

UNIVERSIDADE FEDERAL FLUMINENSE

**THIAGO SOARES SEIDEL**

Um Método Computacional para Detecção de  
Pichação em Tempo Real

NITERÓI

2013

UNIVERSIDADE FEDERAL FLUMINENSE

**THIAGO SOARES SEIDEL**

# Um Método Computacional para Detecção de Pichação em Tempo Real

Monografia de Conclusão de Curso submetida ao Curso de Ciência da Computação da Universidade Federal Fluminense como requisito parcial para a obtenção do título de Bacharel em Ciência da Computação. Área de concentração: Processamento de Imagens.

Orientadora:  
Helena Cristina da Gama Leitão

NITERÓI

2013

# Um Método Computacional para Detecção de Pichação em Tempo Real

Thiago Soares Seidel

Monografia de Conclusão de Curso submetida ao Curso de Ciência da Computação da Universidade Federal Fluminense como requisito parcial para a obtenção do título de Bacharel em Ciência da Computação. Área de concentração: Processamento de Imagens.

Aprovada por:

---

Profa. Helena Cristina da Gama Leitão, D.Sc. UNICAMP-SP  
(Presidente)

---

Prof. Mauricio Kischinhevsky, D.Sc. PUC-Rio

---

Profa. Teresa Cristina de Aguiar, D.Sc. COPPE-UFRJ

Niterói - RJ, 15 de Março de 2013.



# Agradecimentos

*A família.*

Quero agradecer sempre e em primeiro lugar a Deus, que me deu paciência, força e sabedoria para prosseguir rumo ao alvo, sem me deixar desistir e sempre respondendo meus pedidos e orações.

Agradeço à minha família, ao Eitor Jr e à Dâmaris (meus irmãos), agradeço especialmente ao Eitor Seidel (pai) e à Universina Digetti (mãe), pois sempre me apoiaram nessa jornada e sempre me fizeram acreditar na minha capacidade.

Agradeço à Eduarda Gonçalves (minha namorada), que desde sempre me apoiou e esteve ao meu lado nos momentos mais difíceis, me fazendo ir mais além. Agradeço a ela também pela sabedoria e paciência.

Agradeço à Professora Helena pela orientação e sugestões durante esse trabalho, sem as quais eu não teria concluído o trabalho.

Agradeço aos amigos que fiz durante esse tempo de faculdade, tanto aqui na UFF, quanto aos que fiz nos tempos de UFPR.

Agradeço aos membros da banca, Professor Maurício Kischinhevsky e Professora Teresa Cristina de Aguiar, por aceitarem o convite e pelas sugestões que recebi.

# Resumo

Este projeto propõe um método para detecção de pichação em tempo real. Além de pichações o método também detecta alterações no ambiente que não sejam autorizadas. O método proposto detecta esses tipos de vandalismos através de uma sequência de eventos que levam ao reconhecimento dos atos em questão, usando uma única câmera. O vandalismo é detectado quando entra um objeto em cena e há alteração numa área predefinida. Como a questão envolve processamento de imagens, será utilizada a biblioteca openCV, que fornece algoritmos implementados para essa área.

**Palavras-chave:** Identificação de pichações, vandalismo, processamento de imagens.

# Abstract

This project proposes and implements a method for detecting graffiti in real time. Besides detecting the graffiti, the method also detects unauthorized changes in the environment. The proposed method detects these types of vandalism through a sequence of events that lead to the recognition of actions at stake, using a single camera. Vandalism is detected when an object enters the scene and some change takes place in a predefined area. As the issue involves image processing, the library is used OpenCV, which provides algorithms implemented for this area.

**Keywords:** Identification of graffiti, vandalism, image processing.

# Sumário

<b>Agradecimentos</b>	<b>iii</b>
<b>Resumo</b>	<b>iii</b>
<b>Abstract</b>	<b>v</b>
<b>Lista de Figuras</b>	<b>viii</b>
<b>1 Introdução</b>	<b>1</b>
1.1 O que é pichação? . . . . .	1
1.1.1 Crime Ambiental . . . . .	1
1.1.2 Imagens de pichações . . . . .	2
1.2 Problema da identificação de pichações . . . . .	3
1.2.1 Trabalhos relacionados . . . . .	3
1.3 Contribuição deste trabalho . . . . .	4
1.4 Estrutura da monografia . . . . .	5
<b>2 Conceitos básicos de processamento de imagens digitais</b>	<b>6</b>
2.1 Região de interesse . . . . .	6
2.2 Passos para o Processamento . . . . .	7
2.3 Aquisição de Imagens . . . . .	8
2.4 Pré-processamento . . . . .	9
2.5 Segmentação de imagem . . . . .	9
2.6 Representação e descrição . . . . .	10



---

2.7	Reconhecimento e interpretação . . . . .	11
<b>3</b>	<b>OpenCV</b>	<b>12</b>
<b>4</b>	<b>Método proposto</b>	<b>14</b>
4.1	Procedimentos realizados . . . . .	16
4.1.1	Seleção da área restrita e Seleção do quadro (frame) base . . . . .	16
4.1.2	Novo quadro (frame) . . . . .	16
4.1.3	Objeto detectado? . . . . .	16
4.1.4	Objeto na área restrita? . . . . .	17
4.1.5	Área modificada? . . . . .	17
<b>5</b>	<b>Testes e resultados</b>	<b>18</b>
5.1	Teste dos eventos: Selecciona área e Selecciona imagem base . . . . .	18
5.2	Teste do evento: Objeto detectado? . . . . .	19
5.2.1	Teste do tratamento de sombras . . . . .	20
5.3	Teste do evento: Objeto na área restrita? . . . . .	20
5.4	Teste do evento: Área modificada? e Pichação detectada . . . . .	21
<b>6</b>	<b>Conclusões e trabalhos futuros</b>	<b>23</b>
	<b>Referências</b>	<b>24</b>

# Lista de Figuras

1.1	Pichação no Relógio da Central do Brasil no Rio de Janeiro- RJ. . . . .	2
1.2	Pichação no Cristo Redentor de Belo Horizonte - MG. . . . .	2
1.3	Pichação no Cristo Redentor do Rio de Janeiro - RJ. . . . .	2
2.1	Elementos do processo de análise da imagem [9]. . . . .	8
2.2	Exemplo da etapa de pré-processamento. . . . .	9
2.3	Exemplo da etapa de segmentação. . . . .	10
2.4	Exemplo da etapa de representação e descrição. . . . .	11
4.1	Fluxograma do método proposto. . . . .	15
5.1	Imagem dos testes realizados na seção 5.1. . . . .	18
5.2	Imagem do sistema mostrando o vídeo captado. . . . .	19
5.3	Imagem dos testes realizados na detecção de objetos. . . . .	19
5.4	Imagem dos testes realizados no tratamento de sombras. . . . .	20
5.5	Imagem do objeto dentro da área restrita. . . . .	21
5.6	Imagem do vândalo cometendo pichação. . . . .	21
5.7	Imagem dos testes da identificação da pichação. . . . .	22

# Capítulo 1

## Introdução

O processamento de imagens ganha novo papel na sociedade com a popularização de equipamentos capazes de gerar, armazenar e transmitir vídeos. Dessa forma, a utilização de métodos para obter informações automaticamente de vídeos e de sequência de imagens está se tornando cada vez mais comum. As informações extraídas de um vídeo podem ser utilizadas de diversas formas como a geração de estatísticas, classificação, reconhecimento de pessoas ou objetos.

### 1.1 O que é pichação?

Pichação é o ato de escrever ou rabiscar sobre muros, fachadas de edificações, asfalto de ruas ou monumentos, usando tinta em spray aerosol, dificilmente removível, estêncil ou mesmo rolo de tinta.

No geral, são escritas frases de protesto ou insulto, assinaturas pessoais ou mesmo declarações de amor, embora a pichação seja também utilizada como forma de demarcação de territórios entre grupos - às vezes gangues rivais. Por isso difere-se do grafite, uma outra forma de inscrição ou desenho, tida no Brasil como artística.

#### 1.1.1 Crime Ambiental

No Brasil, pela Lei 9.605/98 (dos Crimes Ambientais), mais precisamente no artigo 65, a pichação é considerada vandalismo e crime ambiental e estipula pena de 3 meses a 1 ano e multa para quem pichar edificações ou monumentos urbanos.

Alguns juízes aplicam penas alternativas como por exemplo, o fornecimento de cestas básicas e prestação de serviços comunitários pelo infrator. Por outro lado, policiais em Porto

Alegre, procuram enquadrar os pichadores também em formação de quadrilha, possibilitando penas maiores.

### 1.1.2 Imagens de pichações

Nas Figuras a seguir temos imagens de pichações em monumentos e edificações.

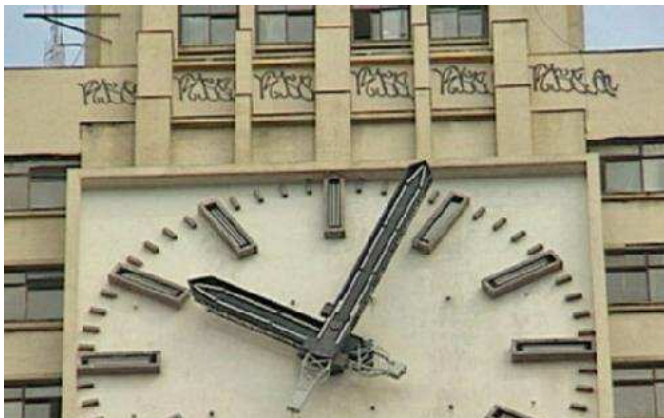


Figura 1.1: Pichação no Relógio da Central do Brasil no Rio de Janeiro- RJ.



Figura 1.2: Pichação no Cristo Redentor de Belo Horizonte - MG.



Figura 1.3: Pichação no Cristo Redentor do Rio de Janeiro - RJ.

## 1.2 Problema da identificação de pichações

A detecção automática de pichação é um procedimento muito difícil, devido a algumas situações que citaremos a seguir:

1. A falta de vídeos reais de vandalismos para teste.
2. A velocidade na qual pode ocorrer uma pichação torna a detecção mais complexa.
3. Dificuldade em discriminar atos normais de pessoas e a ação de pichadores.
4. Variação de luminosidade e o surgimento de sombras no ambiente que está sendo monitorado.
5. E ainda pode ocorrer variação das definições desse tipo de vandalismo, dependendo do contexto social do local.

Para resolver problemas relacionados ao reconhecimento de pichações, a abordagem mais comum é utilização de uma sequência de eventos que nos permitirá identificar se ocorreu um ato de vandalismo ou se tudo permanece em perfeita normalidade.

Já existem alguns trabalhos desenvolvidos relacionados com o reconhecimento de pichações. Falaremos mais sobre eles na próxima seção.

### 1.2.1 Trabalhos relacionados

Poucos métodos foram apresentados para a detecção de vandalismo. O método em [11] define um sistema semi-automático para auxiliar o operador na detecção de atos de vandalismo. A detecção é feita por meio da alimentação das características geométricas e cinemáticas a um classificador de rede neural, que é treinado para reconhecer as variações suspeitas dessas características, algumas das quais incluem as coordenadas da caixa delimitada mínima e o centro de gravidade de objetos monitorados, juntamente com suas áreas e perímetros. A dificuldade desta abordagem vem da falta da confiança da segmentação do vídeo que faz com que exista um número significativo de alarmes falsos.

O método [3] detecta vandalismo através da distinção entre objetos em movimento e expansão de objetos estáticos, que são causados por vândalos. A distinção é feita através de uma média de detecção de mudança de imagem, que tem apenas objetos estáticos e vetores de movimento. O desafio dessa abordagem é em cenas com objetos estáticos, que são muitas vezes confundidas com pichação, resultando em mais alarmes falsos.

Os autores de [6] usam múltiplas câmeras e rastreadores para construir um gráfico combinado de posições e dimensões das pessoas na cena. O gráfico combinado é então utilizado para reconhecer o vandalismo. Essa abordagem requer múltiplas câmeras e reconhecimento de objetos que aumentam o custo da aplicação e complexidade computacional.

Em [12] o trabalho apresenta um sistema automático de detecção de pichações com o uso de câmeras e algoritmos de visão inovadora. A forma humana é reconhecida através da curva da cabeça para desencadear o acompanhamento de operações para a detecção de pichações. Os experimentos deste programa são baseados em imagens reais gravadas em uma estação de ônibus. Os resultados mostram que esse sistema pode detectar ações de vandalismo como pichações na maioria das condições de iluminação e é uma solução prática para a construção de sistemas inteligentes de vigilância em tempo real.

Em [8] o projeto propõe um método para a detecção em tempo real de vandalismo em sequências de vídeo. O método proposto neste trabalho detecta vandalismo através de um algoritmo que a partir de uma sequência de eventos levam ao reconhecimento de pichações usando uma única câmera. O vandalismo é declarado quando um objeto entra em cena e faz uma alteração não autorizada no interior de uma área predefinida, como um telefone ou um sinal. O método proposto mostra resultados satisfatórios na detecção de pichação em sequências de vídeo de vigilância.

Nosso trabalho é muito semelhante aos trabalhos realizados em [12] e [8].

Na maioria dos casos, os métodos propostos necessitam de um pré-processamento da imagem antes do passo final de reconhecimento em si. Portanto, é importante contar com o auxílio de ferramentas computacionais tais como OpenCV para simplificação do problema.

### 1.3 Contribuição deste trabalho

Portanto, este projeto apresenta um algoritmo para reconhecimento de vandalismo em tempo real, mais especificadamente, pichações em muros e paredes. Utilizamos somente uma única câmera. Inicialmente, estamos preocupados em identificar esse tipo de vandalismo em ambientes onde a variação de luz possa ser controlada, ou seja, ambientes internos. Embora, seja plenamente possível expandir o sistema para a realização desse monitoramento em ambientes com variação de luminosidade. Para detecção de pichação, usamos uma imagem base (para efeito de comparação) onde não contenha objetos que possam se mover (ou possam ser retirados posteriormente). Para nos facilitar o trabalho, serão utilizadas as vantagens proporcionadas pela biblioteca OpenCV, para a divisão do vídeo em sequências de frames e também os filtros

disponíveis.

## 1.4 Estrutura da monografia

O restante desta monografia dividimos como segue. No capítulo 2, fazemos uma breve descrição de alguns conceitos básicos de processamento de imagens e no capítulo 3, falamos sobre a biblioteca OpenCV e suas vantagens. O método proposto apresentamos no capítulo 4 e os resultados encontrados discutimos no capítulo 5. Finalmente, no capítulo 6 apresentamos as conclusões e as propostas de trabalhos futuros.

## Capítulo 2

# Conceitos básicos de processamento de imagens digitais

O Processamento de imagens é uma área da tecnologia que está em pleno crescimento. Muitos dos temas científicos são abordados e em alguns casos envolvendo disciplinas de várias áreas. Entre eles podemos citar: a compreensão de imagens, a análise em multi-resolução e em multi-frequência, a análise estatística, a codificação e a transmissão de imagens, etc.

O Processamento de Imagens parte da imagem (que é geralmente captada por uma câmera) ou de uma sequência de imagens para obtenção informações que irão ajudar a determinar eventos, como em nosso caso, a determinação se ocorre ou não um ato de vandalismo. É evidente que neste sentido processar uma imagem, como é feito pelo ser humano, é extremamente complexo. Realizar as mesmas tarefas que o nosso sistema visual, com a ajuda de computadores e máquinas, exige uma compreensão "filosófica" do mundo ou dos conhecimentos humanos. Esta característica faz com que o processamento de imagens seja, atualmente, uma disciplina com extrema dependência do sistema no qual ele está associado. Logo, por este motivo, não existe até o momento, sistemas de análise de imagens, mesmo que complexos, que funcionem para todos os casos.

### 2.1 Região de interesse

Um conceito importante em processamento de imagens é a Região de Interesse (também conhecida como ROI - "Region Of Interest"). Entende-se como Região de Interesse a região definida automaticamente a partir de parâmetros obtidos na própria imagem (ou por um usuário) onde o processamento estará totalmente concentrado. Podemos, por exemplo, como em nosso caso, definir uma região que se sabe ser mais sujeita a ser pichada do que as outras.



Normalmente, com a criação da região de interesse iremos encontrar problemas no tratamento nas bordas dessa região. A solução para esse problema é dependente do contexto ao qual o processamento de imagens está associado, ao que conhecemos e podemos levar em consideração no processamento. No método que desenvolvemos não precisaremos nos preocupar com esse tipo de problema.

## 2.2 Passos para o Processamento

Processamento digital de imagens pode ser conceituado como análise e manipulação de imagens por computador [7]. O objetivo desse processo é transformar a imagem (por exemplo, aumentando o contraste, realçando bordas, etc.), de forma que a informação seja mais bem reconhecida e extrair informações da imagem que a identificarão na etapa de comparação [10].

A definição de uma imagem descrita por Gonzalez [9], consiste em uma função  $f(x,y)$ , onde  $x$  e  $y$  são coordenadas espaciais e o valor de  $f(x,Y)$  corresponde ao brilho (intensidade) da imagem nessa coordenada. Ou seja, podemos representar uma imagem através de uma matriz de pontos, onde cada ponto, representado por  $(x,y)$ , possui um valor de intensidade  $f(x,y)$ .

A menor unidade que constitui uma imagem digital é denominada pixel (picture element). Um pixel é a representação numérica da luminosidade de um ponto da imagem.

Um modelo bastante conhecido é o RGB (R-Red, G-Green, B-Blue), nesse caso a imagem é decomposta nas cores vermelho, verde e azul. Essas cores são combinadas de várias formas para reproduzir outras cores. Cada uma pode variar de 0 (zero) que é completamente escuro até (255) que é completamente intenso. Normalmente uma partição é de 8 bits, para cada cor, possibilitando uma combinação de 256 possíveis valores.

No caso do modelo RGB, um pixel irá representar um valor de intensidade para cada uma das três cores.

Tomando como base as técnicas de análise de imagem descritas por Gonzalez e Woods [9], podemos dividir o processamento em três áreas básicas: processamento de baixo nível, processamento de nível intermediário e processamento de alto nível. A Figura 2.1 mostra os processos dessas áreas.

Nos próximos tópicos apresentaremos com mais detalhes os passos para o processamento digital de imagens, as etapas abordadas neste projeto serão: Aquisição de imagens, Pré-processamento, Segmentação de imagem, Representação e descrição, e Reconhecimento e interpretação.

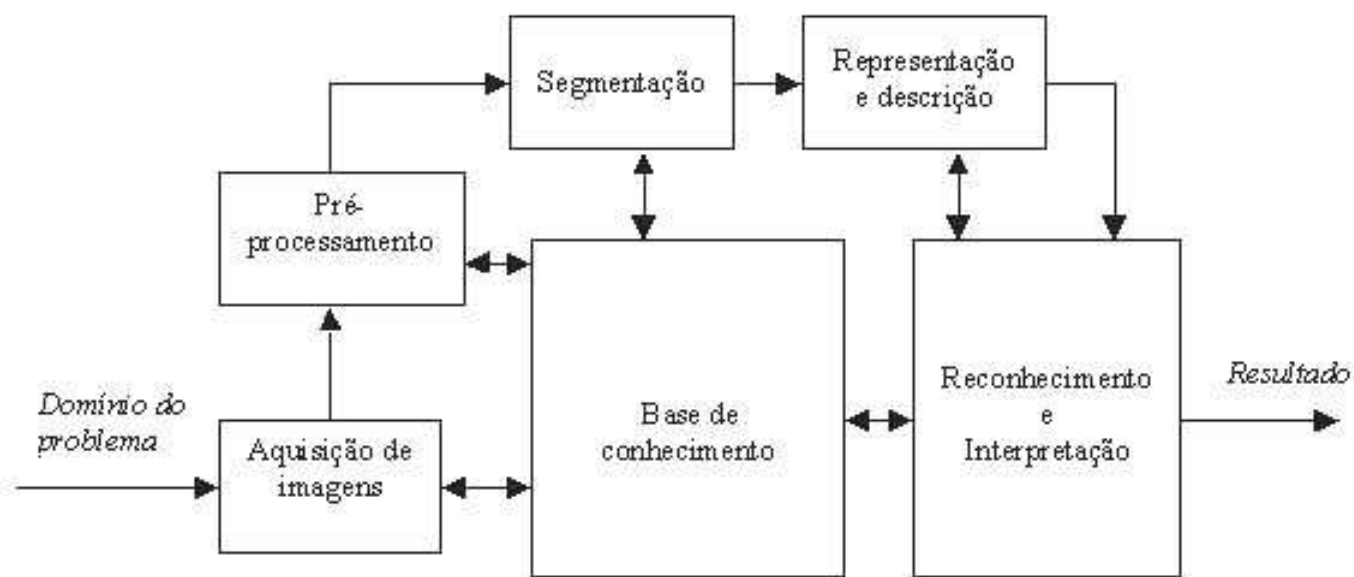


Figura 2.1: Elementos do processo de análise da imagem [9].

## 2.3 Aquisição de Imagens

A aquisição de imagens é a primeira etapa do nosso processo. Segundo Gonzalez e Woods [9], requer apenas um sensor de imagens e a capacidade para digitalizar o sinal produzido pelo sensor. Portanto, são necessários dois elementos: um aparelho físico que é sensível à faixa espectral de energia eletromagnética e que gera um sinal elétrico de saída e um digitalizador, que converte o sinal elétrico capturado na sua forma digital. Scanner, leitor de impressões digitais, câmera fotográfica ou de vídeo são exemplos de sensores que podem ser utilizados para a captura da imagem.

Em nosso caso usaremos uma câmera de vídeo para obtermos a imagem, mas é possível que ocorram alguns ruídos que possam dificultar o reconhecimento automático de pichações. Logo, a imagem ainda não é adequada para ser verificada, havendo a necessidade de aplicar algumas técnicas de realce, tornando-a mais facilmente reconhecível por uma máquina.

Na seção que segue, veremos algumas técnicas de pré-processamento que poderão ser aplicadas nas imagens.

## 2.4 Pré-processamento

Após a aquisição da imagem, o próximo passo é o pré-processamento que visa a melhorar o aspecto da imagem, aumentando as chances de sucesso dos processos seguintes [9]. Nesta etapa, são utilizadas técnicas para aumento de contraste, remoção de ruídos, realce e normalização, com o objetivo de converter os padrões para uma forma que possibilite uma simplificação do posterior processo de reconhecimento.

Em nosso caso o pré-processamento consiste em analisar a imagem capturada, onde será aplicado os tratamentos que farão com que a imagem fique em tom de cinza, assim melhorando o desempenho da aplicação. Após esse tratamento a imagem passa por um filtro para que possamos eliminar os ruídos que possam haver na imagem, assim facilitando o reconhecimento do ato de vandalismo.



(a) Imagem do palhaço com ruídos

(b) Imagem do palhaço após passar pelo filtro

Figura 2.2: Exemplo da etapa de pré-processamento.

A Figura 2.2 nos mostra um exemplo do nosso objetivo após imagem passar pelo filtro de redução de ruídos.

## 2.5 Segmentação de imagem

A segmentação é um processo complexo porque tenta traduzir para o computador um processo extremamente sofisticado realizado através da visão humana. Segmentação de imagens não é o primeiro passo no entendimento da imagem estudada, mas uma consequência.

De uma forma geral, a segmentação consiste em subdividir uma imagem de entrada em suas partes constituintes ou objetos. Cada uma destas partes é uniforme e homogênea com respeito a algumas propriedades da imagem, como por exemplo, cor e textura. Algoritmos de segmentação para imagens monocromáticas são geralmente baseados em duas propriedades básicas de valores em escala de cinza: descontinuidade e similaridade. Na descontinuidade, o particionamento da imagem é baseado no subconjunto de pontos de um objeto que o separa do restante da imagem. Na similaridade, a segmentação é baseada nas técnicas de limiarização, crescimento por regiões, união e divisão de regiões [9].

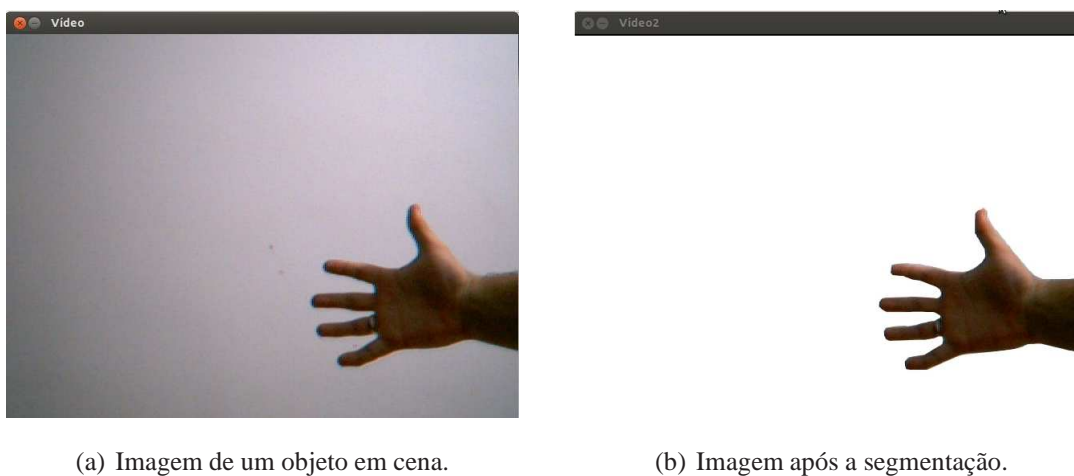


Figura 2.3: Exemplo da etapa de segmentação.

A Figura 2.3(b) representa a imagem após a segmentação, a nossa intenção é separar o objeto que entra em cena do restante da imagem.

## 2.6 Representação e descrição

Após a etapa de segmentação, onde foram separadas as áreas de interesse da imagem em regiões, é necessário representar e descrever esse novo agregado de pixels em uma forma facilmente acessível para o processamento subsequente. Existem duas formas de representação, a primeira leva em consideração características externas da região, ou seja, principalmente bordas e a outra leva em consideração as características internas, ou seja, os pixels que estão nessas regiões [7].

Após a imagem ser representada por algum método escolhido deve-se, então, descrever esta região [9]. Em nosso projeto consideramos as características internas da imagem, ou seja, sempre analisamos os pixels da nossa região de interesse.

A Figura 2.4 nos mostra como uma imagem fica representada para que tenhamos maior

facilidade no processo de reconhecimento e interpretação [2.7](#).

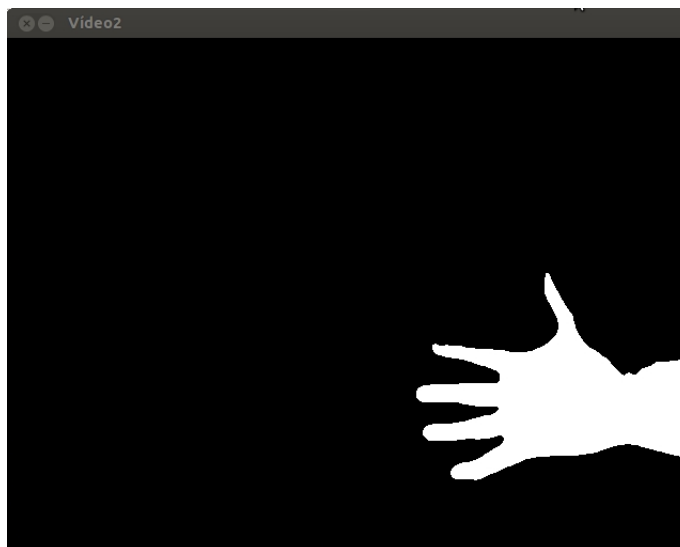


Figura 2.4: Exemplo da etapa de representação e descrição.

## 2.7 Reconhecimento e interpretação

Esta etapa está enquadrada como processamento de alto nível, pois aqui existe a responsabilidade de uma tomada de decisão por intermédio do sistema. Um dos principais objetivos da análise de imagens por computador é dotar uma máquina com a capacidade de interpretação similar a dos seres humanos [9]. Com isso é necessário utilizar algumas técnicas ou até mesmo criar novas soluções para reconhecer padrões existentes na imagem e também interpretá-las.

A Figura [5.7](#) é um exemplo que mostra que o sistema é capaz de reconhecer uma pichação.

A forma como é realizado o reconhecimento de pichações em tempo real será apresentada no capítulo [4](#).

# Capítulo 3

## OpenCV

É possível encontrar ferramentas e bibliotecas para auxiliar na execução das tarefas de reconhecimento e estimação de trajetória, discutidas anteriormente.

Desenvolvida inicialmente pela Intel Corporation, OpenCV (Open Source Computer Vision) é uma biblioteca de programação, de código aberto, que implementa uma variedade de ferramentas de interpretação de imagens. Ela foi idealizada com o objetivo de tornar a visão computacional acessível a usuários e programadores em áreas tais como a interação humano-computador em tempo real e a robótica.

O OpenCV possui módulos de Processamento de Imagens e Vídeo I/O, Estrutura de dados, Álgebra Linear, GUI Básica com sistema de janelas independentes, Controle de mouse e teclado, além de mais de 350 algoritmos de Visão Computacional como filtros de imagem, calibração de câmera, reconhecimento de objetos e análise estrutural.

Segundo Bradski e Kaehler [5], a biblioteca OpenCV está escrita em C e C++ e pode ser executada nos sistemas operacionais Linux, Windows e Mac OS X, podendo tirar proveito de processadores com múltiplos núcleos. O Anexo A apresenta os procedimentos para ajuste do ambiente OpenCV para os sistemas operacionais Linux e Windows. Existe um desenvolvimento ativo em interfaces para dar suporte a programadores das linguagens Python, Ruby, Matlab, e outras linguagens.

O pacote OpenCV está disponível gratuitamente na Internet por meio do endereço eletrônico: <http://sourceforge.net/projects/opencvlibrary>. No pacote da versão 2.4, estão disponíveis o código fonte da biblioteca, os executáveis (binários) otimizados para os processadores Intel, a documentação e exemplos em C/C++ e Python.

As principais bibliotecas do OpenCV são:

1. "cv.h": possui as principais funcionalidades de processamento de imagem e algoritmos de Visão Computacional.
2. "cvaux.h": possui algoritmos de Visão que ainda estão em fase experimental.
3. "cxcore.h": possui estruturas básicas e algoritmos, suporte a XML, funções de desenho.
4. "highgui.h": possui funções ligadas a interface de usuário, imagem e entrada e saída de vídeos.
5. "ml.h": possui métodos de aprendizagem de máquina, agrupamentos, classificação e análise de dados.

Um programa OpenCV, ao ser executado, invoca automaticamente uma DLL (Dynamic Linked Library) que detecta o tipo de processador e carrega, por sua vez, a DLL otimizada para este. Juntamente com o pacote OpenCV, é oferecida a biblioteca IPL (Image Processing Library), da qual a OpenCV depende parcialmente, além de documentação e um conjunto de códigos exemplo.

A biblioteca está dividida em cinco grupos de funções: Processamento de imagens; Análise estrutural; Análise de movimento e rastreamento de objetos; Reconhecimento de padrões e Calibração de câmera e reconstrução 3D.

Portanto, o OpenCV foi de fundamental importância para o desenvolvimento deste projeto pois nos ofereceu suporte para o tratamento de vídeos.

# Capítulo 4

## Método proposto

O método computacional que desenvolvemos para a detecção de pichação em tempo real é baseado na detecção de alterações persistentes no interior de uma região pré-determinada que estará sob vigilância. A região sob vigilância normalmente faz parte do fundo e não sofre alterações persistentes em operações normais. Antes que o sistema seja propriamente iniciado, o usuário determina a área que tenha a maior probabilidade de ocorrer uma pichação. Decidimos usar regiões retangulares para a determinação dessas áreas, pois simplifica o processo. Alterar a forma, de retangular para outra geometria não acarreta impacto sobre o algoritmo.

Depois que a área é definida, nós detectamos todas as mudanças dentro desta área em particular e decidimos se existe uma mudança (uma diferença em relação ao fundo). A mudança pode estar associada a uma utilização normal da área ou ser decorrente de um ato de pichação no local. Temos como base para comportamento normal e comportamento ilegal (pichação ou vandalismo) o seguinte pressuposto:

Objetos de fundo são deixados essencialmente no mesmo estado em que foram encontrados antes de uma normal forma de uso do objeto. Apenas o comportamento ilegal (de vandalismo) pode alterar a aparência externa de um objeto de fundo de forma significativa e permanente.

Objetos como pessoas, automóveis, animais ou qualquer outro elemento que possa se mover, nós os tratamos de igual forma, apenas como sendo objetos que entram e saem do vídeo.

O evento vandalismo é detectado através de uma sequência de eventos relacionados com a região que predefinidos e que está sob vigilância, da seguinte forma:

1. Quando um objeto é detectado no vídeo. A partir desse momento ocorre o monitoramento desse objeto e caso ele esteja dentro da região onde existe a maior probabilidade de ocorrer o vandalismo (pichação), um evento é declarado sinalizando que o objeto está



dentro dessa região predefinida. O evento declarado nesse caso, é o primeiro estágio para a detecção de um vandalismo. É desse evento que pode-se evoluir para um ato de pichação se for detectado um dano permanente (alteração visual, por exemplo) na região pré-determinada. O evento é verificado a cada novo quadro (frame), a fim de determinar se é uma ação normal, ou de vandalismo.

2. Após o objeto sair da região predefinida. Uma vez que o objeto deixou a nossa região de interesse, determinamos duas situações possíveis para monitorarmos:

- (a) Nenhuma modificação identificada dentro da região. Nesse caso o evento declarado anteriormente é convertido para um uso normal da região e o sistema retorna para seu estado inicial.
- (b) Detecção de pichação e de objeto. Nessa situação é detectada uma alteração na região pré-determinada, isso significa que há um dano permanente e foi o último objeto que esteve dentro da região que foi o responsável por esse dano. Nós declaramos que ocorreu vandalismo com o objetivo de identificar os danos e o vândalo.

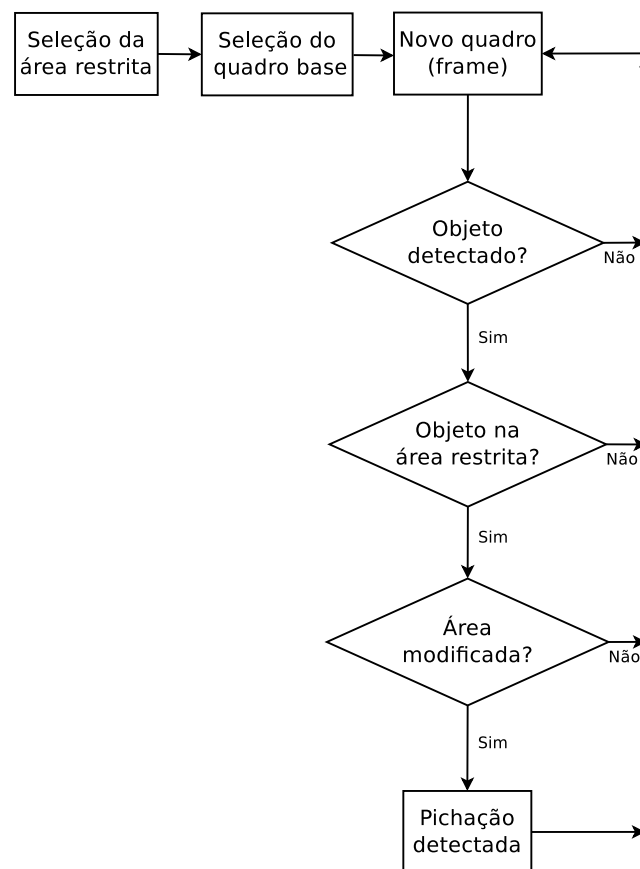


Figura 4.1: Fluxograma do método proposto.

## 4.1 Procedimentos realizados

Nesta seção faremos comentários sobre cada evento proposto no fluxograma da Figura 4.1.

### 4.1.1 Seleção da área restrita e Seleção do quadro (frame) base

Esses eventos correspondem respectivamente a seleção que o usuário irá fazer da área com maior probabilidade de que se ocorra uma pichação e posteriormente também fará a seleção do quadro base que utilizaremos para identificar se ocorreu ou não alguma modificação na área selecionada anteriormente.

### 4.1.2 Novo quadro (frame)

Neste evento é realizada apenas a captura da imagem que está sendo transmitida pela câmera de vídeo ou pelo arquivo de vídeo passado pelo usuário.

No processo de captura da imagem utilizamos a função "cvQueryFrame()" que se encontra na biblioteca "cv.h" do conjunto de bibliotecas OpenCV.

### 4.1.3 Objeto detectado?

Este evento é responsável pela detecção de objetos que estão passando pela área onde a câmera está realizando o monitoramento.

Para sabermos se um objeto está passando, precisamos de uma imagem base (iB), onde essa não contenha nenhum objeto passando no momento da captura, e a imagem atual (iA), onde esta é a imagem capturada em tempo real. Após isso, realizamos alguns procedimentos que citamos logo abaixo:

1. Primeiramente as imagens iA e iB passam por uma filtragem para remoção de algum ruído que possa influenciar no resultado do passo a seguir.
2. Após a filtragem subtraímos a imagem iA da imagem iB. Com isso, conseguimos perceber se houve variações significativas dos pixels na imagem resultante.
3. Também temos o cuidado de verificar se essas variações não são apenas sombras.
4. Caso haja variações é sinal que existe um objeto passando naquele momento. Se isso não ocorrer é por que não existe objeto nesse momento.

5. Após a detecção de que existe um objeto, passamos a monitorá-lo.

#### 4.1.4 Objeto na área restrita?

Logo que sabemos da existência de um objeto passando em nosso vídeo, passamos a rastreá-lo. Esse objeto pode perfeitamente passar pela área restrita. É importante sabermos pois esse objeto pode realizar modificações não autorizadas em nossa área de interesse.

A realização desse evento é facilitada pois já sabemos a posição onde se encontra o objeto, já que o estamos monitorando, e sabemos a posição da nossa área restrita, pois a definimos antes de começarmos o monitoramento propriamente dito. Logo, realizamos os seguintes procedimentos.

1. Buscamos saber se existe algum ponto do objeto monitorado que esteja entre os pontos da área restrita.
2. Caso isso aconteça, o objeto está dentro da área restrita, caso contrário, o objeto não está fora da área.

#### 4.1.5 Área modificada?

Se chegamos até aqui, existe a possibilidade de que ocorra um vandalismo, não significa que isso de fato venha acontecer. Para sabermos isso, temos os seguintes passos.

1. Se existe (ou existiu) um objeto dentro da área restrita.
2. Se o objeto saiu da área, tentamos identificar alguma modificação no local.
3. Para isso, utilizamos os mesmos passos que demos para detectar um objeto. A diferença é que só nos preocupamos com a área restrita.
4. Caso tenham ocorrido variações significativas nos pixels, é sinal que a área foi modificada e o ato de vandalismo é identificado, se isso não ocorrer é tratado apenas como um ato normal.

# Capítulo 5

## Testes e resultados

Neste capítulo mostraremos alguns testes realizados. Foram realizados Testes de Unidade, ou seja, testamos cada evento citado no capítulo 4 separadamente. Também foram realizados Testes de Integração, onde garantimos que todos os componentes unidos (ou combinados) estão funcionando corretamente. Mostraremos a seguir, exemplos destes Testes de Unidade e posteriormente um exemplo do Teste de Integração. Sendo que, as seções 5.1, 5.2 e 5.3 são Testes de Unidade e a seção 5.4 é referente ao Teste de Integração, pois precisamos de todas as partes funcionando em conjunto para a detecção da pichação.

### 5.1 Teste dos eventos: Selecciona área e Selecciona imagem base

Como podemos observar na Figura 5.1, realizamos a seleção da nossa área de interesse com o mouse. Logo após essa seleção, visualizamos uma nova imagem para que venhamos a selecionar a imagem base do nosso programa. Lembramos sempre que a imagem base não deve conter objetos que poderão ser retirados posteriormente.



(a) Seleccionando Área



(b) Seleccionando imagem base

Figura 5.1: Imagem dos testes realizados na seção 5.1.

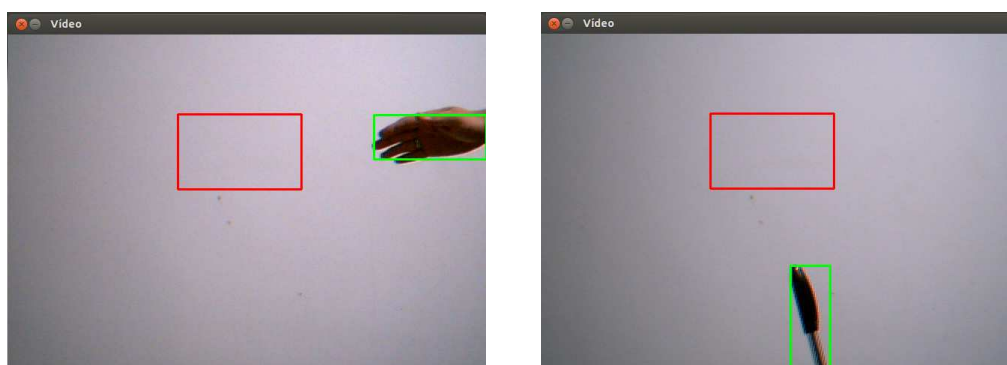
A Figura 5.2 nos mostra o passo seguinte, que consiste em nos mostrar o vídeo captado pela câmera e com a área selecionada demarcada.



Figura 5.2: Imagem do sistema mostrando o vídeo captado.

## 5.2 Teste do evento: Objeto detectado?

Observando a Figura 5.3, podemos perceber que o sistema consegue detectar objetos que entram em cena, pode ser a mão de uma pessoa como mostra a Figura 5.3(a), ou até mesmo uma simples caneta como mostra a Figura 5.3(b). Os objetos estão sendo destacados com um retângulo de cor verde ao redor deles, enquanto a área restrita encontra-se destacada com um retângulo vermelho.



(a) Identificação de uma mão como objeto.

(b) Identificação de uma caneta como objeto.

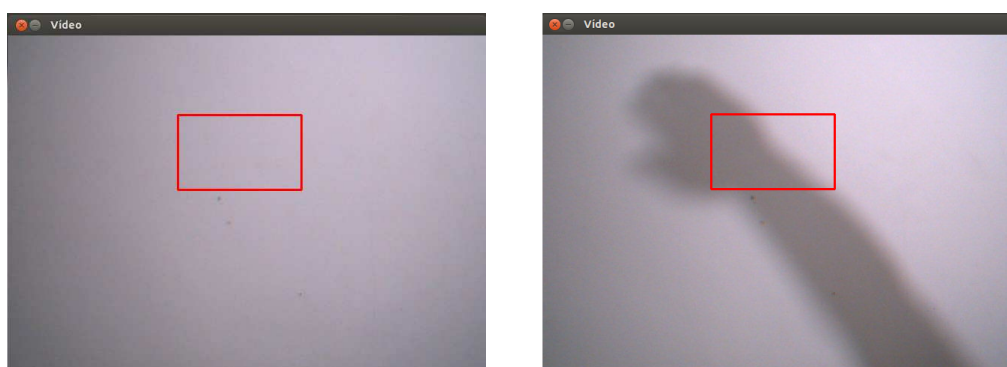
Figura 5.3: Imagem dos testes realizados na detecção de objetos.

Vale lembrar que não estamos preocupados em identificar pessoas, mas sim qualquer objeto que entre em cena.

### 5.2.1 Teste do tratamento de sombras

Esta seção foi criada pois no nosso caso uma simples sombra pode atrapalhar nossos resultados. Sombras podem ser identificadas como objetos e assim o sistema pode gerar resultados equivocados.

Para mostrar que nosso sistema consegue diferenciar sombras de objetos mostramos a Figura 5.4.



(a) Imagem sem sombras.

(b) Imagem com sombras.

Figura 5.4: Imagem dos testes realizados no tratamento de sombras.

Como podemos observar, o teste conseguiu um bom resultado, pois a sombra que se formou como mostra a Figura 5.4(b) não foi identificada como objeto pois ela não está destacada com um retângulo verde. Caso fosse identificada como objeto estaria com o retângulo como no exemplo da Figura 5.3.

### 5.3 Teste do evento: Objeto na área restrita?

Nesta seção mostramos o teste de quando um objeto entra na região da área que selecionamos como restrita. Na Figura 5.5 temos um exemplo desse acontecimento. Podemos perceber que para efeito de teste o sistema está imprimindo no terminal uma mensagem (Existe objeto e Objeto dentro da área restrita) comprovando que o teste foi bem sucedido.

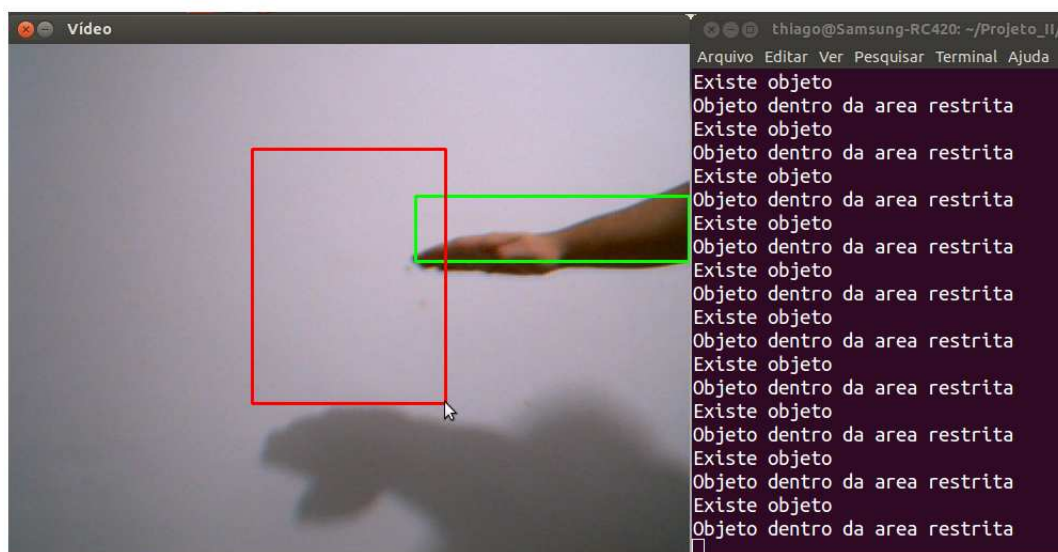


Figura 5.5: Imagem do objeto dentro da área restrita.

## 5.4 Teste do evento: Área modificada? e Pichação detectada

Agora mostraremos na Figura 5.6 que o objeto (neste caso o vândalo) que entra em cena está cometendo o ato de pichação e logo em seguida ocorre a identificação do vandalismo como mostra a Figura 5.7.

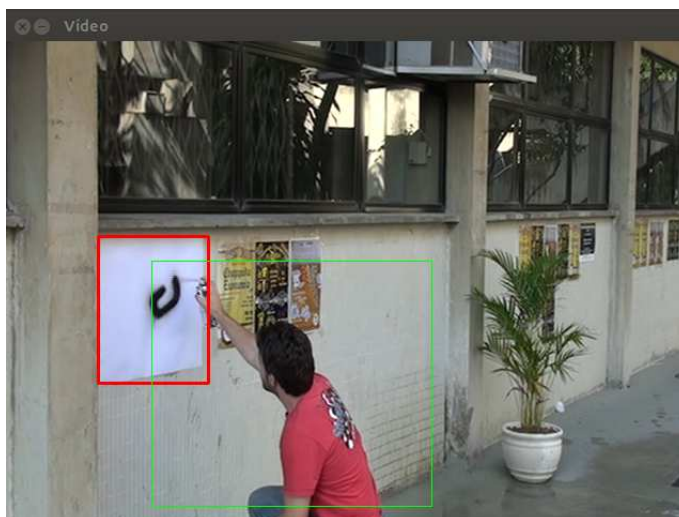
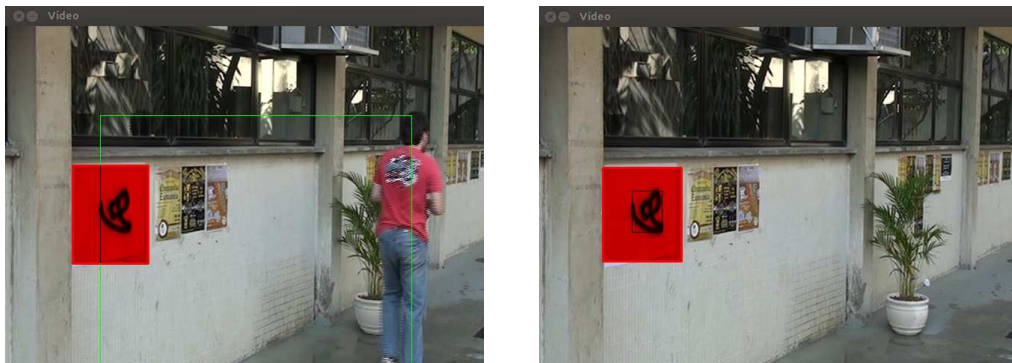


Figura 5.6: Imagem do vândalo cometendo pichação.

Na Figura 5.6 ainda não identificamos esse ato como pichação pois o objeto ainda está dentro da área restrita.



(a) Momento da identificação da pichação.

(b) Imagem com pichação identificada.

Figura 5.7: Imagem dos testes da identificação da pichação.

Na Figura 5.7(a) podemos observar que mesmo antes do vândalo sair de cena o sistema já consegue determinar que a área definida como restrita foi alterada ou, no caso, pichada (vandalizada). E na Figura 5.7(b), após o objeto sair de cena, o sistema continua acusando que existe uma alteração na área restrita.

Com a realização desse teste, terminamos o Capítulo 5. No Capítulo 6 apresentamos conclusões e perspectivas deste trabalho.



# Capítulo 6

## Conclusões e trabalhos futuros

Este trabalho propõe um algoritmo para o desenvolvimento de um sistema de detecção de pichação em tempo real. Durante a pesquisa, técnicas de processamento de imagens foram implementadas, o que melhorou a estabilidade e precisão do sistema. O sistema poderá identificar áreas recentemente pichadas e prováveis criminosos dentro das cenas de imagens coletadas pela câmera de vídeo. O vandalismo é detectado através da extração de eventos que levam a isso sem a necessidade de reconhecimento de seres humanos. O método proposto declara o ato de pichação quando um objeto entra em cena e faz uma alteração não autorizada no interior de uma área pré-definida e sujeita a atos de vandalismo. Testamos o método proposto on-line e off-line e nossos resultados mostram que ele pode detectar vandalismo e distinguir entre atos normais e vandalismo. A única limitação do sistema, dentro do escopo definido na seção 1.3 é que se houver mais de um objeto em cena, nós os tratamos como se fosse apenas um (essa limitação não interfere nos resultados).

O trabalho futuro inclui um estudo para poder aplicar o projeto a um grande conjunto de eventos vandalismo com o estudo de comportamentos humanos. Também há necessidade de desenvolvermos um método para que uma grande variação na intensidade luminosa não atrapalhe nos resultados. Ademais, é necessário desenvolvermos um procedimento para que possamos tratar mais um objeto em cena separadamente.

# Referências

- [1] Pichação. <http://pt.wikipedia.org/wiki/Picha%C3%A7%C3%A3o>, 01/03/2013.
- [2] Opencv wiki. <http://opencv.willowgarage.com/wiki/>, 17/02/2013.
- [3] ANGIATI, D., GERA, G., PIVA, S., REGAZZONI, C. A novel method for graffiti detection using change detection algorithm. *IEEE Conf. on Advanced Video and Signal Based Surveillance* (2005), 242–246.
- [4] BAXES, G. A. *Digital Image Processing*. Wiley, 1994.
- [5] BRADSKI, G., KAEHLER, A. *Learning OpenCV: Computer vision with the OpenCV library*. O'Reilly Media., 2008.
- [6] CUPILLARD, F., AVANZI, A., BREMOND, F., THONNAT, M. Video understanding for metro surveillance. *IEEE Int'l Conf. on Networking, Sensing and Control* (2004), 186–191.
- [7] FACON, J. *Processamento e análise de imagens*. PUCPR, 2002.
- [8] GHAZAL, M., V'ZQUEZ, C., AMER, A. Real-time automatic detection of vandalism behavior in video sequences. *IEEE* (2007), 1056–1060.
- [9] GONZALEZ, R. C., WOODS, R. E. *Processamento de Imagens Digitais*. Edgar Blucher, 2000.
- [10] GREIPEL, S. N. *Classificação e identificação de impressões digitais*. Faculdade de Ciência da Computação, Universidade Federal do Rio Grande do Sul. Dissertação (Mestrado)., 1996.
- [11] SACCHI, C., REGAZZONI, C., VERNAZZA, G. A neural network-based image processing system for detection of vandal acts in unmanned railway environments. *Proc. 11th Int'l Conf. on Image Analysis and Processing* (2001), 529–534.
- [12] WANG, J., XU, Z., O'GRADY, M. *Head Curve Matching and Graffiti Detection*. University of Huddersfield, 2010.