

4

Proposta de Aperfeiçoamento da Técnica

4.1.

Introdução

A utilização de imagens e vídeos digitais com altas resoluções é cada vez mais comum nos diversos meios de comunicação existentes, fazendo com que o tempo de processamento para a busca de objetos em tais imagens torne-se maior devido à elevada quantidade de pixels que precisam ser analisados.

A técnica proposta inicialmente por Viola e Jones (2001) é capaz de detectar objetos de forma robusta em imagens e vídeos digitais em tempo real, porém não consegue manter a mesma velocidade de taxa de detecção em vídeos com grandes resoluções. Além disto, a técnica até então apresentada não é capaz de realizar a busca por mais de um objeto simultaneamente durante o processamento de uma imagem ou vídeo.

Neste capítulo são apresentadas algumas propostas para tornar a detecção de objetos em vídeos digitais de grande resolução em tempo real e, além disto, permitir também a busca em paralelo por mais de um objeto sem a perda de desempenho do algoritmo.

4.2.

Segmentação por Subtração de Fundo

A técnica de Segmentação por Subtração de Fundo (background segmentation) é utilizada para segmentar uma região em movimento numa seqüência de imagens a partir de uma câmera estática comparando cada novo quadro com um modelo de fundo da cena. Uma resenha detalhada das técnicas de Subtração de Fundo pode ser encontrada em Piccardi (2004) e em McIvor (2000).

Um método baseado na técnica de segmentação de fundo, normalmente, compreende as seguintes etapas: treinamento, classificação e atualização. A etapa de treinamento não é obrigatória. Ela consiste em coletar informações de um conjunto de N quadros seqüenciais para fazer uma estimativa de um modelo

de fundo inicial. Nos casos em que essa etapa é inexistente, o modelo é calculado dinamicamente nas etapas seguintes.

Para definir se um pixel x pertence ao fundo ou a um objeto em movimento é necessária uma etapa de classificação. Essa etapa é feita através do uso de um modelo de fundo previamente calculado.

Por fim, a etapa de atualização é a responsável por manter a coerência do modelo de fundo, tratando eventuais mudanças como de iluminação ou de geometria que possam ocorrer no modelo de fundo.

Algumas características essenciais são atribuídas a um método de Subtração de Fundo, sendo estas: tipo de câmera, de informação extraída, de ambiente, espectrais, regra de decisão, modelo de atualização e de representação. A câmera utilizada em um método de Subtração de Fundo pode ser simples (uma única câmera), estéreo (duas câmeras em visão estereoscópica) ou com múltiplas câmeras.

O tipo de informação extraída dos pixels pode ser espacial, temporal ou ambos. Essa informação se refere a que tipo de característica é armazenado para a atualização ou criação do modelo de fundo.

O ambiente em que o método é focado pode ser: interno, externo ou ambos. A grande maioria de métodos da literatura é focada somente para ambientes internos ou externos, é rara a existência de métodos robustos para ambos os ambientes, pois suas características são muito diferentes. Um exemplo é a iluminação. Em ambientes externos há maior influência de luz solar enquanto em ambientes internos há maior incidência de luzes artificiais.

A característica espectral de um método de Subtração de Fundo se refere ao nível de informação extraída de uma imagem baseada em pixel ou região. Além disso, é levado em consideração também o espaço de cor envolvido. Os modelos mais comuns são: RGB, YUV, HSV e em tons de cinza.

A regra de decisão é a forma com que cada pixel do modelo de fundo é classificado, sendo esta através de limiares, mapas ou redes inteligentes. O modelo de atualização, através das características espectrais armazenadas, monta o novo modelo de fundo e utiliza, por exemplo, parâmetros, limiares e funções probabilísticas para isso. A última característica é o modelo de representação, que define a técnica a ser utilizada, por exemplo, distribuição normal, mínimos e máximos e PCA.

Deve-se levar em conta também o desempenho em termos computacionais de um método de Subtração de Fundo. A maioria das aplicações tem necessidade de que a subtração ocorra em tempo real, como é o caso

discutido nesta dissertação. Então, avalia-se a quantidade de memória utilizada, a quantidade de processamento e o valor de quadros por segundo (fps) da aplicação que utiliza o método.

Na literatura existem diversos métodos para tratar a Subtração de Fundo. Estes métodos são divididos basicamente em três abordagens: a **estatística**, a **não-estatística** e as **supervisionadas**. Todos eles buscam sempre um maior custo-benefício entre o processamento e a qualidade da segmentação.

Os primeiros métodos que surgiram na literatura foram os métodos que seguem a abordagem não-estatística. A implementação desses métodos é relativamente simples, com baixo custo computacional, e o resultado da segmentação é bom para casos em que a variação de luz é mais suave e o fundo é relativamente estático. São também conhecidos como “Subtração Básica de Fundo” (Hall et al. 2005).

Dentre os diversos métodos para Subtração de Fundo existentes na literatura, propõe-se, nesta dissertação, a utilização do método básico que usa um filtro recursivo de primeira ordem definido pela equação (4-1):

$$B_i(x) = \begin{cases} I_i(x) & \text{se } i = 1 \\ (1 - \alpha)B_{i-1}(x) + \alpha I_i(x) & \text{se } i > 1 \end{cases} \quad (4-1)$$

onde a imagem de fundo $B(x)$ é atualizada de maneira acumulativa, adaptativa, integrando a informação do frame que chega $I(x)$ na imagem corrente de fundo e α é um coeficiente de adaptação. O uso de filtros recursivos de primeira ordem tem sido usado em várias técnicas nos últimos 10 anos (Wren et al., 1997) (Heikkilä e Silvén, 1999). Wren et al. (1997) usa este tipo de filtro para atualizar a média da representação estatística proposta. Muitos autores, por exemplo, Piccardi (2004), referem-se à atualização recursiva de α como sendo uma média acumulativa (Running Average ou On-line Cumulative Average) – o que termina por nomear a técnica de Wren et al. (1997) como Running Gaussian Average. Desta maneira, é usual se classificar os métodos básicos que usam a Equação (4-1) de Métodos de Média Acumulativa (Running Average Methods), mas que o autor da presente dissertação não considera um termo apropriado. Algumas vezes encontra-se a denominação do fator α como “Taxa de Aprendizagem”, o que também não é um termo recomendado pela presente dissertação.

Nos métodos básicos de Subtração de Fundo, o *background* é identificado pela seguinte equação:

$$|I(x) - B_i(x)| \leq \text{Limiar} \quad (4-2)$$

Desta maneira, há dois parâmetros de ajuste que interferem no resultado: *Limiar* (*threshold*) e α . A Figura 17 ilustra a situação do *Limiar* alto e baixo. Situação semelhante ocorre para α alto e baixo. O α define como o cenário de fundo irá se adaptar às alterações da cena ao longo dos *frames* do vídeo. Caso α tenha um valor baixo, próximo a 0, o fundo irá se adaptar muito lentamente às alterações de cena. Se α for próximo de 1, o fundo irá se adaptar mais rapidamente às alterações no cenário.

O método básico definido pelas equações (4-1) e (4-2) é muito adequado para a segmentação de vídeos em tempo real; pois é fácil e rápido de ser calculado e também é flexível a mudanças de iluminação e no próprio cenário, onde o modelo de fundo é calculado dinamicamente para cada quadro do vídeo. Ademais este método não tem problema em relação ao consumo excessivo de memória como outras técnicas de segmentação possuem.

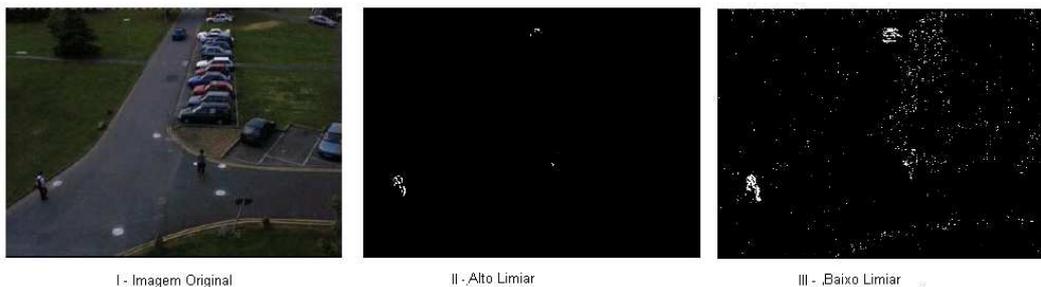


Figura 18. Exemplo de Subtração de Fundo por métodos básicos que usam a equação (4-2), para valores altos e baixos do parâmetro *Limiar*. (extraído de Piccardi (2004b))

4.3. Redução das Áreas Analisadas da Imagem

Para a realização da busca do objeto de interesse na imagem, Viola e Jones (2001) propõem o uso de uma janela de busca com a qual é feita uma varredura em toda a imagem analisando-se cada um dos possíveis tamanhos e posições desta janela de busca no interior da imagem, conforme algoritmo apresentado no Capítulo 3 desta dissertação. É proposta nesta seção uma técnica capaz de reduzir substancialmente a área a ser analisada em cada quadro do vídeo, fazendo assim com que a análise de cada novo *frame* ocorra de forma rápida e eficiente. Para atingir tal objetivo é proposta a combinação de algumas técnicas, entre elas a segmentação de vídeo, a Imagem Integral, e uma

heurística que descarte grandes regiões da imagem sem a necessidade de algum tipo de processamento.

Através da técnica de segmentação de vídeo pelo Modelo de Fundo Adaptativo, é possível distinguir as regiões de um vídeo digital que sofreram alterações ao longo de seus quadros. As áreas que contém pixels de cor branca são as áreas que sofreram alguma mudança entre os quadros da cena, e os pixels de cor preta não sofreram alterações ou foram incorporados ao fundo do cenário. Pode-se observar que quanto mais próximo de 0 for α , menor será a quantidade de pixels brancos da imagem, pois o fundo irá se adaptar mais rapidamente às alterações da cena. O caso contrário ocorrerá conforme α for mais próximo de 1, fazendo com que o fundo se adapte de forma mais lenta às variações da cena, aumentando a quantidade de pixels pretos calculados. Este comportamento é ilustrado pela figura 19, extraída a partir de experimentos com o valor de α realizados no protótipo desenvolvido durante este trabalho.

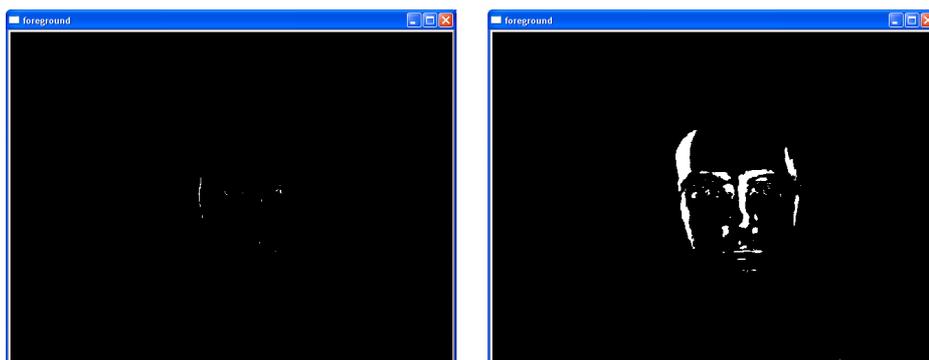


Figura 19. Exemplo de extração de plano de frente utilizando $\alpha = 0.7$ e $\alpha = 0.03$ respectivamente

Durante a execução do vídeo, apenas as regiões da imagem que sofrerem alguma alteração entre dois quadros consecutivos do vídeo precisam ter a sua área processada para detectar o surgimento de um objeto em potencial. Esta área da imagem que sofreu alterações é conhecida como *foreground* ou plano de frente, e apresenta pixels de cor branca. As demais áreas com pixels de cor preta não precisam ser processadas novamente, pois fazem parte da área de fundo da cena que já foram analisadas em um quadro anterior pelo algoritmo detector. No caso de um objeto ter sido encontrado em alguma destas regiões, elas estariam marcadas como contendo o objeto de interesse.

Conforme as características das regiões segmentadas de um vídeo citadas anteriormente, é necessário um meio de realizar o descarte de forma eficiente das regiões da imagem que não fazem parte do plano de frente da imagem, ou

seja, aquelas que não possuem pixels de cor branca, de tal maneira que quando a Janela de Busca do detector passe por uma região qualquer da imagem, a região possa ser descartada facilmente.

A técnica do cálculo da Imagem Integral pode ser utilizada com o objetivo de verificar se uma determinada região da imagem possui algum pixel que tenha sofrido alteração entre dois quadros consecutivos. Com a geração da Imagem Integral têm-se calculadas todas as áreas das regiões retangulares da imagem. Desta forma podem-se descartar regiões de quaisquer tamanhos que fazem parte do plano de fundo da cena em **tempo constante**, bastando-se obter o valor dos quatro pixels que formam os vértices da área na Imagem Integral. Se o valor da área do retângulo formado pelos quatro vértices for igual a zero é porque a área em questão possui apenas pixels de cor preta. Caso contrário, a região possui um valor de área maior que zero, significando que possui um ou mais pixels que sofreram alterações entre os quadros e, portanto, tal área deve ser analisada pelo detector.

No exemplo ilustrado pela figura 20, são analisadas duas regiões do plano de frente da imagem, formadas pelos vértices *A*, *B*, *C* e *D*, e pelos vértices *E*, *F*, *G* e *H*. Tendo-se feito o cálculo da Imagem Integral do plano de frente, ao se fazer $D + A - (B + C) = 0$, pois a área formada por tais vértices não possuem pixels brancos. Ao calcular-se $H + E - (F + G) > 0$, pois a área possui pixels brancos e devem ser analisados por pertencerem a uma área que sofreu alterações entre os quadros de vídeo e são candidatos em potencial de possuírem o objeto procurado.

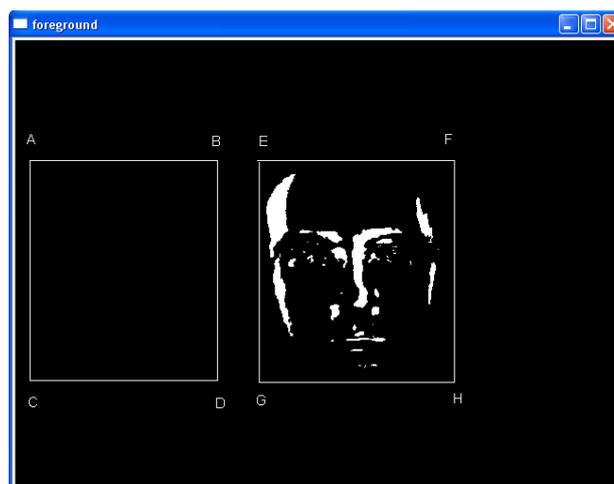


Figura 20. Definição de duas regiões retangulares sobre o plano de frente de um instante t do vídeo

Tendo-se alterado a maneira como as regiões da imagem podem ser facilmente descartadas quando a Janela de Busca estiver sobre elas, o próximo passo é diminuir a quantidade de áreas que a Janela deve percorrer na imagem.

A idéia proposta utiliza o método de Divisão e Conquista, dividindo a imagem de plano de frente em 2 sub-regiões verticais de forma recursiva, até que o tamanho da sub-região encontrada seja pelo menos igual ao tamanho mínimo da Janela de Busca na horizontal (Figura 21). Ao atingir este **caso base**, começa-se a dividir a área atual de forma horizontal, até que o tamanho mínimo da Janela de Busca na vertical seja alcançado. Para cada uma das regiões encontradas, se a o valor da área correspondente na Imagem Integral for igual a zero, a região pode ser inteiramente descartada, não sendo necessário que a Janela de Busca percorra nenhuma vez a região em questão. Ao retornar durante a recursão, as áreas de interseção entre as sub-regiões devem ser analisadas, pois o objeto pode se encontrar justamente na interseção de tais sub-regiões que contenham alguma alteração da cena. Para isto é necessário que a Janela de Busca percorra toda a nova região formada pelas 2 sub-regiões horizontais e verticais da região corrente, descartando as áreas totalmente contidas ou iguais a cada sub-região por já terem sido analisadas dentro da recursão. O pseudo-algoritmo para o descarte de áreas é descrito a seguir:

- Calcular Imagem Integral da região corrente
- Se *tamanho da região corrente* \leq *tamanho inicial da janela de busca* (**teste do caso base da recursão**) Então
 - Se *área formada pelos vértices da janela de busca na Imagem Integral* > 0 Então
 - Analisar a região corrente através da cascata de classificadores e se possuir o objeto, armazena as informações da localização atual do objeto.
 - Retornar do **caso base da Recursão**
- Subdividir a imagem da região corrente em 2 sub-regiões **verticais**;
- Se *área formada pelos vértices da sub-região esquerda na Imagem Integral* > 0 Então
 - Subdivide-se novamente a região esquerda e executa-se recursivamente o algoritmo a partir do 1º passo
- Se *área formada pelos vértices da sub-região direita na Imagem Integral* > 0 Então
 - Subdivide-se novamente a região direita e executa-se recursivamente o algoritmo a partir do 1º passo

- Subdividir a imagem da região corrente em 2 sub-regiões **horizontais**;
- Se *área formada pelos vértices da sub-região superior na Imagem Integral* > 0 Então
 - Subdivide-se novamente a região superior e executa-se recursivamente o algoritmo a partir do 1º passo
- Se *área formada pelos vértices da sub-região inferior na Imagem Integral* > 0 Então
 - Subdivide-se novamente a região inferior e executa-se recursivamente o algoritmo a partir do 1º passo
- Percorrer com a Janela de Busca toda a área formada pela soma das 2 sub-regiões verticais e horizontais, não sendo necessário testar os tamanhos menores ou iguais a cada uma das sub-regiões (pois já foram testadas na recursão) fazendo:
 - Se área corrente da Janela de Busca contém o objeto procurado Então
 - Armazena as informações da localização atual do objeto
- Retornar da Recursão
- Marcar na imagem as localizações encontradas



Figura 21. Heurística para o descarte de áreas sem necessidade de processamento

A figura 21 demonstra alguns passos da heurística proposta. Nota-se, na figura 21-II, que a primeira divisão feita na imagem descarta 50% da imagem em **tempo constante**, através da utilização da Imagem Integral, pois o valor de sua área é igual a zero. A figura 21-III mostra a segunda divisão realizada, onde como as duas novas regiões possuem valor de área superior a zero, ambas devem ser novamente divididas. A figura 21-IV apresenta a geração de todas as sub-regiões verticais e horizontais, até o momento que o último caso base é encontrado, em que o tamanho de uma nova sub-região é menor ou igual ao tamanho inicial da Janela de Busca. A linha pontilhada representa a análise que ainda é realizada pelo algoritmo ao se retornar das recursões das sub-regiões. Através do processamento das interseções das sub-regiões criadas é que o objeto desejado é encontrado, onde neste exemplo, trata-se de uma face humana.

Supondo que a Janela de Busca possua tamanho inicial igual a 35×16 , $\delta = 1,3$, $\Delta = 1$, a imagem tenha resolução de 640×480 e a área de plano de frente igual ao exemplo da figura 21, a Janela de Busca percorreria 162.642 regiões com a utilização do algoritmo de detecção proposto por Viola e Jones (2001), enquanto que ao utilizar a técnica proposta acima, seriam percorridas apenas 4.206 regiões, uma redução de 97%. Desta forma, conclui-se que quanto menos alterações existirem entre os quadros do vídeo, menos regiões serão percorridas. O efeito contrário ocorre quando houver um número maior de regiões que sofre alteração entre as cenas.

4.4. Paralelização

Geralmente durante a execução de um vídeo ou a exibição de uma imagem não há apenas um único objeto de interesse para o usuário. Pode-se citar como exemplo um vídeo onde haja a apresentação de um dado produto com fins comerciais, onde além do produto principal apresentado todos os demais objetos do cenário poderiam também estar disponibilizados para interação do usuário, de acordo com seus interesses individuais. Portanto, é preciso que em casos como estes seja realizada a detecção de vários objetos simultaneamente, otimizando-se a maneira de como tais objetos são rastreados em cada *frame* do vídeo.

O algoritmo proposto por Viola e Jones (2001) apresenta uma solução de busca por objetos de forma individual, ou seja, para cada objeto de interesse é

necessário recriar as estruturas da cascata de classificador, recalculando a imagem integral e por fim reiniciar a busca na imagem as estruturas. Neste trabalho é proposta uma solução relativamente simples, porém eficiente, que pode ser utilizada em vídeos digitais.

Inicialmente é necessário identificar no algoritmo as etapas que podem ser executadas de forma paralela, mas que são únicas para cada objeto que se deseja buscar. São elas:

- Inicialização da estrutura de cascata de classificadores em memória;
- Cálculo da existência do objeto em determinada região da imagem;
- Marcação das áreas da imagem em que existam os objetos;

Uma vez identificadas estas etapas pode-se estender o algoritmo para executá-las de forma paralela. Desta maneira com uma única análise nas regiões da imagem podem ser detectados simultaneamente todos os objetos desejados, ganhando-se em velocidade, de forma proporcional à quantidade de objetos procurados na cena. Portanto, se no algoritmo original proposto por Viola e Jones (2001) forem buscados n objetos em um vídeo, através da utilização da paralelização proposta neste trabalho a velocidade será n vezes maior. Entretanto, é preciso que haja um ambiente computacional que permita a paralelização real do algoritmo, como ambientes que possuem múltiplos núcleos de processamento ou mais de um processador.

A primeira etapa que é realizada no algoritmo de busca e detecção de objetos é a inicialização da cascata de classificadores do objeto de interesse, através da leitura das informações do disco rígido para uma estrutura em árvore na memória com o objetivo de melhorar o desempenho da análise de cada região da imagem. Este passo pode ser totalmente paralelizado com o uso de *threads* independentes para cada um dos objetos que se deseja detectar.

Durante o processamento de cada região da imagem, é possível buscar de forma paralela os n objetos desejados. Enquanto na técnica proposta por Viola e Jones (2001) é preciso que para cada objeto procurado a imagem tenha sua área novamente percorrida desde o início, é proposto neste trabalho que a região seja analisada de forma paralela para cada uma das n cascatas de classificadores em memória. Desta forma, a quantidade de objetos que pode ser processada em tempo real fica diretamente relacionada ao número de processos que podem ser rodados de maneira paralela.

Por fim, todas as regiões que possuem os objetos desejados devem ser marcadas na imagem original para que o usuário possa saber onde foi identificado algum objeto. Ao término do processamento de todas as regiões da imagem, uma estrutura é armazenada em memória que representa todas as regiões que tiveram objetos encontrados e quais foram eles. Para percorrer tal estrutura em memória e marcar a imagem original com os objetos encontrados, o algoritmo pode ser executado de forma paralela, dividindo-se o tamanho da lista pelo número de processadores existentes e percorrendo as subestruturas de maneira independente entre si.

4.5. Conclusão

Este capítulo apresenta avanços no algoritmo de detecção de objetos proposto por Viola e Jones (2001), capaz de encontrar objetos em tempo real durante a execução de vídeos digitais de alta resolução. Enquanto os demais trabalhos relacionados na literatura são focados apenas em aumentar a velocidade de detecção de objetos em imagens estáticas, é apresentada neste capítulo uma solução mais genérica que aprimora o problema de desempenho inerente à detecção em vídeos digitais em tempo real.

Através das melhorias propostas é possível reduzir o tempo de detecção de objetos ao longo de um vídeo entre 30% e 80%, dependendo principalmente da dinâmica da movimentação das cenas do vídeo. Desta forma é minimizada consideravelmente a queda da taxa de quadros por segundo (fps) que ocorre ao utilizar-se a técnica proposta por Viola e Jones (2001) com resoluções de vídeos maiores e ao se buscar mais de um objeto.

Além do ganho com a redução substancial da área em que objetos são procurados em cada *frame* do vídeo, é demonstrada a vantagem de se utilizar o paralelismo com o intuito de se conseguir aumentar a quantidade de objetos que podem ser procurados simultaneamente.