o número de parametros é M x 2^S e uma boa taxa de reconhecimento é alcançada com M entre 30 e 50 e S em torno de 10, como mostrado em [18].

3.3. Fase de Treinamento

O processo de treinamento começa pela construção de um conjunto de H keypoints extraídos da imagem de treinamento (veja seções 3.5 e 3.6). Cada ponto característico detectado nessa imagem corresponde a uma classe diferente. As características dos FERNS, dadas pelas posições $d_{j,1}$ e $d_{j,2}$ de dois pixels localizados no patch são escolhidas aleatoriamente.

Os termos

$$P(F_k|C = c_i), k = 1, \dots, M$$

são estimados calculando-se as características binárias usando diversos exemplos para cada classe. Para ter um grande conjunto de treinamento, diversas vistas são sintetizadas a partir de uma única imagem usando técnicas de renderização como transformações afins e extraindo patches de treinamento para cada uma das classes. Nas imagens geradas automaticamente também são adicionados ruídos e filtros de suavizaçao de maneira a obter um maior realismo. Para cada keypoint da imagem de treinamento, esse processo fornece um bom conjunto de amostras das possíveis aparências tomadas de diversos pontos de vista.

No entanto, apesar de $P(F_k|C = c_i)$ ser apenas uma parte da probabilidade conjunta dada pela equação (1), o seu cálculo continua envolvendo um grande número de parâmetros e a probabilidade empírica não pode ser estimada confiavelmente como acontece na prática.

A fim de explicar como é calculado $P(F_k|C = c_i)$ considere o evento $\theta(F_k)$ que significa "A probabilidade empírica para F_k é confiável". Dessa forma, $P(F_k|C = c_i)$ pode ser expressa da seguite maneira:

$$P\left(F_k\middle|C = c_i, \theta(F_k)\right)P\left(\theta(F_k)\right) + P\left(F_k\middle|C = c_i, \overline{\theta(F_k)}\right)P\left(\overline{\theta(F_k)}\right).$$

$P(F_k|C = c_i, \theta(F_k))$ nada mais é do que a probabilidade empírica de $P(F_k|C = c_i)$, e $P(F_k|C = c_i, \overline{\theta(F_k)})$ pode ser tomada como uma constante e é portanto igual a $\frac{1}{H}$.

Considere $P(\theta(F_k))$ sendo:

$$P(\theta(F_k)) = \frac{\sum_i n_{k,i}}{\sum_i (n_{k,i} + u)}.$$

onde $n_{k,i}$ representa o número de amostras do treinamento que verificam o conjunto de características F_k. Quando o conjunto de treinamento é realmente representativo das variações que existem dentro de cada classe, esse modelo faz sentido uma vez que tende a 1 quando o número de amostras de treinamento cresce e fornece uma maneira simples de estimar $P(F_k|C = c_i, \theta(F_k))$. É fácil verificar que temos então:

$$P(F_k|C = c_i) = \frac{n_{k,i}}{\sum_k (n_{k,i} + u)}.$$

Na prática, o valor de u não influência nos resultados dado que u é maior que zero. Em [18] usa-se u = 1. Esse fator é interpretado pelos autores como Dirichlet prior, uma vez que as probabilidades condicionais das classes são modeladas como multinomiais.

3.4. Construindo o conjunto de vistas

Uma abordagem simples para a construção do conjunto de vistas é extrair keypoints da imagem de treinamento e processar cada keypoint independentemente de forma a encontrar as possíveis aparências desse keypoint. No entanto, uma abordagem mais efetiva foi usada em [17], onde, primeiramente, são geradas vistas do objeto inteiro e os keypoints são extraídos dessas vistas. Dessa forma, é mais fácil selecionar keypoints mais estáveis na presença de ruídos e sob mudanças de perspectivas, tornando o método mais robusto sem acréscimo

de custo computacional. Isso é possível, pois apenas os keypoints identificados nas diferentes imagens geradas serão considerados e aqueles que aparecem poucas vezes são descartados.

As vistas são sintetizadas utilizando-se transformações afins. Uma transformação afim pode ser decomposta em $A = R_\theta R_\emptyset^{-1} S R_\emptyset$, onde R_θ e $R_\emptyset$ são matrizes de rotação parametrizadas pelos ângulos θ e ∅, respectivamente e $S = diag[\lambda_1, \lambda_2]$ é a matriz de escala. θ e ∅ pertencem ao intervalo $[-\pi, +\pi]$ e λ_1 e λ_2 pertencem ao intervalo [0.6; 1.5]. De acordo com Lepetit e Fua [17] , essa faixa de valores é suficiente para tratar variações dentro de uma octave, enquanto variações maiores de escala são tratadas tomando-se pontos característicos detectados em diversas escalas (mais de uma octave), como mostrado na próxima seção.

3.5. Detecção de pontos característicos

Como definido em [15], uma característica local nada mais é do que uma amostra da imagem que difere da sua vizinhança imediata. Essa característica local está normalmente associada com a variação de uma ou mais propriedades da imagem simultaneamente. As propriedades da imagem que são normalmente consideradas são intensidade, cor e textura. Características locais podem ser pontos, mas também podem ser arestas ou pequenos pedaços da imagem (image patches). Tipicamente, medições são tomadas em uma região centrada em uma característica local e convertidas em descritores. Esses descritores são então utilizados em diversas aplicações como base para o reconhecimento.

Diversos métodos eficientes já foram propostos para detecção de pontos característicos em uma imagem. Com foco em aplicações em tempo real, Lepetit e Fua [16] propuseram um método rápido e estável para realização dessa tarefa. Como os próprios autores afirmam, apesar de métodos mais robustos já terem sido propostos, a baixa complexidade dessa abordagem a torna uma alternativa atrativa quando se deseja trabalhar com aplicações em tempo real. Esse método será brevemente descrito nesta seção.

Como mostrado na figura 3 (a), a idéia básica do método consiste em considerar as intensidades ao longo de um círculo centrado em cada candidato à

keypoint. Inicialmente, todos os pixels são candidatos a pontos característicos. Se dois pixels diametralmente opostos nesse círculo tiverem aproximadamente a mesma intensidade que o candidato à keypoint do centro, esse ponto não é considerado um ponto característico. Em áreas uniformes ou ao longo de arestas, sempre é possível encontrar um par de pontos diametralmente opostos com intensidades semelhantes ao ponto candidato à keypoint. Por esse motivo, o círculo é examinado realizando-se testes do tipo:

$Se\ \left|\tilde{I}(m) - \tilde{I}(m + dR_\alpha)\right| \leq +\tau$ e

$Se\ \left|\tilde{I}(m) - \tilde{I}(m - dR_\alpha)\right| \leq +\tau$ então m não é um keypoint,

onde $dR_\alpha = (Rcos\,\alpha;\ Rsin\,\alpha)$, R sendo o raio escolhido e α variando entre [0; π]. Na prática, como as imagens são discretizadas, é necessário comparar não apenas os pixels diametralmente oposto, mas também seus vizinhos, de modo a evitar respostas próximas de arestas, como mostrado na figura 3 (b). Geralmente, pontos que não são característicos são rejeitados rapidamente, sem necessidade de percorrer o círculo completo. Para os pontos a que se atribui uma resposta positiva, isto é, foram considerados como sendo pontos característicos, calcula-se um score e são selecionados como keypoints os ótimos locais. O laplaciano da gaussiana é utilizado para cálculo do score e é aproximado por um fator de escala da seguinte maneira:

$$LoG(m) \approx \sum_{\alpha \in [0;\,\pi]} \tilde{I}(m - dR_\alpha) - \tilde{I}(m) + \tilde{I}(m + dR_\alpha).$$

Essa expressão pode ser calculada à medida que o círculo é percorrido e usando apenas um número limitado de pixels, tornando o método mais rápido e mantendo sua eficácia.

Uma orientação é também atribuída ao keypoints. A orientação α_m é dada por:

$$\alpha_m = \underset{\alpha \in [0;2\pi]}{argmax} |\tilde{I}(m) - \tilde{I}(m + dR_\alpha)|.$$

De acordo com os autores, essa orientação é estável o suficiente para normalizar a vizinhança do keypoint no que diz respeito às rotações 2D. O cálculo da orientação também pode ser feito à medida que o círculo é percorrido.

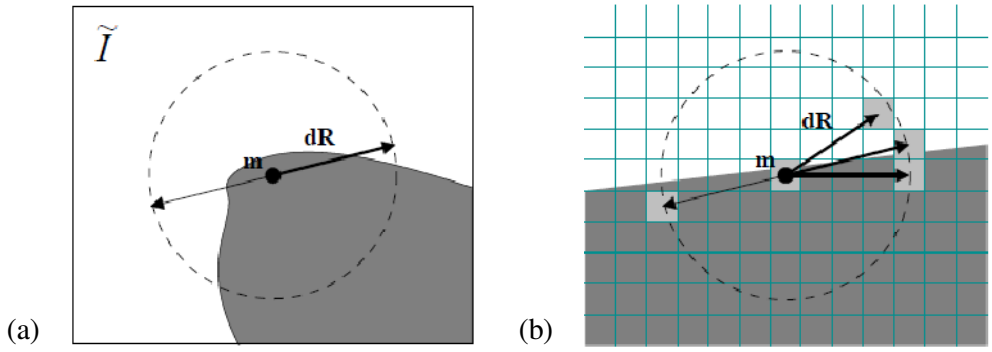

Figura 3: (a) Princípio da detecção de keypoints: se $\tilde{I}(m)$, $\tilde{I}(m+dR)$, $\tilde{I}(m-dR)$ são semelhantes, m não é um keypoint. (b) Como lidamos com imagens discretizadas, é necessário comparar os pixels opostos diametralmente e seus vizinhos para evitar respostas positivas próximas de arestas. Imagem retirada de Lepetit e Fua 2004 [16].

O algoritmo é aplicado nas primeiras octaves da imagem e os pontos característicos detectados em cada uma das octaves são utilizados para treinar o classificador. Cada octave representa a imagem em escalas diferentes, o que torna o algoritmo mais robusto em relação às variações de escala (zoom).

3.6. Selecionando os keypoints

Idealmente, um keypoint K_i em K deve possuir uma alta probabilidade $P(K_i)$ de ser encontrado quando está visível, mesmo apresentando distorções de perspectiva e ruídos.

Considere τ uma transformação geométrica utilizada para sintetizar uma nova vista como descrito na seção 3.2.2, e $\tilde{k}$ um ponto característico extraído dessa vista usando o procedimento descrito na seção 3.2.1. τ é também uma transformação afim ou projeção e aplicando τ^{-1} a $\tilde{k}$ pode-se encontrar o keypoint k correspondente no sistema de referência. No entanto, nem todos os keypoints vão ter um correspondente na nova imagem gerada. Dessa forma, dado um conjunto de vistas sintetizadas automaticamente, P(k) pode ser estimada contando-se quantas vezes o keypoint correspondente é encontrado. Dessa forma,

o conjunto K pode ser construído pegando-se os n keypoints com maior probabilidade P(k). Para cada keypoint $k_i \in K$, constrói-se o conjunto de vistas correspondentes pegando-se a vizinhança correspondente à $\tilde{k}$ nas imagens geradas, como mostrado na figura 4. Quando um keypoint é detectado em duas imagens diferentes, tanto a mudança de perspectiva quanto a existência de ruídos podem ocasionar pequenos deslocamentos em relação à sua localização exata. Para resolver esse problema, um ruído branco é atribuído nas vistas sintetizadas antes de se fazer a extração dos keypoints, o que força o classificador a aprender a lidar com esses pequenos deslocamentos.

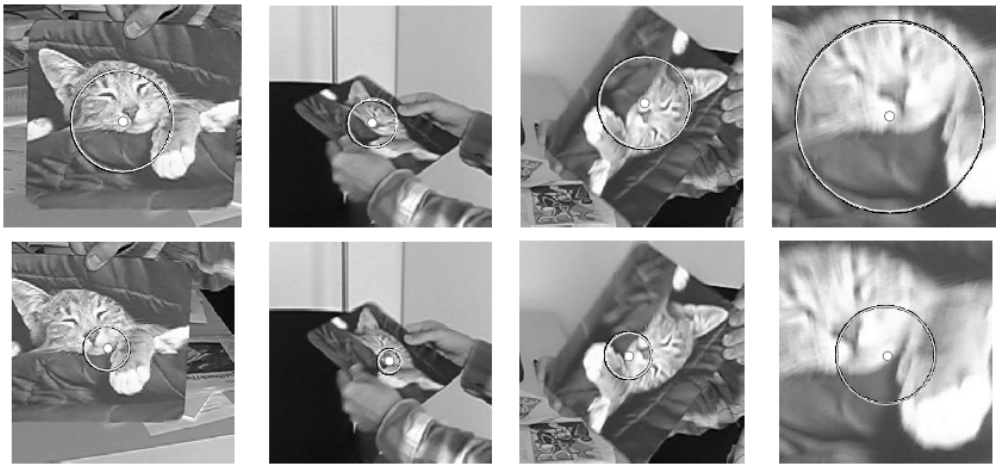

Figura 4: Linhas de cima: A imagem da esquerda corresponde a um keypoint extraído da imagem de treinamento. As próximas três imagens mostram o mesmo keypoint em diferentes vistas geradas com escalas diferentes e sob o efeito de blur. O mesmo para a linha de baixo. Imagem retirada de Lepetit e Fua [17].