

UNIVERSIDADE SAGRADO CORAÇÃO

GABRIEL ALVES SCAVASSA

**UTILIZAÇÃO DE PROCESSAMENTO DE IMAGENS
DIGITAIS DINÂMICAS NA IDENTIFICAÇÃO DE FOCO DE
INCÊNDIO**

**Bauru
2012**

GABRIEL ALVES SCAVASSA

**UTILIZAÇÃO DE PROCESSAMENTO DE IMAGENS
DIGITAIS DINÂMICAS NA IDENTIFICAÇÃO DE FOCO
DE INCÊNDIO**

Trabalho de conclusão de curso apresentado à
Universidade Sagrado Coração, para a
obtenção do título de bacharel em Ciência da
Computação, sob orientação da Profa. Ms.
Patricia Bellin Ribeiro.

**Bauru
2012**

S288u

Scavassa, Gabriel Alves

Utilização de processamento de imagens digitais dinâmicas na identificação de foco de incêndio / Gabriel Alves Scavassa -- 2012.

38f. : il.

Orientadora : Profa. Dra. Patricia Bellin Ribeiro

Trabalho de Conclusão de Curso (Graduação em Ciência da Computação) – Universidade do Sagrado Coração – Bauru – SP.

1. Processamento de imagem. 2. Foco de incêndio. 3. Combate a incêndio. 4. Identificação de foco de incêndio. I. Ribeiro, Patricia Bellin. II. Título.

GABRIEL ALVES SCAVASSA

**UTILIZAÇÃO DE PROCESSAMENTO DE IMAGENS DIGITAIS
DINÂMICAS NA IDENTIFICAÇÃO DE FOCO DE INCÊNDIO**

Trabalho de Conclusão de Curso apresentado ao Centro de Ciências Exatas e Sociais Aplicadas da Universidade Sagrado Coração como parte dos requisitos para obtenção do título de Bacharel em Ciência da Computação, sob orientação da Prof. Ms. Patricia Bellin Ribeiro.

Banca examinadora:

Prof. Ms. Patricia Bellin Ribeiro.
Universidade Sagrado Coração

Prof. Dr. Dariel de Carvalho
Universidade Sagrado Coração

Prof. Dr. Elvio Gilberto da Silva
Universidade Sagrado Coração

Bauru, 19 de novembro de 2012.

RESUMO

Este trabalho visa desenvolver um algoritmo que consiga processar imagens geradas dinamicamente por meio de uma *webcam*, buscando por padrões de cores pré-determinadas que denominem a presença de foco de incêndio. Estas imagens são capturadas por uma *webcam* e salvas em uma pasta a cada segundo, e a cada segundo o algoritmo lê nesta pasta a imagem criada e avalia cada pixel da imagem procurando por padrões de cores, que se enquadrem nas tonalidades de fogo de chama vermelha. Para determinar que exista na imagem a presença de foco de incêndio, leva-se em consideração o número de pixels identificados, este deve ser superior a 3% do total de pixels da imagem. Quando este for menor, a imagem é descartada. Este valor mínimo é considerado para descartar a interferência de objetos com cores semelhantes no ambiente monitorado. Caso a identificação do padrão de cores ocorra em mais de uma imagem processada, faz-se a verificação de movimentação do foco de incêndio, comparando valores da última imagem processada dentro do padrão de cores com a atual imagem. Se a nova imagem tiver mais pixels encontrados, é possível que seja uma movimentação do foco de incêndio, como um alastramento, e então o algoritmo gera um arquivo contendo as informações do resultado do processamento. As imagens que forem determinadas como dentro do padrão, são salvas em uma pasta. Caso a imagem atual, não tenha um número igual ou superior ao de pixels identificados da anterior, é descartada a informação. Para a validação do algoritmo foram realizados testes com vídeos de um, três e cinco minutos. Quando somado a existência de foco de incêndio em todos os vídeos, o algoritmo conseguiu identificar 77% do vídeo deste tempo total.

Palavras chave: Processamento de imagem. Foco de incêndio. Imagens dinâmicas. Detecção de incêndio.

ABSTRACT

This course conclusion paper aims to develop an algorithm that can process images dynamically generated via a webcam, looking for patterns of colors predetermined that call the fire presence. These images are captured by a webcam and saved in a folder every second, and every second the algorithm reads the image created in this folder and evaluates each pixel in the image by looking for patterns of colors, which fit in shades of fiery red flame. To determine which image exists in the presence of an outbreak fire, it takes into account the number of identified pixels, this should be more than 3% of total pixels of the image. When it is less than 3% the image is discarded. With this minimum value being considered to discard a possible interference caused by objects with same colors in the environment in which it is monitoring. If the color's pattern identification occurs in more than a processed image, it will check the fire's movement, comparing the last processed image values within the standard colors with the current image. If the new image has more pixels found, it may be a fire drive, as a spread, and then the algorithm generates a file containing the information of the processing result. The images that are determined as in the standard, are saved in a folder. If the current image does not have an equal or greater number of the pixels than that identified above, the information is discarded. For the algorithm validation where made one minute, three and five minutes videos. When calculated all the time with the fire existed, the algorithm could recognise 77% of the video in the total.

Key words: Image processing. Fire outbreak. Dynamic images. Fire detection.

ÍNDICE DE TABELAS

Tabela 1: Valor médio das camadas, desvio e intervalos calculados com base na pesquisa levantada do banco de imagem.	26
---	----

ÍNDICE DE GRÁFICOS

Gráfico 1: Relação de quantidade de imagem capturada, imagem com foco de incêndio e imagem identificada.	34
---	----

ÍNDICE DE FIGURAS

Figura 1: Fogo em madeira.	12
Figura 2: Esquema de funcionamento em um sistema simples de processamento de imagem. Fonte: SBEGHEN (2007).....	14
Figura 3: Movimentação das chamas de um foco de incêndio.....	16
Figura 4: Exemplo de núcleos que um foco de incêndio pode conter. Onde (a) é um foco de incêndio com um núcleo, (b) um foco de incêndio com dois núcleos.....	17
Figura 5: Foco de incêndio com um núcleo.....	18
Figura 6: Foco de incêndio com vários núcleos.	18
Figura 7: Fluxograma do algoritmo.....	24
Figura 8: Estrutura do projeto.....	25
Figura 9: Armazenamento de informações da imagem antes do processamento.....	27
Figura 10: Estrutura de controle de iterações.....	27
Figura 11: Estrutura que avalia o pixel.	28
Figura 12: Cálculo da porcentagem de pixels.	28
Figura 13: Tela desenvolvida para iniciar a captura de imagem, com captura não iniciada.	29
Figura 14: Tela da interface com captura de imagem iniciada.	30
Figura 15: Imagem com foco identificado.....	31
Figura 16: Informações sobre a quantidade de pixels processados.....	32
Figura 17: Estado do computador com algoritmo não executado.	33
Figura 18: Estado do computador quando o algoritmo estava em execução.....	33

ÍNDICE

1	INTRODUÇÃO.....	10
2	FUNDAMENTOS SOBRE O PROCESSAMENTO DO FOGO	12
3	MATERIAIS E METODOS	19
3.1	Tipo de Pesquisa.....	19
3.2	Materiais.....	19
3.3	Tipo de Pesquisa.....	19
3.4	Materiais.....	19
3.4.1	Computador.....	19
3.4.2	<i>Webcam</i>	19
3.4.3	Linguagem de programação.....	19
3.4.4	Levantamento de base de dados.....	20
3.5	Procedimentos	20
4	RESULTADOS	26
5	CONSIDERAÇÕES FINAIS.....	36
5.1	Trabalhos Futuros	36
	REFERÊNCIAS.....	37

1 INTRODUÇÃO

Desde a época mesolítica, quando o homem pré-histórico tomou ciência de como produzir o fogo, os mesmos o utilizavam em quase tudo que faziam, seja no cozimento de alimentos, na proteção contra outros animais e até no próprio conforto, aquecendo-os quando necessário (SCAVASSA; VERA, RIBEIRO, 2011a).

Com o passar dos anos, esse mesmo fogo foi utilizado para movimentar máquinas, produzir bens de consumo, queimar vastas áreas de vegetação até descobrirem que ele não faz apenas coisas “boas”, e que de forma descontrolada pode mais atrapalhar do que ajudar (FERROLI, 2002).

Quando esse fogo se apresenta de forma descontrolado, seja um fogo criminoso em uma vegetação ou curto-circuito em uma rede elétrica, há sistemas que o acusam e alertam, porém esses métodos tendem a tardar para alertar (TORINO; BARRET, 2002).

A detecção de incêndio com processamento de imagem é muito estudada com o enfoque principal em floresta ou hidrelétricas usando imagens de satélites, porém, pouco estudada para ambientes internos, como o reconhecimento de incêndio dentro de indústrias, comércio, escolas, repartições entre outros (TORINO; BARRET, 2002; FIGUEIRÊDO, 2007), servindo assim, como um alarme de incêndio que identificará as chamas no início e com isso, proporcionando uma ação mais rápida dos bombeiros ou mesmo dos responsáveis pela empresa.

Processamento de Imagens e sinais é uma das técnicas utilizadas para a detecção de objetos em imagem digitais e quase sempre voltado para identificação biométrica, como digitais, face e etc. Como também em geoprocessamento, reconhecimento de placas de veículos, visão de robô, detecção de nódulos em mamografia, entre vários outros estudos realizados para o reconhecimento em imagens digitais (TORINO; ALMEIDA, 2000; TORINO; BARRET, 2002).

O tema proposto é um aprimoramento da iniciação científica, para trabalhar com processamento de imagem em tempo real na identificação de foco de incêndio.

O processamento foi para imagens que são capturadas dinamicamente por uma *webcam* e estas imagens podem ou não conter foco de incêndio. O algoritmo desenvolvido deverá reconhecer as imagens como contendo ou não presença de fogo.

Existem diversas colorações de chama para o fogo dependendo de seu material, dentre estas podendo ser azulada, esverdeada ou avermelhada. O algoritmo irá apenas reconhecer um tipo de padrão, as chamas avermelhadas.

O problema estaria em como criar um algoritmo que seja capaz de avaliar uma imagem capturada por uma *webcam* e identificar se existe ou não a presença de foco de incêndio de chama vermelha nesta imagem com baixo custo computacional?

Assim seria preciso desenvolver um algoritmo de baixo custo computacional capaz de identificar focos de incêndio com chamas de cor avermelhada em imagens dinâmicas capturadas por uma *webcam*, tornando viável sua aplicação em futuros projetos de monitoramento de incêndio e com além de :

- Pesquisar um padrão de cores das chamas de incêndio de cor avermelhada;
- Pesquisar sobre integração de captura de imagem por *webcam* com algoritmo de processamento desenvolvido;
- Desenvolver um prototipo da integração do algoritmo com a captura de imagem.

O processamento de imagem é um grande avanço para a computação, sendo muito utilizado em monitoramento e avaliações de imagens com o propósito de dar mais autonomia ao computador na tomada de decisões. Este trabalho contribui para o estudo de processamento de imagens e valores quantitativos na avaliação de imagens dinâmica, além de contribuir também para a identificação de foco de incêndio em ambientes monitorados por câmeras.

Este trabalho poderá auxiliar no monitoramento de ambientes onde não existe a possibilidade de aplicar os tradicionais métodos de identificação de incêndio seja por ineficiência ou inviabilidade.

No desenvolvimento devemos enfatizar:

- Processamento de imagens dinâmicas com recursos básicos;
- Desenvolvimento de técnica de integração de aplicações;

O projeto quando aplicado poderá auxiliar no combate de incêndio em ambientes controlados por câmeras, proporcionando rapidez no combate do foco de incêndio, evitando maiores danos.

2 FUNDAMENTOS SOBRE O PROCESSAMENTO DO FOGO

O fogo é um efeito secundário de uma matéria em transformação, por uma reação química entre o oxigênio e algum tipo de combustível (madeira, por exemplo). A madeira não pega fogo espontaneamente pelo fato de estar em um ambiente com oxigênio, então, para que a combustão ocorra é necessário aquecer este combustível até a temperatura de combustão, ou ignição de combustão, do elemento(SÖNMEZ, 2007).

No caso da madeira ela é aquecida a 150 graus Celsius. Neste momento ocorre a decomposição da celulose da madeira. Quando os gases voláteis atingem uma determinada temperatura ocorre a quebra de moléculas e os átomos se recombinaem com o oxigênio e formam outros produtos e esta reação gera muito calor.

Segundo Sonmez (2007), com este calor aumentando os átomos de carbono aumentam e emitem luz, que então nos é conferido com a chama do fogo. Este efeito é conhecido como incandescência. Neste momento a cor da chama se dá em função do que está sendo queimado e as colorações dentro da chama são causadas pela queima de diferentes materiais.

Normalmente a chama azul é a parte mais quente do fogo e as partes mais frias são laranja e amarela.

O calor da chama continua mantendo o combustível aquecido na temperatura de ignição e assim este continua a queimar enquanto existir o combustível e oxigênio . A exemplo a Figura 1 como fogo de cor avermelhada.



Figura 1: Fogo em madeira.

Segundo Sönmez (2007), devido à pressão da terra os gases quentes presentes no foco de incêndio são mais quentes e menos densos que o ar circundante, assim o fogo tem a tendência de seguir em direção para cima (subindo encostas, por exemplo). Também por este fato que as chamas têm suas pontas direcionadas para cima.

Ainda segundo Sönmez (2007), como a maioria dos métodos para detecção de incêndio convencionais não são feitos para detectar o fogo em si, mas sim seus derivados, como a fumaça que é uma consequência, um derivado do surgimento do fogo, não permite a identificação imediata do surgimento do foco de incêndio, pois, a menos que os sensores do sistema sejam atingidos pela derivação, ele não será ativado.

Outros sistemas como os baseados em infravermelho e ultravioleta são comumente usados, mas geram muitos falsos alarmes.

O processamento de imagem é uma subárea da Computação Gráfica. Esta subárea consiste na análise de informação no formato de imagem.

O Processamento de Imagens são técnicas para a análise de dados multidimensionais, que manipula imagens para obter informações que podem conduzir à tomada de decisão. O elemento principal é a imagem ou um conjunto delas, captadas através de uma câmera, de um scanner laser, de um ultrassom, de um ressonador magnético, ou por qualquer outro meio.(SBEGHEN, 2007, p. 11).

A análise de dados é uma técnica que visa identificar as propriedades dos componentes de uma imagem seja pela sua forma geométrica ou pelas suas escalas de cores.

A análise de dados junto com a técnica de processamento de imagem é muito importante para a visão computacional.

Segundo Sbeghen (2007, p.11), esta técnica implica na melhora da informação visual para a interpretação humana e processamento de dados de cenas para percepção automática através de máquinas.

O processamento digital de imagens tem como objetivo melhorar as características estruturais da imagem fornecendo subsídios para a sua interpretação, de tal forma a permitir, por exemplo, a sua utilização para o reconhecimento de ambientes, onde os algoritmos realizam operações

com o objetivo de segmentá-las e classificá-las. (SBEGHEN,2007, p. 11).

Para Sbeghen (2007, p.14), um sistema de processamento de imagem simples conta com a aquisição da imagem, que pode ser feito por qualquer meio, processamento da imagem, armazenamento e a exibição dos resultados. A Figura 2 a seguir demonstra este processo em um sistema simples.



Figura 2: Esquema de funcionamento em um sistema simples de processamento de imagem. Fonte: SBEGHEN (2007)

A aquisição de imagem para um *software* de processamento de imagem pode ser feito com imagens estáticas, carregadas para o algoritmo de processamento ou com uma captura de imagens dinâmicas. Estas capturas dinâmicas usam sensores (câmeras) que transformam os frames de vídeo em imagens, que assim são passadas para o algoritmo de processamento (SCAVASSA; RIBEIRO, 2012 a).

Alguns exemplos de módulos de aquisição são: as câmeras, os scanners ou ainda alguns sensores dedicados (como sistemas de infravermelho, mapeamento magnético de uma superfície, sensoriamento de radares, etc). Sendo que a câmera de vídeo é o sensor de aquisição de imagem mais frequentemente utilizado (SBEGHEN, 2007, p. 15).

Para Sbeghen (2007, p.12) o número de sensores (câmeras) e seu tipo que são usadas para pesquisas com quantificação e projeção de ambiente, como por exemplo cartografia e mapas 3D, para a criação destes ou averiguação, existe um grande volume de dados para se processar que então geraria um alto custo computacional.

Para sistemas que contam com um sensor (câmera) este problema de custo operacional não é um problema, pois esta é uma solução econômica, mas que assim tem suas desvantagens como o limite de informação sobre profundidade, o que não seria ideal para sistemas onde esta informação é importante.

Ainda Sbeghen (2007), também resalta importantes considerações sobre estes sistemas de processamento de imagens onde, em alguns casos, a complexidade do algoritmo de processamento sendo alta, poderá gerar problemas com tempo de resposta devido ao tempo que este levaria para processar uma imagem.

O problema com custo computacional também pode não só estar na relação do processamento da imagem em si, mas também em todo o processo de captura e avaliação.

Como exemplo, temos a alta demanda de informações que são submetidas ao sistema de visão, em função da relação existente entre um sistema de decisão em tempo real, com um sistema de leitura de imagens que opera em uma determinada taxa de captação/leitura, isto é, quantidade de imagens geradas por segundo (SBEGHEN, 2007, p. 13).

Segundo Sbeghen (2007, p.15) as imagens são descritas espacialmente por coordenadas em um plano cartesiano, ou seja, em x e y. Também é denotada a sua luminescência que pode ser colorida ou em tons de cinza

Uma câmera de vídeo apresenta características de sensibilidade, resolução e função de transferência (entre a imagem captada da cena e a intensidade do sinal de vídeo fornecido). Algumas câmeras mais sofisticadas permitem ainda controlar o "offset" ou ganho do sinal de vídeo, possibilitando uma correção do contraste da imagem em tempo real. Em alguns casos podemos encontrar módulos que permitem o ajuste da iluminação de fundo ou ainda realçar os contornos dos objetos através da execução de funções específicas. (SBEGHEN, 2007, p.15).

Para Sönmez (2007), com a ajuda de técnicas de visão de máquina que por meio de imagens adquiridas dinamicamente (capturada por câmeras) e de

processamento de imagem, podemos melhorar os resultados de informação sobre a detecção de fogo, pois podem fornecer informações mais confiáveis.

Segundo Scavassa e Ribeiro (2012 b), o fogo possui um arranjo de cores de acordo com seu tipo de combustível, como a cor azulada, quando este combustível é algum gás. Por meio desta característica, podemos utilizá-la para a identificação do corpo do fogo, destacando este do resto do ambiente.

Usando *softwares* que fazem edições de imagens, conseguimos selecionar pixels dentro do corpo do foco de incêndio, para saber qual o valor RGB (Sistema *Red*, *Green* e *Blue*). Para isso, é criado um banco de imagem e pesquisado neste banco utilizando a técnica de extração de cores das chamas, assim, é possível construir uma tabela com os valores desta pesquisa e assim calcular a média destes valores (RGB) para então usá-los como um fator de determinação para a identificação de fogo (SCAVASSA; VERA; RIBEIRO, 2011 b).

Segundo Phillips III, Shah e Lobo (2002), para a identificação do fogo não é suficiente à identificação deste pela coloração.

Apesar de ser feito um levantamento muito específico para a coloração da chama do fogo, outras coisas podem partilhar do mesmo padrão de cores encontrado. Para distinguir um objeto que tem a coloração parecida com a encontrada pelos padrões do fogo, temos a natureza de seu movimento. Como o fogo tem uma movimentação natural, então suas chamas, em dois quadros de imagens diferentes estariam localizados em lugares distintos. Como por exemplo, na Figura 3.



Figura 3: Movimentação das chamas de um foco de incêndio.

Ainda segundo Phillips III, Shah e Lobo (2002) realizaram um trabalho onde a base de pesquisa sobre a coloração seria usada no processamento da imagem

baseada em modelos onde os valores eram retirados manualmente da imagem e usados para a descrição deles, como cor de fogo.

Neste trabalho a avaliação de imagens passa por uma classificação para distinguir as possíveis imagens com foco de incêndio. Estas imagens que são avaliadas como, identificadas com fogo, são comparadas com a próxima imagem a ser processada, assim ajustando a técnica de detecção de movimento.

Liu e Ahuja (2004) desenvolveram um trabalho com propósitos de avaliação de imagens baseadas em fogo com análise espectral, espacial e temporal. O trabalho mostrou a relação entre estes três pontos, onde o espectral representado na coloração das chamas do fogo, o espacial como qual a região que o fogo é encontrado e o temporal como a variação dos espacial e espectral durante o tempo de captura de imagens.

Pois para eles a identificação da geometria do fogo seria importante para identificar vários focos de incêndio. Essa identificação seria possível determinando o número de núcleos de foco, que tem características de coloração mais próximas ao branco ou transparente, que é diferente do resto do fogo que as cores tendem para o vermelho.

A Figura 4 ilustra o número de núcleos que um foco de incêndio pode ter.

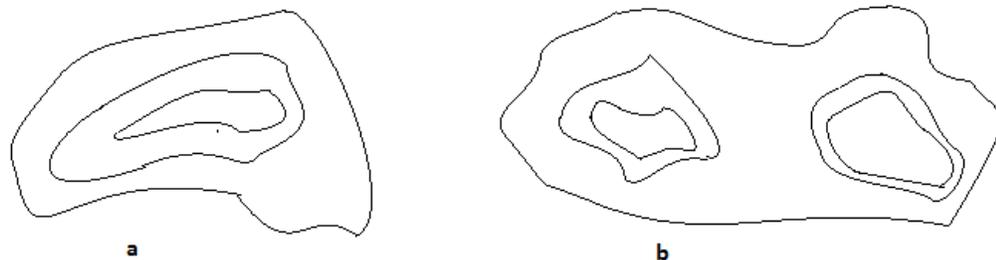


Figura 4: Exemplo de núcleos que um foco de incêndio pode conter. Onde (a) é um foco de incêndio com um núcleo, (b) um foco de incêndio com dois núcleos.

A Figura 5 representa um foco de incêndio com um núcleo e a Figura 6 representa um foco de incêndio com vários núcleos.



Figura 5: Foco de incêndio com um núcleo.



Figura 6: Foco de incêndio com vários núcleos.

3 MATERIAIS E METODOS

3.1 Tipo de Pesquisa

Segundo Gil(1999, p.43), o método de estudo deste projeto é definido como exploratório pois, desenvolvemos uma pesquisa sobre processamento de imagens dinâmicas com integração com *webcam* para monitoração de foco de incêndio.

3.2 Materiais

3.3 Tipo de Pesquisa

Segundo GIL(1999, p.43), o método de estudo deste projeto é definido como exploratório pois, desenvolvemos uma pesquisa sobre processamento de imagens dinâmicas com integração com *webcam* para monitoração de foco de incêndio.

3.4 Materiais

3.4.1 Computador

O computador em que foi desenvolvido o algoritmo usa o sistema operacional Windows 7 Ultimate de 64bits. Conta com um processador AMD Phenom IIX4 965 Processor 3.40GHz. A placa de vídeo usada é a NVIDIA GeForce7025. A memória RAM instalada é de 4 GB e com um disco de armazenamento (HD) de 500GB.

3.4.2 Webcam

A Câmera escolhida para o trabalho é uma *Webcam* Multilaser WC040 da marca Multilaser. A webcam tem resolução para filmes de 1.3 mp para vídeos, conta com sistema plug & play. Pode ser conectada a qualquer computador por não precisar de drive de instalação.

3.4.3 Linguagem de programação

A linguagem de programação utilizada no projeto foi a JAVA. E para a interação com o usuário a API Swing. Esta é uma API de interface completa para aplicações Java que renderiza (desenhas) seus próprios componentes.

Para a captura de imagens usamos e JMF (*Java Media Frame*) que é uma biblioteca que permite trabalhar com a captura de imagem e som. O ambiente de desenvolvimento integrado (IDE) foi o NetBeans.

3.4.4 Levantamento de base de dados

Para a coleta de valores de cada camada RGB de cada imagem do banco de imagens, utilizamos o programa Adobe Photoshop CS4 e para realizar a média e o desvio padrão, o programa Excel do pacote Office, na versão 2007. As imagens do banco são do formato JPEG, com tamanho padrão de 640x480.

3.5 Procedimentos

Na primeira etapa foram coletadas imagens que continham foco de incêndio onde as cores das chamas eram de tons avermelhados. Estas imagens eram do formato JPEG com tamanho de 640 X 480 pixels ou 307.200 pixels. O tamanho do foco de incêndio nas imagens não tinham o tamanho menor que 3% do total da imagem, em torno de 92.160 pixels, para evitar interferências, tendo a qualidade de 75 DPI (“*Dot Per Inch*” Pontos por Polegadas) e intensidade de 42 bits. No total foram selecionadas vinte imagens para formar o banco de imagens. Usando estas imagens, foi possível estimar valores base, que foram utilizados na pesquisa, pois as tonalidades de vermelho não foram tão discrepantes devido ao tipo de combustão. Todas estas imagens foram pesquisadas na Internet, evitando o uso de imagens com direitos autorais restritos.

A segunda parte consistiu em criar uma tabela no Excel e utilizou o *software* Photoshop para capturar valores de imagem. No Excel a tabela criada contém as colunas “R”, “G”, “B”, “Média”, “Desvio padrão”, “Intervalo 1” e “Intervalo 2”. Estas colunas foram preenchidas no decorrer da coleta de dados. Com o Photoshop CS4

em cada imagem do banco de imagens para coletar valores das camadas RGB dos pixels dentro do corpo do sinistro.

Para a coleta, foi utilizada a ferramenta conta-gotas, do *software* Photoshop CS4, que permitiu selecionar um determinado ponto dentro da imagem e identificar o valor de cada camada RGB deste. Partes aleatórias dentro do corpo do sinistro foram selecionadas. Estas partes representam um pixel dentro de toda uma área da imagem e assim foi possível coletar os valores das camadas RGB deste pixel. Para cada imagem, após esta coleta de valores, em uma planilha do Excel, foi anotado o valor coletado da camada, e assim as colunas referentes a cada camada foram preenchidas. Onde, a coluna “R” recebeu o valor referente a camada R (*red*) do RGB, em seguida a coluna “G” com o valor referente a camada G (*green*) do RGB e por ultimo a coluna “B” recebeu o valor referente a camada B (*blue*) do RGB. Após realizar este procedimento com cada imagem do banco de imagens, uma média de valores para cada uma das camadas foi calculado e informado na coluna “Média”. Também foi calculado o desvio padrão, anotado na coluna “ Desvio padrão”, podendo assim, criar um valor máximo e um valor mínimo médio para cada camada. Ao final estes valores foram utilizados na estrutura condicional do algoritmo de identificação.

A terceira parte do projeto consistiu em desenvolver uma ferramenta com a API JFM (*Java Media Frame*) para capturar vídeos. Esta API permitiu que uma *webcam* instalada no computador, capturasse vídeos tornando possível a manipulação deste. Com isto foi criada uma captura de imagem do vídeo. Esta captura é feita quando o programa estava marcado como inicializado e só parando a captura de imagem quando este for marcado como parado ou ele for encerrado pelo usuário.

Através de métodos herdados desta API foi criado um *software* que foi inicializado pelo usuário por meio de uma *interface*, onde é possível executar a função de captura de vídeo. Este *software* foi capaz de capturar vídeos criados pela *webcam* e separar em frames, com tamanho de 640X480, que consiste em uma imagem estática, e salvar este em uma pasta. Esta pasta é compartilhada para o algoritmo também, criando assim um elo entre a captura e o processamento de imagem.

A quarta parte do desenvolvimento consistiu na elaboração do algoritmo para a identificação do foco segundo os valores médios pesquisados. Para toda imagem

antes de iniciar o processo de avaliação, usamos uma função que retornou a quantidade de pixels existentes nela e armazenamos em uma variável para futura comparação de valores. Na avaliação da imagem, percorremos cada pixel dela.

O algoritmo percorreu todos os pixels da imagem usando uma estrutura de controle de iterações chamada "FOR", que forçou a passagem dos pixels pela avaliação enquanto o número de pixels processados não se igualou ao número de pixels existentes na imagem. A avaliação da condição de existência de um foco de incêndio foi feita por meio de uma estrutura seletiva encadeada formada por três condições, que são referentes a cada camada do RGB. Este processo se inicia após a criação de um *frame* de imagem na pasta compartilhada entre o *software* de captura e o algoritmo de processamento. No primeiro momento é feita uma leitura da imagem e por meio de uma função já existente, é obtido a quantidade de *pixels* da imagem e este é salvo em uma variável. Após isto o processamento da imagem se inicia.

O primeiro *pixel* da imagem entra na estrutura de condições para avaliação. Na primeira condição avalia-se a camada R (*red*) do RGB do pixel atual é igual ao valor máximo ou igual ao valor mínimo da camada R (*red*) pesquisada, ou ele se está entre eles. Quando esta condição é verdadeira, a interação continua e passa para a próxima condição de verificação que será a avaliação da camada G (*green*) do RGB do pixel atual, que verifica se este é igual ao valor máximo ou igual ao valor mínimo da camada G (*green*) pesquisada, ou se ele está entre o valor máximo e o valor mínimo da camada G. Caso esta condição seja verdadeira, a interação passa para a terceira e última condição.

A última condição verifica se a camada B (*blue*) do RGB do pixel atual é igual ao valor máximo ou igual ao valor mínimo da camada B (*blue*) pesquisada, ou se ele está entre o valor máximo e o valor mínimo. Se todas estas três condições forem verdadeiras, é somado 1 a um contador de interações, que é usado para definir que existe um pixel dentro do padrão de cores.

Caso o pixel não se enquadre no padrão, independente se não foi enquadrado dentro da condição de valor da camada R(*red*) ou G(*green*) ou B(*blue*), este será finalizado e o processo se dará para o próximo pixel. Este processo se dá até que todos os pixels da imagem sejam avaliados.

No final do processamento, se o número de vezes que foram encontrados pixels com cores padrões for igual ou superior a 3% do número de pixels da

imagem, esta imagem é salva em uma pasta de imagens com padrão encontrado, e uma variável recebe a quantidade de pixels encontrados como dentro do padrão, para uma futura análise e uma variável será marcada como verdadeira, indicando que já existiu uma imagem identificada com foco.

Se a imagem processada não se enquadrar na porcentagem mínima, ela é descartada e a próxima imagem é lida. Se esta próxima imagem também for identificada dentro dos padrões, e a variável que controla se já existe uma imagem identificada no padrão for verdadeira, será feita uma comparação de valores da quantidade de pixels encontrados na imagem anterior e na imagem processada atual. Se a quantidade for igual, existe então uma possibilidade de ser uma imagem estática que surgiu, e esta imagem será descartada. Se esta imagem tiver um número menor de pixels, esta será descartada, pois poderá ser uma interferência causada por um objeto em movimento ou pessoa. Se for maior, é gerado um arquivo texto detalhando hora, data, quantidade de pixels encontrados e as imagens são salvas, indicando que foi encontrado um possível foco de incêndio, pois o comportamento de um incêndio é se alastrar, conseqüentemente, aumentando de tamanho.

Após o processamento, avaliamos se a variável que controla se a aplicação ainda está inicializada está com o valor “*true*” (verdadeiro), se estiver, o processamento continua, caso esteja com o valor “*false*” (falso) esta será encerrada.

O padrão estabelecido em 3% do número de pixels da imagem foi estabelecido para evitar que objetos pequenos ou detalhes no fundo não atrapalhem no processamento.

A quinta parte foi desenvolver uma *interface* para o usuário. Esta tela conta com um botão de inicialização um botão de finalização e uma área para o exibir o vídeo do ambiente que esta sendo monitorado. Esta tela foi criada usando a biblioteca Swing do Java e a JMF. O botão de inicialização faz com que toda a estrutura de captura e processamento inicie. No momento que o usuário clica no botão, uma variável recebe o valor “*true*” (verdadeiro). Enquanto esta variável estiver marcada como “*true*” o software executara a captura e o processamento da imagem normalmente. O botão de finalização faz com que o *software* encerre. No momento que o usuário clicar neste botão, a variável antes com o valor de “*true*” recebe o valor de “*false*” (falso).

Neste momento, a captura de imagem e o processamento desta, serão encerrados. Quando o algoritmo enviar a imagem e a mensagem de que foi identificada uma imagem com coloração padrão de fogo, é criada uma imagem destacando o foco de incêndio, e um arquivo é gerado contendo informações de hora, data e quantidade de pixels identificado.

A Figura 7 é a representação em fluxograma do modo como o algoritmo trabalha.

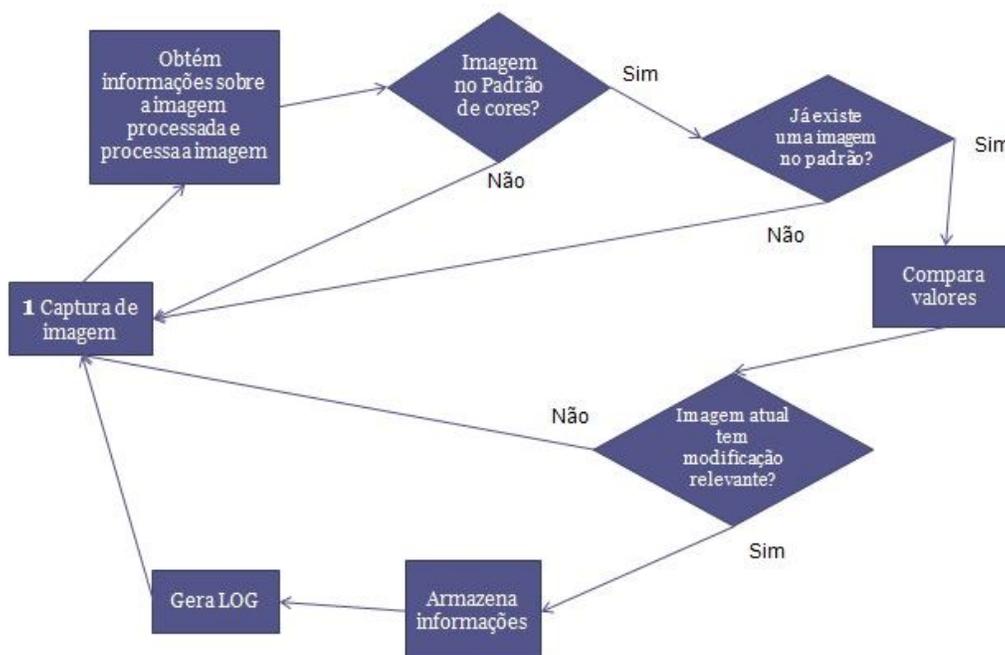


Figura 7: Fluxograma do algoritmo.

A sexta parte consistiu no teste da aplicação. Nesta etapa foram feitos vídeos com tempos de duração de um minuto, três minutos e cinco minutos. Os vídeos foram gravados em ambientes diversos, sem a existência de chamas de cor avermelhada. Estes testes serão para constatar que independente do ambiente não existirá uma interferência relativa na avaliação dos resultados, que aponte dados falsos.

Para os testes foram gravados vídeos em ambientes diversos onde criamos chamas em momentos aleatórios durante a gravação. Estes testes foram para avaliar que a chama foi identificada não importando o momento que ela foi criada durante a gravação do vídeo. O fogo deve ser criado a uma distancia entre um metro

e três metros. Esta distancia ajudará na prevenção de falsos alarmes quanto a grandeza do fogo. Quando a imagem da chama estiver muito perto, podendo atrapalhar na classificação devido à qualidade da coloração que a câmera captura e se estiver muito longe que o tamanho da chama não fique pequeno para que o algoritmo não o exclua como uma interferência na imagem. A criação do fogo para o teste será controlado.

No final, reestruturamos o projeto dividindo em arquivos para cada especialidade. Com esta estruturação, foi possível manter arquivos sem afetar funcionalidades diretamente em outros. A estrutura final do projeto pode ser vista na Figura 8, que mostra a separação dos arquivos de criação de imagem resultante “CreateRGBImage.java”, o código de inicio “Main.java”, o controle de execução “NovaThread.java” o gerador de execuções repetidas “ProdutorThread.java” e a captura de imagens “SnapFrame.java”.

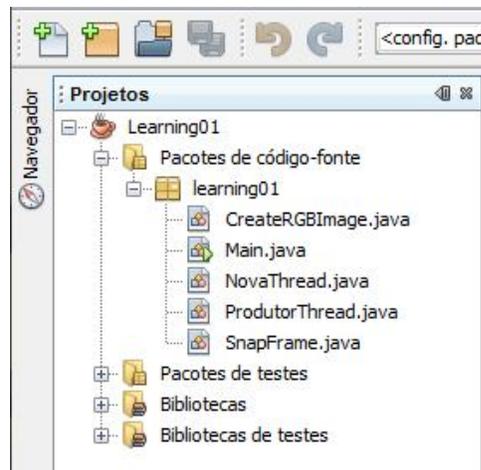


Figura 8: Estrutura do projeto.

Ao final destes procedimentos pudemos executar o projeto, testar e aprimorar cada ponto e finalizá-lo.

4 RESULTADOS

O levantamento de imagens para o banco foi refinado para usar apenas imagens com intensidade 42 bits e também com a qualidade de 75 DPI (“*Dot Per Inch*”, pontos por polegadas). Esta questão precisou ser levantada pois a qualidade das imagens deve ser mantida para todas, minimizando as discrepâncias a fim de melhorar a média dos valores.

Os valores obtidos na Tabela 1 é o resultado da pesquisa de valores com estas imagens do banco de imagens.

Tabela 1: Valor médio das camadas, desvio e intervalos calculados com base na pesquisa levantada do banco de imagem.

CANAL	Média Simples	Desvio	Intervalo 1	Intervalo 2
Red	247	6	255	242
Green	193	52	236	131
Blue	93	50	145	44

A captura de imagens para o processamento, foi feita pela *webcam* e foi adquirida pela JFM, que constroi esta imagem capturada em um array de bits, assim é possível o trabalho com as informações. A qualidade de captura de imagem da webcam influencia em todo o processamento, pois se a webcam não possuir uma boa qualidade de cores, DPI e pixels, pode por em risco o processamento por não se adequar aos parametros usados na pesquisa.

A etapa seguinte a captura, é o processamento da imagem. Toda a etapa de processamento inicia com o armazenamento de informações sobre a imagem.

A estrutura pode ser vista na Figura 9.

```

ImageIO.write(image, "JPG", new File("rbgpattern" + i + ".jpg"));

int tipo;
int a = 0, b = 0;
fileImage = new File("rbgpattern" + i + ".jpg");
// Abrindo Imagem
imagem = ImageIO.read(fileImage);
Raster raster = imagem.getRaster();
WritableRaster rasterout2 = imagem.getRaster();

int w = imagem.getWidth();
int h = imagem.getHeight();

```

Figura 9: Armazenamento de informações da imagem antes do processamento.

Após esta etapa de armazenamento, a imagem passa para a avaliação, que conta com uma estrutura encadeada para percorrer todos os pixels da imagem. O resultado desta pode ser vista na Figura 10 a baixo.

```

//Estrutura para percorrer todos os pixels
for (h = 0; h < imagem.getHeight(); h++) {
    for (w = 0; w < imagem.getWidth(); w++) {

```

Figura 10: Estrutura de controle de iterações.

Dentro desta estrutura de controle ficam as condições de avaliação que verificam as camadas RGB dos pixels da imagem.

Quando o pixel está sendo processado, a primeira condição é comparar se a camada R(*red*) do pixel atual é igual aos intervalos da camada R pesquisada na primeira fase, ou se está entre estes intervalos. Quando esta condição é verdadeira, a próxima condição é a avaliação da camada G(*green*).

O mesmo processo se dá avaliando se a camada G do pixel atual é igual aos intervalos da camada G pesquisado na primeira fase ou se está entre estes intervalos. Quando este valor se enquadra, a camada B(*blue*) do pixel é avaliada. Se este estiver entre os intervalos ou se este for igual aos intervalos a um deles pesquisados para a camada B na primeira fase, este pixel é marcado com a cor azul e é somado um para uma variável que conta o número de pixels encontrados dentro do padrão, simbolizando que este pixel estava dentro do padrão.

Quando uma condição não é atendida, o pixel é recriado na imagem de saída com a cor verde, e é somado um para a variável que controla o número de pixels fora do padrão encontrado.

A função “*raster.getSample(w,h,0)*” representa o pixel na posição W, que é referente a posição horizontal do pixel atual na imagem, na posição H, que é referente à posição vertical do pixel, e a propriedade 0, que representa a letra R da camada RGB, e compara se este é igual ou menor que o valor 255 que simboliza o intervalo pesquisado na primeira fase que refere a um tom de vermelho ou se é igual ou maior ao valor 242 que simboliza um tom de vermelho pesquisado também na primeira fase. Abaixo dela temos a mesma sintaxe, porém, com os parâmetros 1 e 2 na função “*raster.getSample()*” que simbolizam a camada G e a B respectivamente, com a comparação de seus determinados intervalos.

A Figura 11 mostra o trecho do código que avalia as camadas de acordo com os intervalos referentes a cada camada.

```

if ((raster.getSample(w, h, 0) <= 255) && (raster.getSample(w, h, 0) >= 242)) {
    if ((raster.getSample(w, h, 1) <= 255) && (raster.getSample(w, h, 1) >= 121)) {
        if ((raster.getSample(w, h, 2) <= 144) && (raster.getSample(w, h, 2) >= 4)) {
            //Conta o número de pixels dentro do padrão
            a++;
            //Pinta a imagem de saída com Azul
            rasterout.setSample(w, h, 0, 0);
            rasterout.setSample(w, h, 1, 0);
            rasterout.setSample(w, h, 2, 255);
        }
    }
}

```

Figura 11: Estrutura que avalia o pixel.

A função “*rasterout.setSample()*” é quem cria a imagem de saída(um arquivo de imagem) mostrando onde é encontrado fogo e onde não é encontrado.

Após o processamento contabilizamos a porcentagem de pixels detectados em relação a quantidade de pixels da imagem. A Figura 12 podemos ver o trecho de código que faz esta conta.

```

double percent = b * 0.03;
int percent_int = 0;
percent_int = (int) percent;

```

Figura 12:Calculo da porcentagem de pixels.

Depois de processada, se enquadrada ela é armazenada e o mesmo processo ocorre e existe uma comparação com a próxima imagem avaliada.

Assim ficou a estrutura de avaliação que procura por valores crescentes entre as imagens que são avaliadas.

A tela para a interação do usuário, foi desenvolvida a partir do conteúdo já existente na API JMF (*Java Media Frame*). Como não existe muitas possibilidades de alteração e o foco do trabalho era na construção do algoritmo, esta tela é simples e conta com os botões “Iniciar Processamento” e “Paly / Pause”.

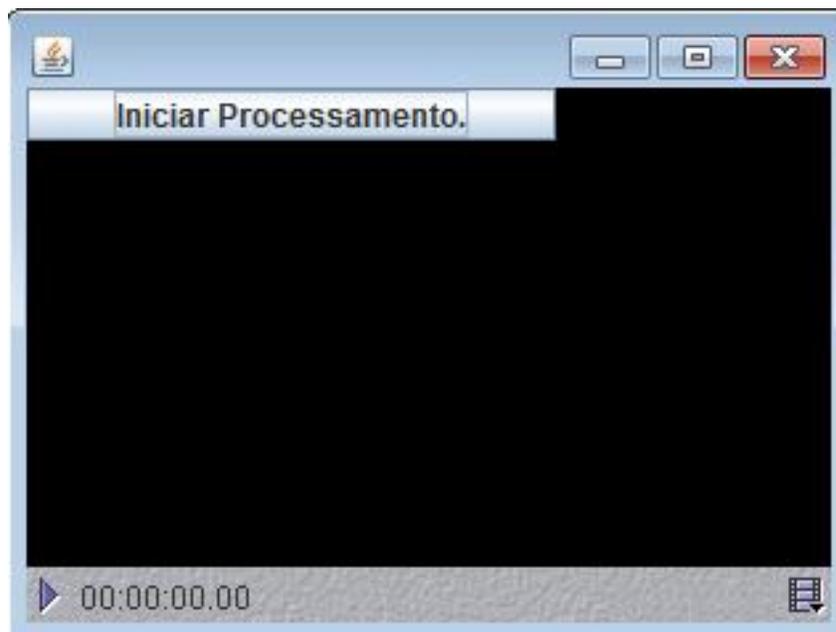


Figura 13: Tela desenvolvida para iniciar a captura de imagem, com captura não iniciada.



Figura 14: Tela da interface com captura de imagem iniciada.

Na etapa de teste, foram feitos vídeos de ambiente que continham ou não focos de incêndio. O tempo dos vídeos foram de um minuto, três minutos e cinco minutos para ambos os casos.

Os vídeos onde existiam focos de incêndio (com um total de seis minutos de existência de foco, somados os três vídeos), o algoritmo conseguiu identificar a existência do foco de incêndio em cinco minutos. Com este resultado foi possível constatar o bom desempenho do algoritmo.

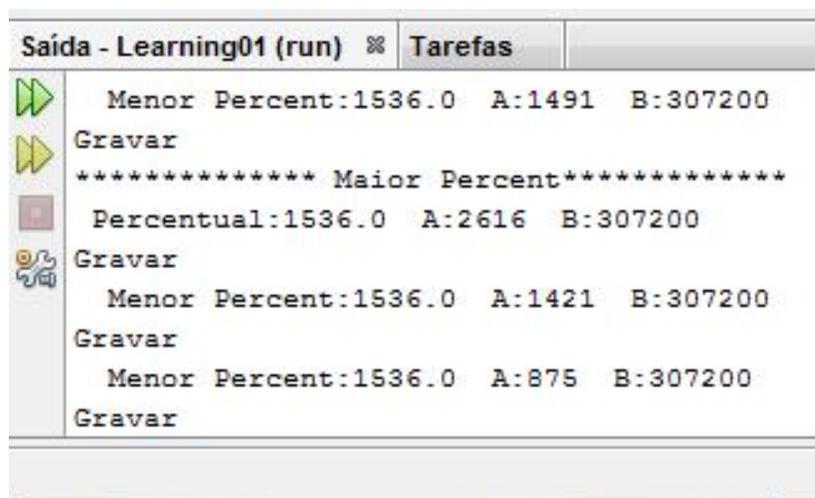
O resultado do processamento é o demonstrado na Figura 15 onde a imagem capturada dinamicamente é avaliada e o corpo do incêndio é identificado com a coloração azul, mesmo estando em um ambiente com cores próximas ao vermelho .



Figura 15: Imagem com foco identificado

As partes marcadas em azul são as que continham pixels dentro do intervalo pesquisado. Como esta foi registrada, a porcentagem de pixels identificados supera ou igual a 3% do total da imagem.

Durante o processamento foi registrado a quantidade de pixels encontrado dentro do padrão na imagem e a quantidade de pixels total da imagem. Os valores superiores a 3% do total são armazenados para futuras comparações. Na Figura 16 é possível ver o resultado do processamento feito pelo algoritmo durante o processamento em tempo real.



```

Saída - Learning01 (run) % Tarefas
Menor Percent:1536.0 A:1491 B:307200
Gravar
***** Maior Percent*****
Percentual:1536.0 A:2616 B:307200
Gravar
Menor Percent:1536.0 A:1421 B:307200
Gravar
Menor Percent:1536.0 A:875 B:307200
Gravar

```

Figura 16: Informações sobre a quantidade de pixels processados.

Das imagens resultantes do processamento, o algoritmo conseguiu identificar 73% do total do corpo do foco de incêndio. Os 27% de erro foram gerados devido a a baixa qualidade da câmera, que faz com que as cores mais claras, como o amarelo por exemplo, sejam transformadas em branco. A cor branca é descartada como um identificador de foco de incêndio devido a sua vasta presença em ambientes.

O custo computacional foi baixo em relação ao processamento. A figura 16 mostra o estado do computador quando o algoritmo não estava sendo executado e a Figura 17 mostra o estado do computador quando o algoritmo estava executando.

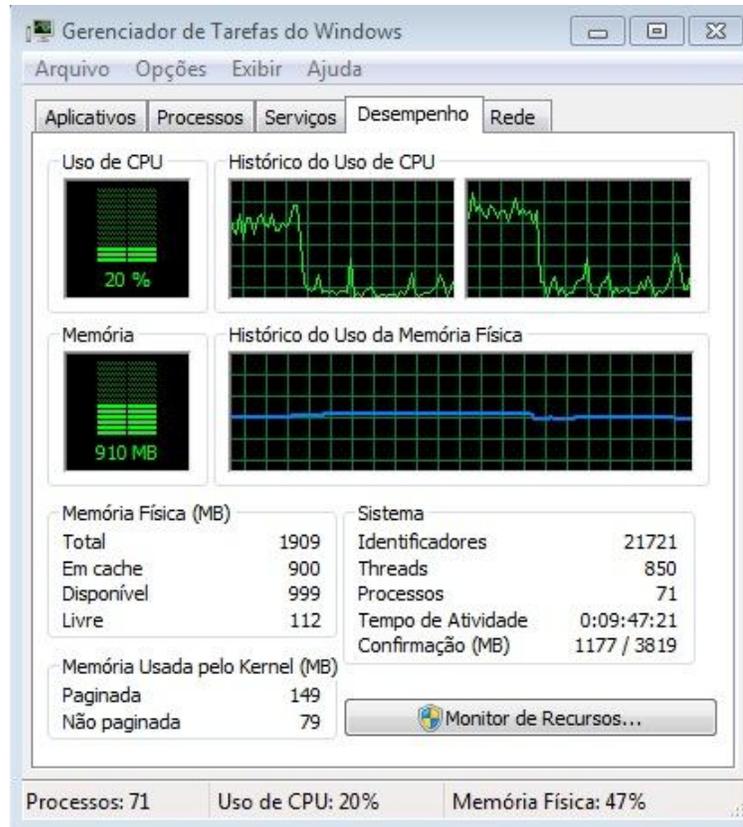


Figura 17: Estado do computador com algoritmo não executado.

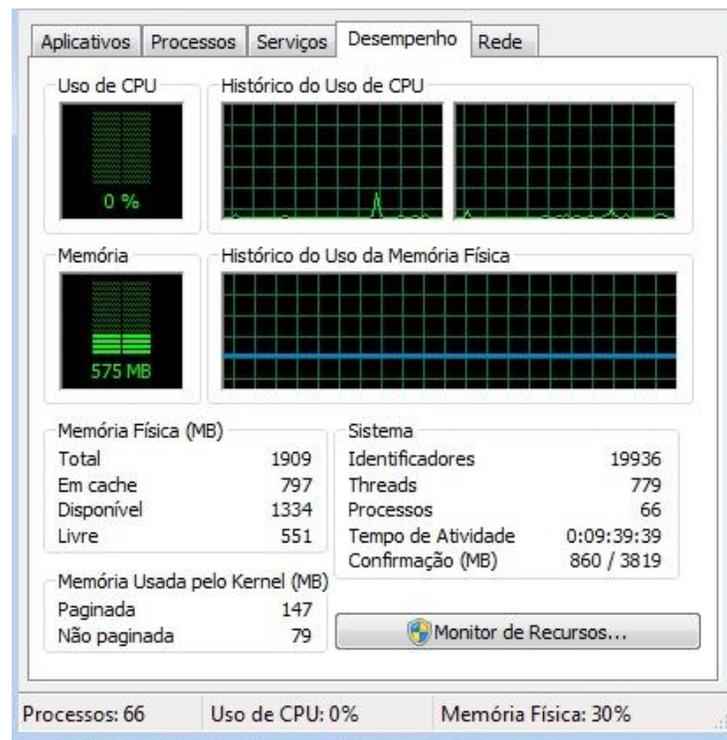


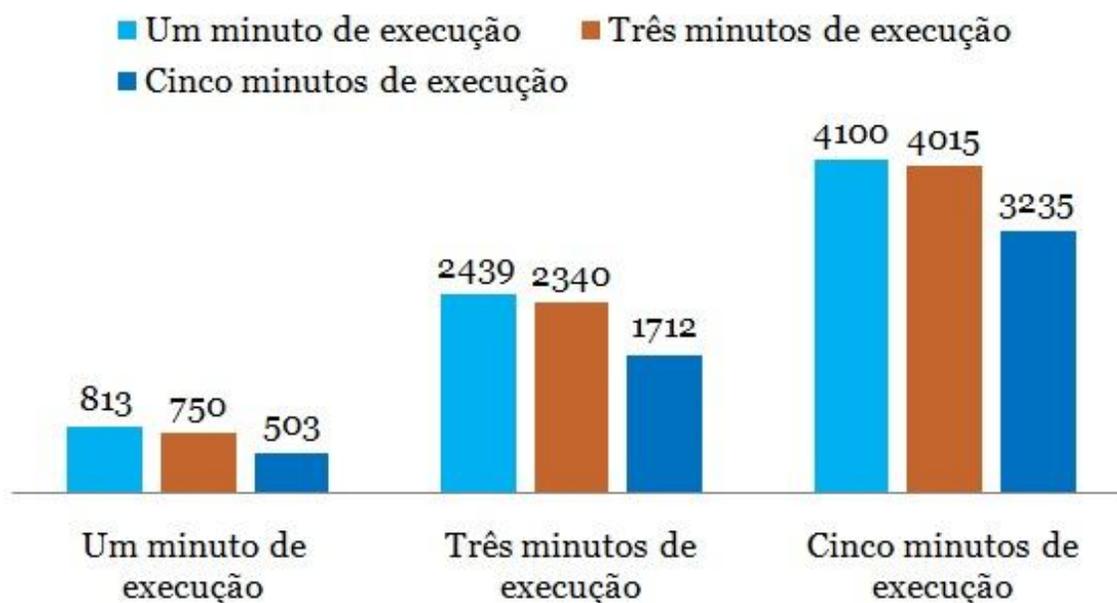
Figura 18: Estado do computador quando o algoritmo estava em execução.

Quando o algoritmo esteve processando as imagens, o custo de processamento do computador foi de 20% e quando não estava sendo executado foi de 0%. A memória alocada para que a aplicação funcionasse foi de 910 Megabytes e quando não estava foi de 575 Megabytes. Uma diferença de 335 MB de memória alocada.

O algoritmo aplicado ao processamento dinâmico atendeu a expectativa não super alocando memória, o que geraria um custo computacional altíssimo, tornando inviável a sua aplicação. Desta maneira, a estrutura criada, que é composta da avaliação da imagem que será processada, armazenamento de informações sobre a imagem e comparação de valores conseguiu manter um baixo tempo de processamento com qualidade de informações.

O Gráfico 1 apresenta a relação das imagens capturadas versus a quantidade de imagens identificadas.

Gráfico 1: Relação de quantidade de imagem capturada, imagem com foco de incêndio e imagem identificada.



No gráfico, as colunas em azul claro representam a quantidade de imagens geradas por minuto pela JMF, que são de 813 frames por minuto, as colunas vermelhas representam a quantidade de imagens que continham foco de incêndio e em azul, a quantidade de frames que o algoritmo conseguiu identificar. Para todos os testes realizados, em média, o algoritmo identificou cerca de 77% das imagens que continham foco de incêndio.

A maioria dos registros ocorreu quando os focos de incêndio estavam a dois metros de distância em relação a *webcam*. Este resultado se deve as condições da proporção do fogo em relação à distância e a qualidade da captura da câmera influenciou na avaliação e isto é crucial para o algoritmo, pois, afeta a comparação de valores para separar ruído do verdadeiro foco.

5 CONSIDERAÇÕES FINAIS

Somando os três testes realizados, que compreendiam o tempo de um minuto, três e cinco de vídeo. Durante a existência de fogo neste total, foi possível identificar 77% deste total com o algoritmo, além de conseguir identificar 73% do total do corpo o foco de incêndio que foi analisado, assim, mostrando sua eficiência.

Seu relativo baixo custo computacional faz com seja possível aplicar esta técnica de monitorando de ambientes por longas horas, isto é, sem a necessidade de parar o processamento e reiniciá-lo.

Esta forma de processamento também proporciona vastas possibilidades de aprimoramento, visando aumentar a segurança no apontamento das informações.

Além do resultado satisfatório, foi possível observar a relação da qualidade da câmera que faz a captura de imagens e a influência do espaço monitorado, nos resultados obtidos. Isto só pode ser percebido nos testes finais, onde o algoritmo processou imagens capturadas por outra *webcam* e por distancias maiores do foco de incêndio em relação a *webcam*, constatando disparidades em resultados anteriores.

Este estudo demonstra que é possível usar processamento de imagens digitais e sinais como método de avaliação de imagens dinâmicas, porém devem-se avaliar as condições do ambiente como a distância do objeto ou o perímetro que se pretende monitorar além da qualidade da câmera que irá capturar as imagens.

5.1 Trabalhos Futuros

Para trabalhos futuros, estudar a melhor relação sobre distância e qualidade de câmeras para captura de imagens. Como não foi o foco deste trabalho estudar qualidades do ambiente e dos equipamentos para melhor trabalhar com as

REFERÊNCIAS

FIGUEIRÊDO, R. C. **Um Estudo de Algoritmos de Processamento de imagem e Reconhecimento de Padrão em Imagem Digital Capturada em Ambiente Aberto**. (Dissertação de Mestrado) Universidade Federal da Bahia – UFBA, Escola Politécnica, Programa de Pós-Graduação em Mecatrônica, 2007.

GIL, A. C. Métodos e Técnicas de Pesquisa Social. 5. Ed. São Paulo: Atlas, 1999.

LIU, C.; AHUJA, N. **Vision Based Fire Detection**. University of Illinois Urbana-Champaign Urbana 2004 <http://vision.ai.uiuc.edu/newpubs/icpr04_fire.pdf> acesso em 15 de maio de 2012

Oracle, mantenedora da linguagem de programação JAVA <<http://www.oracle.com/technetwork/java/javase/tech/index-jsp-140239.html>>. Acesso em: 08 maio de 2012.

Oracle, mantenedora da biblioteca JMF. Disponível em : <<http://www.oracle.com/technetwork/java/javase/tech/index-jsp-140239.html>> acessado em 10 de Setembro de 2012.

SBEGHEN, R. C. **Processamento Digital de Imagem**. Jaguariúna: 2007. Disponível em <<http://bibdig.poliseducacional.com.br/document/?code=51>> Acessado em: 05 maio de 2012.

SCAVASSA, G. A, RIBEIRO, P. B. Utilizando processamento de imagem digital para identificação de foco de incêndio em imagens dinâmicas, **SIICUSP – Sinpósio Internacional de Iniciação Científica da USP**, 2012, São Paulo, 2012 a. Disponível em:<
<https://uspdigital.usp.br/siicusp/cdOnlineTrabalhoObter?numeroInscricaoTrabalho=2006&numeroEdicao=20&print=S>> acessado em 01 de novembro de 2012.

SCAVASSA, G. A, RIBEIRO, P. B. Utilizando Processamento de Imagem Digital na Identificação de Foco de Incêndio, **XIX Fórum de Iniciação Científica da Universidade Sagrado Coração**. Bauru, São Paulo, 2012 b.

SCAVASSA, G. A., VERA, J. H., RIBEIRO, P. B. Processamento de Imagem Digital para a Identificação de Foco de Incêndio In: VI Jornada Científica das Faculdades Integradas de Bauru, 2011, Bauru. **VI Jornada Científica - FIB - ISSN 1980-5543.** , 2011 a. Disponível em:

<<http://fib.brpwebprojects.com/jornada/download/trabalhos/anais/0022011ads.pdf>> acessado em 04 de junho de 2012.

SCAVASSA, G. A. VERA, J. H., RIBEIRO, P. B. Detecção de incêndio usando PIS In: IX Salão de Iniciação Científica: Pós Graduação, Pesquisa e Extensão da Universidade Luterana do Brasil, 2011 b, Torres, Rio Grande do Sul **IX Salão de Iniciação Científica - ULBRA**, 2011.

SÖNMEZ, A. C. **Flame recognition in video.** Yildiz Teknik Universitesi Elektrik-Elektronik Fakültesi Bilgisayar Mühendisliği Bölümü, Istanbul, 2007. Disponível em: <http://www.emo.org.tr/ekler/842858ddde53cb0_ek.pdf> acessado em: 01 de maio de 2012.

TORINO, C. V. BARRET, D. T. Detecção de Fumaça Através do Processamento de Imagem Utilizando Redes Neurais em um Sistema Híbrido. XIII CBA 2002 - **Congresso Brasileiro de Automtica.** pp 1-6, 2002

TORINO, C. V.; ALMEIDA, L. A. O. L. Detecção de Fumaça Através do Processamento de Imagem Uitlizando Descritores Espaço-Temporais. XII CBA 2000 - **Congresso Brasileiro de Automtica.** pp 2090-2095, 2000.

PHILLIPS III, W., SHAH, M., Lobo, N. V. Flame Recognition in Video, **Pattern Recognition Letters**, Volume 23 (1-3), Sayfa 319-327, Jan. 2002