



**UNIVERSIDADE FEDERAL RURAL DO SEMI-ÁRIDO
UNIVERSIDADE DO ESTADO DO RIO GRANDE DO NORTE
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA
COMPUTAÇÃO**



BRUNO RAMON DE ALMEIDA E SILVA

**SISTEMA DE CONTAGEM AUTOMÁTICA DE OBJETOS
UTILIZANDO PROCESSAMENTO DIGITAL DE
IMAGENS EM DISPOSITIVOS MÓVEIS**

MOSSORÓ – RN

2014

BRUNO RAMON DE ALMEIDA E SILVA

**SISTEMA DE CONTAGEM AUTOMÁTICA DE OBJETOS
UTILIZANDO PROCESSAMENTO DIGITAL DE
IMAGENS EM DISPOSITIVOS MÓVEIS**

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação – associação ampla entre a Universidade do Estado do Rio Grande do Norte e a Universidade Federal Rural do Semi-Árido, para a obtenção do título de Mestre em Ciência da Computação.

Orientador: Prof. Dr. Dannel Cavalcante Lopes – UFERSA.

MOSSORÓ – RN

2014

**Catálogo da Publicação na Fonte.
Universidade do Estado do Rio Grande do Norte.**

Silva, Bruno Ramon de Almeida e.

Sistema de contagem automática de objetos utilizando processamento digital de imagens em dispositivos móveis. / Bruno Ramon de Almeida e Silva. – Mossoró, RN, 2014

55 f.

Orientador(a): Prof. Dr. Dannel Cavalcante Lopes.

Dissertação (Mestrado em Ciência da Computação). Universidade do Estado do Rio Grande do Norte. Universidade Federal Rural do Semi-Árido. Programa de Pós-Graduação em Ciência da Computação.

1. Contagem - Dissertação. 2. PDI - Dissertação. 3. Visão computacional - Dissertação. I. Lopes, Dannel Cavalcante. II. Universidade do Estado do Rio Grande do Norte. III. Título.

UERN/BC

CDD 006.6

Bibliotecária: Elaine Paiva de Assunção CRB 15 / 492

AGRADECIMENTOS

A Deus, pela sua infinita misericórdia e por ter me iluminado meu caminho ao longo dessa trajetória.

A meus pais, pelo apoio e amor incondicional, sem os quais, nada disso seria possível.

Aos professores orientadores, pelo auxílio e orientação na concepção e desenvolvimento do trabalho.

Aos amigos e colegas, pelo incentivo e pelo apoio constantes.

RESUMO

A contagem de objetos é um problema comum em diversas áreas da indústria, comércio e serviços em geral. Na indústria da cerâmica vermelha, por exemplo, a contagem de telhas é um dos fatores que impactam no escoamento da produção. Uma forma de realizar a contagem de objetos é através de técnicas de Visão Computacional. No presente trabalho, foi desenvolvido um sistema automático para contagem de telhas utilizando processamento digital de imagens em dispositivos móveis. O sistema recebe como entrada imagens gravadas pelo dispositivo e calcula a quantidade total de telhas presente no vídeo.

Palavras-Chave: Contagem, PDI, Visão computacional, Telhas.

ABSTRACT

Counting objects is a common problem in many areas of industry, trade and services in general. In the ceramic industry, for example, counting tiles is one of the factors which impact the flow of production. One way to accomplish the object count is through computer vision techniques. In this paper, an automatic system was developed for counting tiles using digital image processing on mobile devices. The system receives as input images recorded by the device and calculates the total amount present in the video tiles.

Keywords: Counting, DIP, Computer vision, Tiles.

Lista de Figuras

1	Exemplo de escadas de peixes	4
2	Técnica para estimação de movimento	7
3	Etapas fundamentais em processamento de imagens digitais	9
4	Exemplo de limiarização	11
5	Exemplo de dilatação	12
6	Exemplo de erosão	13
7	Uma máscara genérica 3 x 3	14
8	Máscara usada na detecção de pontos	14
9	Exemplo de detecção de pontos	15
10	Máscara usada na detecção de linhas	15
11	Exemplos de detecção de linhas	16
12	Máscaras utilizadas no filtro de Sobel	17
13	Resultado da segmentação por divisão e agrupamento	19
14	Estrutura da RNA utilizada na identificação dos objetos	28
15	Exemplo de objetos utilizados na simulação	30
16	Fluxo de processos do sistema contador de telhas	35
17	Caminhão carregado com telhas	36
18	Exemplo de imagem capturada pela câmera	37
19	Conversão em escala de cinza	37
20	Exemplo de limiarização	38
21	Exemplo de dilatação	38
22	Segmentação das bordas das telhas	40
23	Dimensões da telha colonial	40
24	Seletores utilizados no reconhecimento	41
25	Região de contagem	42
26	Fluxograma - telhas a serem rastreadas	43
27	Elementos do algoritmo de contagem	43
28	Movimentação das telhas	45
29	Exemplo de movimentação das telhas	46

30	Telhas fora do padrão	48
31	Gráfico comparativo resultado da contagem	50

Conteúdo

1	Introdução	1
1.1	Motivação	1
1.2	Objetivos: geral e específicos	2
1.3	Trabalhos relacionados	2
1.3.1	Rastreamento e contagem de peixes utilizando filtro preditivo	3
1.3.2	Um estudo prático para contagem volumétrica automática de veículos usando Visão Computacional	4
1.3.3	Sistema automático de reconhecimento e contagem de ovos da dengue	6
1.3.4	Sistema computacional para contagem automática de pessoas baseado em análise de sequências de imagens	6
1.4	Organização do texto	7
2	Processamento digital de imagens	8
2.1	Introdução	8
2.2	Pré-processamento	10
2.2.1	Representação em escala de cinza	10
2.2.2	Limiarização	11
2.2.3	Operadores morfológicos	11
2.3	Segmentação de imagens	13
2.4	Segmentação baseada em pontos, linhas e bordas	13
2.4.1	Detecção de pontos	14
2.4.2	Detecção de linhas	15
2.4.3	Detecção de bordas	16
2.5	Segmentação orientada a regiões	17
2.5.1	Crescimento de regiões por agregação de <i>pixels</i>	17
2.5.2	Difusão e fusão de regiões	18
3	Dispositivos móveis	20
3.1	Introdução	20
3.2	Trabalhos relacionados	22

3.3	Conclusão	24
4	Deteccção e rastreamento de objetos	25
4.1	Introdução	25
4.2	Técnicas para deteccção de objetos	25
4.2.1	Redes Neurais Artificiais	26
4.2.2	Métodos exatos	26
4.2.3	Trabalhos relacionados	27
4.3	Técnicas para rastreamento de objetos	29
4.3.1	Filtro de Kalman	30
4.3.2	CONDENSATION (<i>Conditional Density Propagation</i>)	31
4.3.3	Trabalhos relacionados	32
4.4	Conclusão	33
5	Sistema proposto	34
5.1	Visão geral do sistema	35
5.2	Aquisição das imagens	36
5.3	Pré-processamento	37
5.4	Segmentação	39
5.5	Reconhecimento	39
5.6	Rastreamento	41
5.7	Contagem	46
6	Testes e Resultados	47
6.1	Aplicativo desenvolvido	47
6.2	Resultados	48
7	Conclusões e trabalhos futuros	51
7.1	Trabalhos futuros	51

1 Introdução

A construção civil é um dos setores relevantes da economia brasileira. Só em 2012, o setor fechou o ano com um crescimento de 4% do Produto Interno Bruto (PIB), segundo dados do Sindicato da Indústria da Construção Civil do Estado de São Paulo (SindusCon-SP). Nesse segmento, a telha é uma importante matéria prima utilizada no revestimento de casas e outras edificações. Ela é produzida em olarias ou cerâmicas, como são popularmente conhecidas. De acordo com a Associação Nacional da Indústria Cerâmica (ANICER), no Brasil existem aproximadamente 6.903 empresas atuando nesse setor, o que representa 4,8% da indústria da construção civil.

A modernização da indústria ceramista aconteceu de forma tardia. Até poucos anos atrás, o processo de produção de telhas era manual e oneroso tanto em relação ao tempo quanto ao custo de produção. No entanto, esse cenário vem se modificando nos últimos anos. Alguns setores desse segmento vêm passando pelo processo de modernização, como o setor de corte por exemplo, que já conta com uma máquina denominada de cortadeira transversal de telhas, responsável por toda a automação do corte e alinhamento [da Silva, 1994]. O setor de cozimento também passa por esse processo, hoje em dia existem medidores de alta temperatura (pirômetros) que auxiliam no controle do aquecimento da matéria prima, a argila, possibilitando que todo o material seja cozido de maneira uniforme. Isso gera como resultado telhas com um padrão de cores uniforme, o que aumenta a qualidade e consequentemente o valor comercial do produto.

1.1 Motivação

Apesar dos avanços conseguidos, alguns processos ainda são realizados de maneira totalmente manual, como a contagem de telhas. Após modelada, a argila passa pelo setor de corte e em seguida pelo cozimento. A partir daí, as telhas estão prontas para transporte. Elas são empilhadas nos veículos e contadas uma a uma de forma manual, sem o auxílio de qualquer equipamento. Esse processo leva demasiado tempo, haja vista que a produção do material ocorre naturalmente em larga escala e o transporte é feito em grandes quantidades. Logo, a etapa de contagem acaba se tornando um fator limitante do escoamento da produção.

Outro problema decorrente da contagem manual de telhas é a margem de erro, que ge-

ralmente é alta, já que os trabalhadores precisam contar uma enorme quantidade de telhas no menor intervalo de tempo possível, o que torna o erro praticamente inevitável. O desenvolvimento de uma forma eficaz de contagem aumentaria a capacidade de distribuição das cerâmicas, o que traria um grande benefício para a construção civil como um todo.

1.2 Objetivos: geral e específicos

O objetivo geral deste trabalho consiste no desenvolvimento de um sistema capaz de detectar e contar objetos em imagens obtidas a partir de dispositivos móveis. Como estudo de caso foi utilizado a construção civil, especificamente o problema da contagem de telhas. O sistema desenvolvido é capaz de contar a quantidade total de telhas presentes em um vídeo.

Para alcançar o objetivo geral, foi necessário cumprir os seguintes objetivos específicos:

- Implementação de um protótipo de *software* para aplicação de filtros em imagens o qual fosse possível testar e avaliar filtros e operações morfológicas na etapa de pré-processamento;
- Elaboração de um algoritmo capaz de realizar a segmentação das telhas de modo a fornecer entradas válidas para etapa de rastreamento;
- Implementação de um algoritmo de rastreamento para solucionar o problema de contar objetos mais de uma vez ao longo do vídeo;
- Elaboração de um algoritmo de contagem dos objetos rastreados.

1.3 Trabalhos relacionados

A seguir são apresentados alguns trabalhos relacionados ao tema proposto, visando investigar o estado da arte e as técnicas utilizadas por pesquisadores da área em problemas semelhantes.

A detecção e contagem de objetos é um problema bastante explorado na literatura. Existem diferentes trabalhos referentes ao tema dentre os quais pode-se destacar: detecção e contagem de pessoas, contagem de veículos, rastreamento de objetos e análise de tráfego. Apesar de semelhantes, cada categoria de problemas possui características particulares, tais como as seguintes:

- Posicionamento da câmera - é um ponto importante a se considerar no problema, já que influencia diretamente na etapa de reconhecimento dos objetos. Geralmente a câmera é mantida estática e posicionada de maneira a obter uma visão frontal, lateral ou em perspectiva da cena onde os objetos estão inseridos.
- Quantidade de objetos - o número de objetos a serem rastreados também influencia no processo, sobretudo na etapa de rastreamento. Existem classes de algoritmos capazes de rastrear apenas um objeto por vez, como também múltiplos objetos em uma mesma cena. A característica do problema definirá qual classe deve ser utilizada.
- Direção do movimento - alguns problemas se restringem a ambientes controlados onde a direção do movimento dos objetos é conhecida previamente. Em linhas de produção, por exemplo, as esteiras em sua maioria se movem em um único sentido. Em outros casos no entanto, o fluxo do movimento não é controlado e os objetos podem se mover de maneira arbitrária na cena aumentando a complexidade do problema.

Existem diferentes maneiras de contar objetos. Em algumas aplicações, é comum a utilização de sensores ou outros equipamentos de medição como forma de monitorar a quantidade de itens produzidos. Contudo, o levantamento foi feito apenas em trabalhos que envolvam contagem através de análise e processamento de imagens, ou seja, aqueles que utilizaram processos relacionados à área de pesquisa conhecida como Visão Computacional. Uma característica desse tipo de contagem é que ela não requer a instalação de uma infraestrutura física no local, requerem apenas que a câmera seja posicionada de forma a obter um bom campo de visão da área de interesse.

1.3.1 Rastreamento e contagem de peixes utilizando filtro preditivo

O trabalho apresentado em [de Moraes, 2005] aborda o problema de estimar o número de peixes que nadam em mecanismos de transposição, também conhecidos como escada de peixes. Esse mecanismo visa auxiliar os peixes a transpor barragens no período de reprodução. Nesse período, os peixes viajam longas distâncias até as cabeceiras dos rios, mas ao encontrarem as barreiras, ficam impedidos de chegarem ao local da desova. A escada de peixes é uma das formas de solução desse problema. O sistema de contagem proposto no artigo, juntamente com um módulo complementar para classificação de peixes, compõem

um sistema mais amplo cujo objetivo é realizar o monitoramento e a avaliação da eficácia de escadas de peixes. A Figura 1 mostra exemplos de um peixe subindo uma escada de transposição.



(a) Escada de peixe



(b) Salmão subindo uma escada de transposição

Figura 1: Exemplo de escadas de peixes

Fonte: [de Moraes, 2005]

A ideia principal do trabalho é aplicar técnicas de visão computacional na contagem de peixes que nadam em mecanismos de transposição construídos em barragens hidrelétricas.

O trabalho utiliza uma abordagem baseada em análise de sequência de imagens e no uso de um rastreador bayesiano de múltiplos objetos, o qual permite a realização de um rastreamento mais confiável de peixes mesmo diante de variações ambientais bruscas, como variações de luminosidade e propriedades da água.

As etapas do processo são descritas a seguir: inicialmente as imagens são adquiridas por uma câmera e é feito um pré-processamento para que possam ser eliminados possíveis efeitos espúrios das imagens causados pela água. Após essas etapas ocorre a detecção dos peixes que é feita através de uma segmentação probabilística cujo resultado serve de entrada para o algoritmo de rastreamento. O algoritmo impede que um mesmo peixe seja contado mais de uma vez durante o processo, no trabalho foi utilizado o filtro de partículas CONDENSATION. Com base nas trajetórias, são escolhidas sub-imagens contendo o peixe para classificação e contagem.

1.3.2 Um estudo prático para contagem volumétrica automática de veículos usando Visão Computacional

A contagem volumétrica de veículos trata-se da coleta de informações sobre a quantidade de veículos que trafegam em um determinado local e horário determinados. Essas informações

são utilizadas para o planejamento de tráfego urbano. Em [Feitosa, 2012] é proposto um trabalho cujo principal objetivo é realizar a contagem volumétrica de veículos que trafegam em uma rua, a partir de imagens obtidas com uma câmera fixada na lateral da via.

O método proposto possui 5 etapas: pré-processamento da imagem de entrada, reconhecimento da imagem de fundo, reconhecimento da área de movimento, segmentação de objetos e rastreamento dos objetos na sequência de quadros.

Inicialmente é feito um pré-processamento das imagens obtidas a fim de eliminar possíveis ruídos oriundos do processo de aquisição da imagem. Logo após é realizada a segmentação dos objetos através de uma técnica denominada de subtração de fundo. Segundo [Yang et al., 2011], esse método se caracteriza pela utilização de duas imagens: uma imagem de referência de fundo e outra contendo os objetos de interesse, a qual será subtraída o fundo. A imagem de saída é gerada a partir da subtração *pixel a pixel* da imagem de referência pela segunda imagem. Um limiar é definido para classificar os *pixels* da subtração como sendo pertencentes ao fundo ou ao objeto.

Após a etapa de segmentação inicia-se o rastreamento dos objetos. Normalmente em um processo de contagem por processamento de imagens, define-se uma determinada área de contagem na qual os objetos de interesse se movimentam. Essa área geralmente é delimitada previamente, contudo, o algoritmo desenvolvido no trabalho realiza a determinação automática da região de contagem, baseado na quantidade de movimento em cada região da imagem.

Após delimitada a região de interesse, a contagem dos objetos é feita através do acúmulo e comparação de informações dos objetos segmentados ao longo da sequência de imagens. Para cada objeto encontrado, são armazenadas informações como: ponto mínimo e máximo considerando o menor retângulo capaz de englobar tal objeto, ponto central e último quadro que conta com a presença do objeto. A cada iteração os *frames* são comparados com seus respectivos antecessores para que se defina o grau de similaridade dos objetos encontrados. Um algoritmo desenvolvido no trabalho determina quais objetos são correspondentes nas imagens e assim é evitada a contagem de cada objeto mais de uma vez, possibilitando a contagem de maneira satisfatória.

1.3.3 Sistema automático de reconhecimento e contagem de ovos da dengue

No trabalho apresentado em [da Silva, 2010] foi desenvolvido um sistema capaz de realizar a contagem automática de ovos do mosquito *Aedes Aegypti*, em imagens. As imagens foram obtidas através da digitalização de palhetas, chamadas de ovitrampas, que são armadilhas espalhadas por uma região com o intuito de coletar ovos depositados pelos mosquitos. Para classificar e contabilizar os ovos encontrados nas imagens foram utilizadas redes RBF.

Após realizada a aquisição das imagens, é feita a extração de características, baseadas nos descritores adotados pela abordagem. Os descritores utilizados foram: a média das componentes RGB dos *pixels* da imagem e seus respectivos desvios padrões.

Para que o sistema fosse capaz de reconhecer os ovos do mosquito nas imagens, foi necessário um treinamento da rede. No treinamento, foram construídos conjuntos de teste para dois padrões: ovos e fundo. Foram utilizadas 50 imagens aleatórias contendo as mais diferentes tonalidades de ovos e 30 imagens com ruídos.

1.3.4 Sistema computacional para contagem automática de pessoas baseado em análise de sequências de imagens

Em [da Silva, 2008] é apresentado um sistema automatizado para contagem de pessoas. No trabalho proposto, uma câmera é posicionada em direção à região que deseja se monitorar. Toda vez que uma nova pessoa entra na região de fronteira da área monitorada a pessoa é detectada e rastreada. O rastreamento serve para que uma mesma pessoa não seja contada mais de uma vez.

Para rastrear os objetos de interesse da cena, nesse caso as pessoas, é utilizado um círculo deslizante que envolve cada pessoa detectada. Esse círculo representa a região relativa aquele objeto na imagem. Uma rotina de rastreamento baseada no filtro de Kalman é utilizada toda vez que um novo objeto entra na região de interesse. Cada objeto possui coordenadas de localização determinadas pelo centróide do conjunto de *pixels* relativos a ele.

A cada iteração a rotina de rastreamento compara a ocorrência dos objetos em uma área estimada ao redor deles. A Figura 2 mostra a comparação entre duas iterações consecutivas, o instante t_i e seu antecessor.

A Figura 2a mostra o centróide do conjunto de pontos no instante t_i , localizado na co-

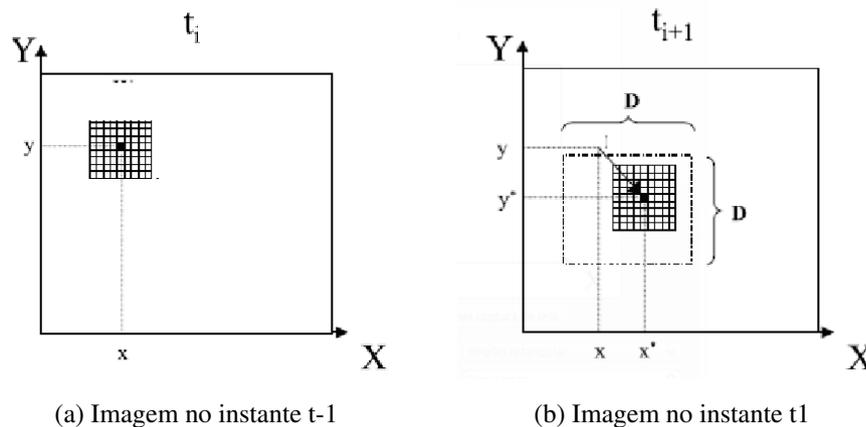


Figura 2: Técnica para estimação de movimento

ordenada (x,y) . Na iteração seguinte (t_i+1), ilustrada na Figura 2b, há um deslocamento do centróide para a posição (x',y') . O algoritmo verifica a ocorrência do ponto no espaço delimitado por D .

A abordagem utilizada poderá apresentar problemas quando duas pessoas ocuparem espaços bastante próximos, por exemplo, quando estiverem abraçadas. A abordagem proposta trata as colisões através da análise de dois parâmetros: distância entre os centróides e dimensão das pessoas. Caso a distância entre os centróides seja menor que a soma dos raios, há uma interseção. A solução encontrada foi utilizar a álgebra vetorial para transladar as duas pessoas para a posição a qual elas apenas se tocam, sem interseção.

1.4 Organização do texto

Os próximos capítulos dessa dissertação estão organizados conforme apresentado a seguir: o capítulo 2 apresenta um referencial teórico sobre as técnicas de análise e processamento de imagens; no capítulo 3 é feita uma abordagem sobre dispositivos móveis e sua utilização no desenvolvimento do trabalho; o capítulo 4 apresenta uma discussão sobre o problema da detecção de telhas e são mostrados alguns algoritmos de rastreamento; o capítulo 5 mostra detalhes sobre a metodologia utilizada no desenvolvimento do trabalho; o capítulo 6 apresenta os testes realizados e resultados obtidos; e o capítulo 7 a conclusão e considerações finais.

2 **Processamento digital de imagens**

2.1 **Introdução**

O Processamento Digital de Imagens (PDI) é uma importante área da computação e possui aplicações em diversos setores da indústria, comércio e outras áreas, como a saúde por exemplo. Desde a aplicação de um simples filtro até operações complexas como a restauração de imagens, o PDI possui ferramentas robustas que permitem o desenvolvimento de aplicações que ajudam a solucionar problemas do mundo real.

Nesse capítulo é apresentada a fundamentação teórica das técnicas de processamento de imagens utilizadas nesse trabalho. Primeiramente é feita uma introdução sobre o PDI e abordados filtros e operações elementares necessárias para o pré-processamento de imagens. Em seguida são apresentadas técnicas de segmentação, ou seja, a subdivisão de imagens em objetos constituintes, outra etapa fundamental do processo.

Segundo [Filho & Neto, 1999], a área de processamento de imagens viabiliza um grande número de aplicações em duas categorias distintas: (1) o aprimoramento de informações visuais para interpretação humana; e (2) a análise automática por computador, de informações extraídas de uma cena. O presente trabalho está inserido na segunda categoria mencionada, já que um dos objetivos definidos é identificar objetos em uma cena e extrair informações relativas a eles, para então processá-las.

O processamento de imagens é realizado em etapas bem definidas que compartilham informações entre si para produzir o resultado esperado, de forma que o resultado de cada etapa interfere diretamente nas etapas subsequentes. Segundo [Gonzalez & Woods, 1992], as etapas fundamentais para o processamento digital de imagens se dividem em: aquisição da imagem, pré-processamento, segmentação, descrição, reconhecimento e interpretação. A Figura 3 ilustra o fluxo dessas etapas.

Na aquisição das imagens ocorre a captura e digitalização de informações visuais, ou seja, através de um dispositivo (geralmente uma câmera) capturar uma determinada cena e armazenar as informações em um formato digital. Esse processo pode gerar ruídos, seja por falha mecânica do dispositivo de captura ou mesmo por características intrínsecas da cena. Por isso, antes de realizar de fato o processamento das informações, pode ser necessário ajustes para melhorar a imagem. Essa etapa é denominada de pré-processamento.

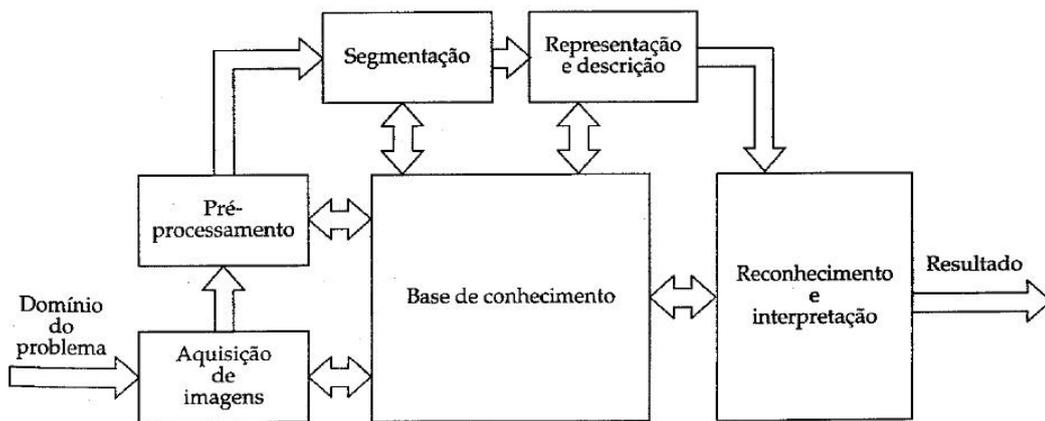


Figura 3: Etapas fundamentais em processamento de imagens digitais

Fonte: [Gonzalez & Woods, 1992]

O pré-processamento envolve operações tais como realce de contrastes, remoção de ruído e suavização de regiões. Essas operações são ditas de baixo nível, pois trabalham diretamente com os valores de intensidade de *pixels*, sem nenhum conhecimento de informações relativas ao domínio do problema. A imagem resultante é uma imagem digitalizada de melhor qualidade que a original [Filho & Neto, 1999].

Na etapa de segmentação ocorre a divisão da imagem de entrada em partes ou objetos constituintes. Em geral, a segmentação automática é uma das tarefas mais difíceis no processamento de imagens digitais [Gonzalez & Woods, 1992]. Um processo de segmentação robusto favorece substancialmente a solução bem sucedida de um problema de imageamento.

A segmentação gera uma saída constituída por *pixels* contendo tanto informações de uma região, quanto de sua fronteira. Uma decisão importante a ser tomada é em qual dessas duas formas representar o resultado. A representação por fronteira é adequada quando o interesse na características se encontra na forma externa, como cantos ou bordas. Já a por região, é interessante quando se concentra em regiões internas, como forma e textura [Gonzalez & Woods, 1992].

O processo de descrição ou seleção de características procura extrair características que resultem em alguma informação de interesse para o domínio do problema para que possa ser utilizada nas etapas posteriores.

O reconhecimento procura atribuir um rótulo a um objeto, baseado na informação fornecida pelo descritor e a interpretação atribui um significado a um conjunto de objetos reco-

nhecidos.

O conhecimento sobre o domínio do problema está codificado em um sistema de processamento de imagens na forma de uma base de conhecimento [Gonzalez & Woods, 1992]. Este pode ser simples ou complexo dependendo do domínio do problema. Em ambos os casos, referem-se ao conhecimento prévio sobre o problema, características específicas do problema que possam ser exploradas de forma a auxiliar as etapas de processamento.

2.2 Pré-processamento

O pré-processamento é uma etapa importante no tratamento de imagens. Ela permite remover possíveis ruídos e preparar a imagem de forma que possa ser utilizada nas etapas posteriores. A seguir são apresentadas operações utilizadas no pré-processamento.

2.2.1 Representação em escala de cinza

A luz sem cor é conhecida por monocromática ou acromática. A mesma possui um atributo chamado intensidade (também conhecida como quantidade de luz). A intensidade da luz monocromática pode variar entre o preto (valor mínimo), passando por tons de cinza, até o branco (valor máximo), chamado de nível de cinza. Quanto maior o valor do nível de cinza de um *pixel*, significa uma maior quantidade de luz. A variação que encontramos na intensidade é conhecida por escala de cinza [Feitosa, 2012].

Segundo [Gonzalez & Woods, 1992], para diminuir o peso do processamento digital de imagens, antes de qualquer operação, as imagens devem ser convertidas do formato RGB para Escala de Cinza. Computacionalmente essas imagens são armazenadas utilizando-se 8 *bits* (um *byte*) por *pixel*, o que possibilita 256 intensidades possíveis. A conversão de uma imagem colorida RGB em escala de cinza é uma operação simples, aplicada a todos os pontos da imagem. O procedimento é feito da seguinte forma: para cada *pixel* da imagem, deve-se extrair os componentes R, G e B, e adicionar 30% do vermelho mais 59% do verde mais 11% do azul. Também existe uma variante que consiste em simplesmente somar os valores R, G e B e dividir por 3. Ambos resultam em uma imagem em tons de cinza.

2.2.2 Limiarização

Em muitos casos é comum o desejo de dividir a imagem em regiões de interesse com base em informações de sua cor. Nesses casos, a limiarização é a técnica indicada. O princípio da limiarização consiste em separar as regiões de uma imagem quando esta apresenta duas classes (o fundo e o objeto) [Filho & Neto, 1999]. Esse processo também é chamado de binarização, já que é caracterizado pela bipartição do conjunto de cores, segundo um limiar determinado. Ou seja, cada *pixel* da imagem é convertido em branco, caso seu valor de cinza seja igual ou superior ao limiar, ou preto, caso contrário. As Figuras 4a e 4b mostram um exemplo de limiarização de uma imagem. Na Figura 4a é mostrada a imagem original em 256 tons de cinza. Já na Figura 4b é mostrada a mesma imagem após aplicada uma limiarização com o limiar 128, valor escolhido de maneira arbitrária.

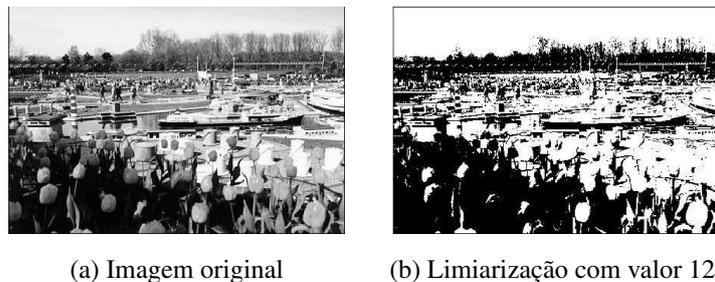


Figura 4: Exemplo de limiarização

Até o momento, assumiu-se que o valor escolhido para o limiar é arbitrário e subjetivo. Contudo, sabendo que a limiarização é feita com base nos valores de tons de cinza da imagem, e que estes podem ser representados através de um histograma, é possível também utilizar técnicas para definição de um valor limiar ótimo com base nas propriedades estatísticas da imagem.

2.2.3 Operadores morfológicos

A morfologia matemática, elaborada inicialmente por Georges Matheron e Jean Serra [Banon & Barrera, 1998], concentra esforços no estudo da estrutura geométrica das entidades presentes na imagem. O princípio da morfologia matemática consiste em extrair as informações relativas à geometria e à topologia de uma imagem, pela transformação através de um elemento estruturante [Filho & Neto, 1999]. Algumas das aplicações possíveis da

morfologia matemática são: realce, filtragem, segmentação, detecção de bordas, esqueletização e afinamento. Duas operações elementares da morfologia utilizadas nesse trabalho são a dilatação e a erosão.

O processo de dilatação consiste em obter a reflexão de B sobre a sua origem e depois deslocar esta reflexão de x . A dilatação de A por B é, então, o conjunto de todos os x deslocamentos para os quais a interseção de B e A inclui pelo menos um elemento diferente de zero. A Figura 5 mostra um exemplo dessa operação.

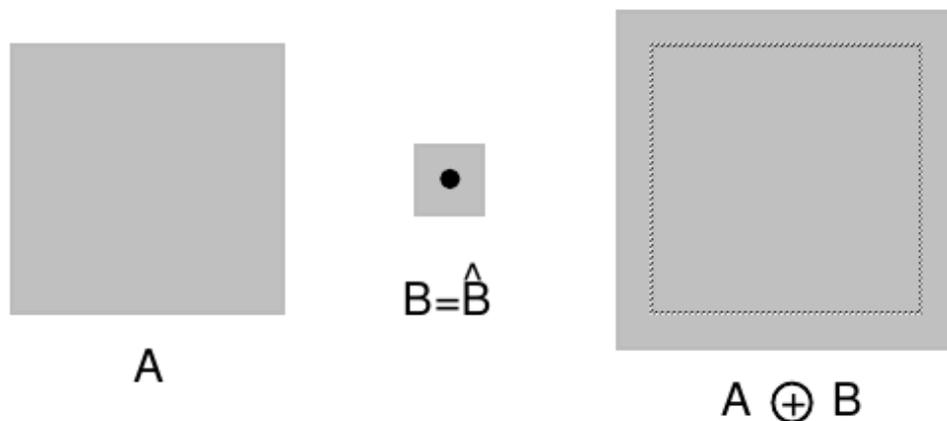


Figura 5: Exemplo de dilatação

Fonte: [Filho & Neto, 1999]

O início da figura é a imagem original, denotada por A. Ao centro o elemento estruturante utilizado na operação e à direita o resultado da operação. Nessa operação, os seguintes passos são realizados: o elemento estruturante irá percorrer toda a imagem, e cada vez que seu *pixel* central coincidir com o mesmo *pixel* na imagem original, o elemento estruturante é inserido na imagem. Com isso, como o próprio nome sugere, ocorrerá uma dilatação da imagem, já que serão inseridos a mais os *pixels* do elemento estruturante.

Já o processo de erosão é definido da seguinte forma: a erosão de A por B resulta no conjunto de pontos x tais que B, transladado de x , está contido em A. A Figura 6 mostra um exemplo desse processo. Ela é organizada da mesma forma que Figura 5, ou seja, à esquerda, centro e direita são apresentados a imagem original, elemento estruturante e imagem resultante, respectivamente.

A diferença básica da erosão para a dilatação é que quando o *pixel* central do elemento estruturante coincide com o da imagem, ao invés de acrescentar os demais *pixels* desse ele-

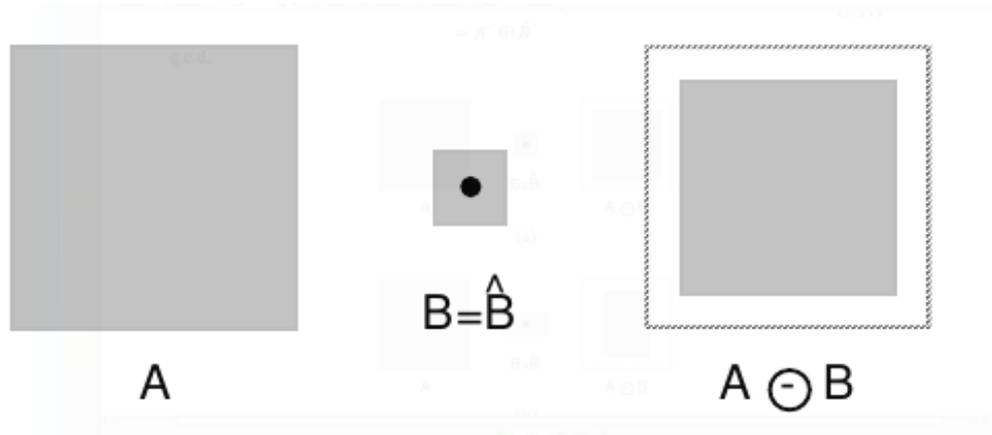


Figura 6: Exemplo de erosão

Fonte: [Filho & Neto, 1999]

mento, estes são retirados da imagem. Logo, a imagem resultante será menor que a imagem original, já que serão removidos os *pixels* do elemento estruturante.

2.3 Segmentação de imagens

A segmentação é um processo fundamental no processamento digital de imagens. Segundo [Gonzalez & Woods, 1992], segmentar significa subdividir uma imagem em suas partes ou objetos constituintes. A segmentação de objetos feita de forma automática tem sido baseada primariamente em técnicas de reconhecimento de similaridade ou de diferenças mensuráveis na imagem.

As técnicas empregadas pelos algoritmos de segmentação são geralmente baseadas em duas propriedades básicas: descontinuidade e similaridade. Na descontinuidade, a abordagem consiste em particionar uma imagem tomando por base as mudanças abruptas da intensidade do ponto (*pixel*). As principais áreas de interesse desta categoria são: detecção de pontos isolados, de linhas e de bordas da imagem. Na similaridade, a abordagem é baseada em limiarização e crescimento de regiões [da Silva, 2008].

2.4 Segmentação baseada em pontos, linhas e bordas

A segmentação baseada em pontos, linhas e bordas são realizadas através da varredura da imagem por uma máscara definida inicialmente. O uso de máscaras espaciais para processamento de imagens é comumente chamado de filtragem espacial [Gonzalez & Woods, 1992].

A Figura 7 mostra uma máscara genérica de tamanho 3 por 3 composta pelos pesos $(w_1 + w_2 + \dots + w_9)$.

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

Figura 7: Uma máscara genérica 3 x 3

Fonte: [Gonzalez & Woods, 1992]

A aplicação da máscara é feita através de sua convolução ao longo da área que se deseja calcular. Para cada *pixel* da imagem, quando o centro da máscara estiver numa posição (x,y) , o nível de cinza nesse ponto é substituído por R . O cálculo de R é feito a partir da equação (1).

$$R = w_1z_1 + w_2z_2 + \dots + w_9z_9 \quad (1)$$

Como a equação mostra, o cálculo é realizado a partir da soma das multiplicações dos pesos da máscara com as respectivas posições $z_1 + z_2 + \dots + z_9$ na imagem.

2.4.1 Detecção de pontos

A máscara para detecção de pontos é mostrada na Figura 8. Considera-se que um ponto foi detectado na posição da máscara quando o resultado da máscara naquele ponto for maior ou igual a um limiar definido.

-1	-1	-1
-1	8	-1
-1	-1	-1

Figura 8: Máscara usada na detecção de pontos

Fonte: [Gonzalez & Woods, 1992]

A máscara mede as diferenças ponderadas entre o ponto central e seus vizinhos. A ideia é que o nível de cinza de um ponto isolado seja completamente diferente do nível de cinza de seus vizinhos. A Figura 9 mostra um exemplo de aplicação dessa máscara.

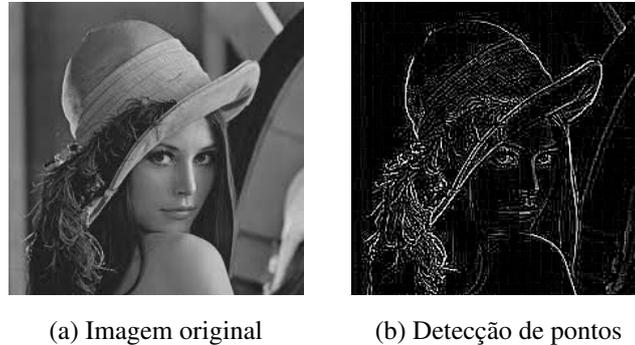


Figura 9: Exemplo de detecção de pontos

2.4.2 Detecção de linhas

A detecção de linhas é uma operação mais complexa que a detecção de um simples ponto e envolve a orientação da direção da linha. A Figura 10 mostra quatro máscaras utilizadas na detecção de linhas.

-1	-1	-1	-1	-1	-1	2	-1	-1	-1	2	-1	-1	-1	2	-1	-1	-1
2	2	2	-1	2	-1	-1	2	-1	-1	-1	2	-1	-1	-1	2	-1	-1
-1	-1	-1	2	-1	-1	-1	-1	2	-1	-1	-1	2	-1	-1	-1	2	-1
Horizontal			+45°			Vertical			-45°								

Figura 10: Máscara usada na detecção de linhas

Fonte: [Gonzalez & Woods, 1992]

As máscaras da Figura 10 se diferenciam pela direção preferencial que cada uma possui. A máscara horizontal por exemplo, é ponderada com um coeficiente maior (nesse caso, 2) na direção horizontal, logo, responderá fortemente às linhas que possuem essa orientação. Da mesma forma, as máscaras 45°, vertical e -45° respondem às suas respectivas orientações.

Isso implica que ao aplicar a máscara horizontal em uma determinada imagem, as linhas horizontais serão destacadas e as demais serão suavizadas. O mesmo se aplica para as demais orientações. As Figuras 11a, 11b e 11c mostram um exemplo de aplicação das máscaras

horizontal e vertical em uma imagem de origem. É possível notar que no primeiro caso, os contornos horizontais da imagem foram realçados, como era esperado. Já no segundo caso os contornos realçados foram os verticais, conforme a máscara foi definida.

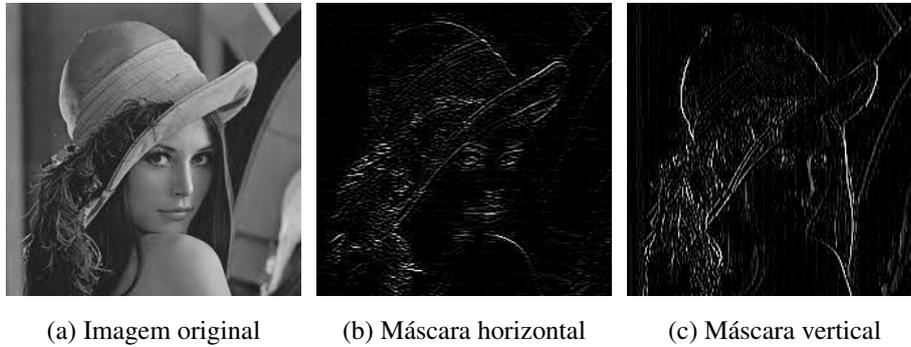


Figura 11: Exemplos de detecção de linhas

2.4.3 Detecção de bordas

Uma borda é o limite entre duas regiões com propriedades relativamente distintas de luminosidade. As bordas na imagem caracterizam os contornos dos objetos e são bastante úteis para segmentação e identificação de objetos na cena [da Silva, 2008].

Segundo [Gonzalez & Woods, 1992], embora a detecção de pontos e linhas sejam elementos de qualquer discussão sobre segmentação, a detecção de bordas é utilizada com mais frequência que elas, já que pontos e linhas isoladas não são ocorrências comuns na maioria das aplicações práticas.

A ideia por trás da maioria das técnicas para a detecção de bordas é a computação de um operador local diferencial. Ainda segundo [Gonzalez & Woods, 1992], a magnitude da primeira derivada do perfil de níveis de cinza de uma imagem pode ser usada na detecção da presença de uma borda em uma imagem, já o sinal da segunda derivada pode ser usado para determinar se um *pixel* da borda localiza-se no lado escuro ou claro da mesma. A primeira derivada em qualquer ponto da imagem é obtida usando-se a magnitude do gradiente naquele ponto e segunda derivada é obtida utilizando-se o laplaciano.

A aplicação do operador gradiente através do cálculo da derivada de primeira ordem em toda a imagem, permite realçar regiões onde a transição dos sinais de intensidade luminosa ocorre de forma abrupta, ou seja, um destaque especial para cantos e bordas dos objetos em

imagens, descartando toda a parte constante.

O operador de Sobel permite realizar o cálculo aproximado da função gradiente sobre os *pixels* de imagens digitais. O operador fornece de uma só vez efeitos de diferenciação e suavização, característica bastante atrativa desse filtro, uma vez que a derivação aumenta o ruído na imagem. As máscaras utilizadas no operador de Sobel são mostradas na Figura 12.

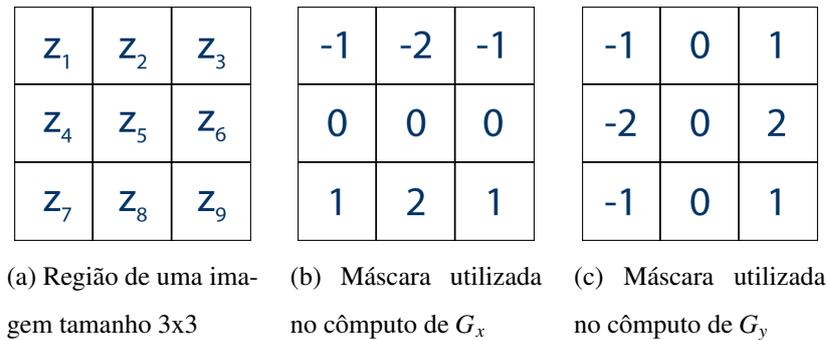


Figura 12: Máscaras utilizadas no filtro de Sobel

Fonte: [Gonzalez & Woods, 1992]

As respostas das máscaras da Figura 12 estão relacionadas a orientação do gradiente que estas calculam, ou seja, a máscara que calcula o gradiente na horizontal G_x (Figura 12b) responde melhor às bordas orientadas nesse sentido. Da mesma forma se aplica ao gradiente vertical G_y (Figura 12c).

O operador Laplaciano é utilizado de maneira semelhante ao gradiente, porém o cálculo é feito sobre a derivada de segunda ordem. Esse operador é insensível à rotação, ou seja, é capaz de realçar ou detectar bordas em qualquer direção. Porém, seu uso é restrito devido a sua grande suscetibilidade a ruído [da Silva, 2008].

2.5 Segmentação orientada a regiões

2.5.1 Crescimento de regiões por agregação de *pixels*

A segmentação orientada a regiões é uma abordagem baseada na similaridade dos níveis de cinza. Nessa abordagem, os *pixels* são agrupados em subregiões e a partir de um conjunto inicial, denominado de sementes, as regiões crescem agregando *pixels* que possuam propriedades semelhantes.

Os critérios para seleção das sementes dependem diretamente da natureza do problema.

Dois problemas fundamentais relativos à abordagem são a seleção de sementes que representem adequadamente as regiões de interesse, bem como a seleção de propriedades apropriadas para a inclusão de pontos nas várias regiões durante o processo de crescimento. Em aplicações militares de imageamento em infravermelho por exemplo, os alvos de interesse são geralmente mais quentes que o fundo e por tanto aparecem mais claros [Gonzalez & Woods, 1992].

Outro problema diz respeito aos critérios de parada. Basicamente, o crescimento de uma região deveria parar quando nenhum outro *pixel* satisfizer os critérios de inclusão naquela região. Outros critérios podem ser adicionados para aumentar o poder do algoritmo de crescimento, tais como: tamanho, similaridade entre o *pixel* candidato e os *pixels* crescidos até aquele momento e a forma da região sendo crescida.

2.5.2 Difusão e fusão de regiões

Segundo [Gonzalez & Woods, 1992], segmentar uma imagem por meio dessa técnica consiste em subdividir a imagem de entrada sucessivamente em quadrantes que satisfaçam certa propriedade. Basicamente consiste em agrupar regiões adjacentes caso sejam similares ou dividir a região caso ela seja homogênea. Durante o processo, deve-se verificar para cada quadrante se a propriedade estabelecida é satisfeita, quando isso não ocorre, o quadrante é subdividido e o processo continua. O processamento termina quando não é mais possível agrupar ou dividir as regiões.

Uma desvantagem na utilização desta técnica é que a segmentação final tende a ter regiões com formas quadradas, devido à segmentação ser feita a partir da produção de divisões quadradas da imagem original.

Outra característica dessa abordagem é que ela tende a perder regiões dentro de grandes áreas uniformes. A Figura 13 ilustra um exemplo disso. Na figura, é possível notar que alguns quadrantes maiores possuem pequenas variações de tonalidade que não foram detectadas pelo processo.

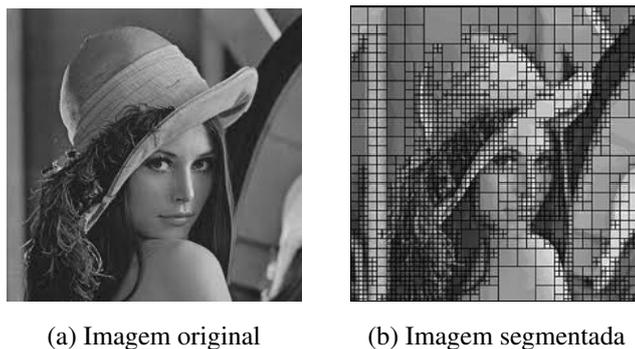


Figura 13: Resultado da segmentação por divisão e agrupamento

Fonte: [Gonzalez & Woods, 1992]

3 Dispositivos móveis

3.1 Introdução

As tecnologias de computação móvel encontram-se em franca evolução, parecem destinadas a se transformar no novo paradigma dominante da computação atual e, provavelmente, das gerações futuras [Marcal et al., 2005]. No Brasil, são mais de cem milhões de usuários de aparelhos celulares, ao passo que 22 milhões de pessoas possuem acesso a computadores e Internet.

Um dos fatores responsáveis pela disseminação e uso de aparelhos celulares é o baixo custo dos aparelhos e dos serviços de telefonia móvel em comparação aos valores dos computadores e serviços de acesso à internet [da Silva & Consolo, 2007].

A utilização de dispositivos móveis como *smartphones* e *tablets* auxilia a vida dos usuários em várias tarefas. A mobilidade desses dispositivos permite que operações possam ser realizadas em qualquer local e o tempo necessário para a obtenção de resultados seja reduzido de forma significativa, viabilizando o desenvolvimento de diversas soluções.

No problema abordado nesse trabalho, a mobilidade permite que a contagem de telhas seja realizada no próprio local de carregamento dos caminhões de transporte, o que torna o processo mais veloz. Assim, é possível aumentar a capacidade de escoamento da produção.

A propagação do uso de dispositivos móveis com maior poder de processamento abre uma nova gama de aplicações para a visão computacional, mas ao mesmo tempo adiciona novos desafios considerando as mudanças de arquitetura, usabilidade e padronização quando comparadas as aplicações desenvolvidas para computadores [Bhargava et al., 2011].

Considera-se que esse campo ainda não tem sido muito explorado e verifica-se a necessidade de adaptação de alguns algoritmos para a arquitetura móvel já que mesmo os melhores aparelhos celulares não possuem ainda a capacidade de processamento necessária para a execução em tempo real destes algoritmos [Sousa & Marengoni, 2012].

A programação para dispositivos móveis requer cuidados especiais por parte do programador, em relação às plataformas que utilizam processadores de uso geral como os computadores pessoais. Esses dispositivos possuem recursos de *hardware* limitados, logo, é necessário programar de acordo com os recursos disponíveis de forma a conseguir o melhor desempenho possível com uma maior economia de recursos.

No entanto, essa realidade vem mudando com o passar dos anos. Os dispositivos móveis tem evoluído cada vez mais seu poder de processamento e arquitetura do *hardware*. Os *smartphones* são verdadeiras caixas de ferramentas. Com diversos aplicativos disponíveis em repositórios como a *Play Store* da Google e *AppStore* da Apple, e um *hardware* com uma diversidade de sensores embutidos, tais como: acelerômetro, compasso digital, giroscópio, GPS, microfones e câmeras, tornam-se poderosas plataformas de desenvolvimento [Lane et al., 2010].

Grande parte dos dispositivos móveis atualmente possuem um *hardware* melhor que computadores de mesa de pouco tempo atrás. Esses dispositivos contam com sistemas operacionais próprios, com interfaces que possibilitam a manipulação dos sensores disponíveis no aparelho por parte das aplicações. A plataforma Android por exemplo, utiliza o *kernel* do Linux e por isso herda grande parte das características desse sistema como sistema de arquivos, *kernel*, servidores de terminais (*X server*), etc. Exemplos de características suportadas pela plataforma:

- *Framework* de Aplicação permitindo reuso de componentes;
- Máquina Virtual Dalvik otimizada para dispositivos móveis;
- Navegador *Web* Integrado baseado na engine open source WebKit;
- Gráficos Otimizados por meio de uma biblioteca de gráficos 2D; e gráficos 3D baseados na especificação OpenGL ES 1.0;
- SQLite para armazenamento de dados em formato de estruturas relacionais;
- Suporte para mídias de audio (formatos MP3, AAC, AMR), vídeo (MPEG4 e H.264) e imagens (formatos JPG, PNG, GIF);
- Telefonia GSM (dependente de *hardware*);
- *Bluetooth*, EDGE, 3G, e *WiFi* (dependente de *hardware*)
- Câmera, GPS, bússola, e acelerômetro (dependente de *hardware*)

- Poderoso Ambiente de Desenvolvimento, incluindo um emulador de dispositivo, ferramentas para depuração, analisador de memória e performance; e um *plugin* para a IDE Eclipse

A arquitetura da plataforma Android é dividida em várias camadas: *Applications*, *Application Framework*, *Libraries* e *Android Runtime*; e *Linux Kernel*. Essas camadas permitem ao desenvolvedor acessar aplicativos nativos do dispositivo ou desenvolver aplicativos próprios utilizando as APIs disponíveis na plataforma, utilizando a linguagem Java.

Outro fator que têm contribuído com o aumento da capacidade de processamento dos dispositivos é o uso do paralelismo. A grande maioria dos dispositivos modernos já apresentam processadores com mais de um núcleo de processamento. Aparelhos da Apple e da Samsung, como o Iphone 5s e Galaxy s4, por exemplo, possuem processadores de alto desempenho, um quad-core de 1,9 Ghz e um dual-core de 1,3 Ghz, respectivamente.

3.2 Trabalhos relacionados

O avanço no poder de processamento dos dispositivos móveis tem possibilitado o desenvolvimento de trabalhos cada vez mais robustos nessas plataformas, fator que tem contribuído para o avanço das pesquisas nessa área.

A visão computacional vem sendo bastante utilizada em conjunto com dispositivos móveis no desenvolvimento de soluções. A seguir, são apresentados dois trabalhos nesse sentido.

Em [Lameira et al., 2012] é apresentado um sistema de detecção e reconhecimento de objetos em dispositivos móveis que processa as informações localmente no dispositivo, ou seja, sem a necessidade de qualquer interação com servidor.

Em muitos trabalhos envolvendo dispositivos móveis, os algoritmos são executados em servidores dedicados sob a arquitetura cliente-servidor devido a limitação da capacidade de processamento desses dispositivos. A proposta do trabalho é evitar essa comunicação e com isso economizar energia do dispositivo e diminuir o tempo de resposta do sistema.

O método inicia com a aquisição da imagem. O usuário deve apontar a câmera para o objeto de interesse de modo a posicioná-lo ao centro da imagem. Após capturada a imagem é feita a detecção do objeto, ou seja, identificar o objeto e o fundo da imagem. Esse proce-

dimento é realizado para diminuir a quantidade de informação a ser processada na etapa de identificação. A detecção é dividida em duas etapas: divisão e avaliação da imagem.

Na etapa de divisão, a imagem é segmentada em pequenos blocos. A etapa de avaliação consiste em classificar cada bloco como pertencente ao fundo ou ao objeto, com base em um critério estatístico estabelecido. O critério adotado foi a variação máxima dos níveis de cinza entre blocos.

A identificação do objeto ocorre em três etapas: detecção, normalização e reconhecimento. A detecção é feita conforme foi descrito, caso algum objeto tenha sido detectado, a normalização é aplicada. A normalização consiste em extrair uma forma padrão do objeto detectado, ou seja, a partir de um retângulo desenhado ao seu redor, os espaços restantes entre o retângulo e o objeto são preenchidos até que se obtenha a forma do objeto.

A etapa final do processo é o reconhecimento do objeto. Esse processo é dividido em duas partes: descrição do objeto e classificação. A descrição é feita a partir da divisão da imagem em pequenos blocos de área 30x30. É feito um cálculo do histograma para cada bloco, representando a textura de cada área. Já o reconhecimento utiliza máquinas de vetores de suporte que após passarem pelo treinamento são utilizadas para classificar os objetos.

Em [Jung, 2012] é proposto um sistema de reconhecimento de objetos implementado para dispositivos móveis. O trabalho tem dois objetivos principais: reconhecer objetos tridimensionais em imagens bidimensionais, mesmo que os objetos estejam em diferentes perspectivas, rotacionados ou apresentem sombras; reduzir a complexidade do algoritmo para que possa ser utilizado em dispositivos móveis.

A ideia do sistema é realizar o treinamento de máquinas de vetores de suporte para classificar padrões de objetos, os quais suas características estão armazenadas localmente. O vetor de características é constituído de diferentes partes: características baseadas em cores, bordas, frequência e pontos de interesse. As características são obtidas a partir de uma análise estatística das imagens. Os pontos de interesse são detectados através do algoritmo SURF.

Na classificação dos objetos, é feito um treinamento supervisionado na máquina de vetor de suporte é com base no conjunto de características e nos pontos de interesse obtidos com o SURF.

3.3 Conclusão

O avanço no poder de processamento dos dispositivos móveis possibilitou o desenvolvimento de diversas soluções, com diferentes níveis de complexidade. A utilização do processamento digital de imagens em conjunto com as tecnologias móveis é uma ferramenta poderosa no desenvolvimento de aplicações práticas no mundo real.

O presente trabalho faz uso da tecnologia móvel a fim de agilizar a obtenção do resultado da contagem do sistema contador de telhas. Dessa forma, é possível aumentar a capacidade de escoamento da produção, já que a demora na contagem manual se mostra um gargalo dessa etapa do processo.

4 Detecção e rastreamento de objetos

4.1 Introdução

Diversos problemas cotidianos envolvendo sistemas de visão computacional requerem a localização e o reconhecimento de objetos ao longo de uma sequência de imagens ou vídeos digitais, são exemplos: vigilância automatizada, inspeção de peças de produção, cálculo da velocidade de veículos, etc.

Formalmente, define-se rastreamento como o processo de determinação da posição de certo(s) objeto(s) ao longo do tempo em uma sequência de imagens [Graciano, 2007]. Os critérios de seleção durante o rastreamento devem tratar aspectos como: robustez ao ruído, às variações de intensidade luminosa, ao nível de contraste, à dimensão das imagens e em alguns casos, ser capaz de lidar com oclusões [da Silva, 2008].

A ideia fundamental é associar características como cor ou segmentos, quadro-a-quadro na sequência de imagens. Técnicas mais recentes utilizam filtros preditivos capazes de estimar o estado de um alvo no próximo quadro da sequência. A estimativa permite determinar qual objeto detectado no próximo quadro melhor corresponde ao objeto rastreado. Com a correspondência quadro-a-quadro, é possível extrair parâmetros do movimento realizado como trajetória e velocidade [de Moraes, 2005].

As etapas de detecção e rastreamento de objetos serão discutidas nos tópicos a seguir. Primeiramente serão abordadas técnicas para detecção de objetos em imagens. Em seguida será discutido como é feito o rastreamento dos objetos detectados, ao longo da sequência de imagens.

4.2 Técnicas para detecção de objetos

Métodos eficientes para detecção de objetos representam uma importante área de pesquisa na Visão Computacional, já que muitas aplicações requerem a determinação da presença e localização de objetos em imagens [He et al., 2011].

Existem diferentes maneiras de se detectar objetos em imagens. Geralmente, a detecção é baseada na identificação de características dos objetos de interesse que estejam presentes na imagem. A seleção das características a serem utilizadas depende diretamente do problema

e é uma escolha fundamental para o sucesso da identificação.

A seguir são apresentados dois métodos utilizados na detecção automática de objetos em imagens: através de Redes Neurais Artificiais e utilizando métodos exatos.

4.2.1 Redes Neurais Artificiais

Uma das formas de identificar objetos em imagens é através de Redes Neurais Artificiais (RNA). De acordo com [Haykin, 1999], uma rede neural em sua forma mais geral é uma máquina que é projetada para modelar a maneira como o cérebro realiza uma tarefa particular ou função de interesse. Trata-se de uma unidade de processamento maciçamente paralela e distribuída, na qual é realizado um processamento simples, e tem a propensão natural para armazenar conhecimento experimental e torná-lo disponível para o uso [Lopes, 2009].

A RNA é capaz de aprender e, portanto, de generalizar. A generalização se refere ao fato de a rede neural produzir saídas adequadas para entradas que não estavam presentes durante o treinamento [Haykin, 1999]. Após a etapa de segmentação, as imagens são subdivididas em objetos constituintes, logo é possível treinar uma RNA para classificar esses objetos e assim identificar quais deles são de interesse para a aplicação.

O treinamento serve para a rede aprender aquele determinado padrão para ser capaz de identificá-lo posteriormente. O conceito de aprendizado no contexto das RNA foi definido em [Mendel & McLaren, 1970] como sendo: um processo nos quais os parâmetros livres (pesos sinápticos), da rede neural, são modificados devido à um processo de estimulação no ambiente que a rede está inserida.

Não existe nenhum algoritmo padrão para o treinamento das redes, existe um conjunto de técnicas que são usadas para gerar algoritmos de aprendizagem. O modo como a rede está estruturada internamente, a forma como seus neurônios estão interligados influenciam no modo de aprendizagem, porém a diferença entre os modos de aprendizado, é basicamente a forma pela qual são ajustados os pesos sinápticos dos neurônios [da Silva, 2010].

4.2.2 Métodos exatos

Os métodos exatos são uma alternativa para a detecção de objetos. Esses métodos se baseiam em características peculiares dos objetos para a detecção, tais como: comprimento, perímetro, raio, distância ao centro de massa, altura, rotação, etc. Podem ser utilizadas uma

ou mais características combinadas para definir se determinado objeto pertence ou não a uma determinada classe.

Esse método é recomendado quando os objetos de interesse possuem características bem definidas e constantes. Pode ainda servir de auxílio para outros métodos mais complexos, pois como sua aplicação se baseia em operações simples sobre imagens, apresenta um baixo custo computacional. Logo, pode ser utilizado como etapa complementar para evitar o uso desnecessário desses métodos em determinadas situações.

É importante observar que as características utilizadas nos métodos exatos são relativas e variam de acordo com alguns aspectos: distância, posicionamento e ângulo do(s) objeto(s) em relação à câmera. Por isso, outro requisito fundamental para o uso desse método é que o posicionamento da câmera em relação à cena seja fixo e previamente conhecido, o que geralmente ocorre em ambientes controlados.

4.2.3 Trabalhos relacionados

Em [Cruz et al., 2013] é proposto um sistema de reconhecimento e detecção de objetos baseado na forma e padrão de cores dos objetos. Para a classificação é utilizada uma RNA com 52 neurônios na camada de entrada, 20 na camada oculta e 5 na camada de saída, como mostra a Figura 14.

O sistema é dividido em três etapas principais: processamento da imagem, processamento da RNA e interpretação.

Inicialmente é feito o pré-processamento da imagem. Para isso, a imagem é convertida em escala de cinza e realizada uma binarização. Em seguida são extraídas informações sobre as bordas do objeto e aplicada uma segmentação dinâmica, de forma a selecionar apenas o menor retângulo capaz de envolver o objeto de interesse. O restante do fundo da imagem é descartado.

No passo seguinte, o resultado da segmentação é processado por duas equações que calculam a distribuição de cores da imagem. A imagem é então redimensionada em uma matriz 25x25 e as linhas são somadas com as colunas, resultando em 50 números distintos.

Após o processamento inicial da imagem ocorre o treinamento e classificação dos objetos através da RNA. Os 52 neurônios de entrada da rede são alimentados pelos 50 números gerados na etapa anterior e os resultados das duas equações. As equações trazem informações

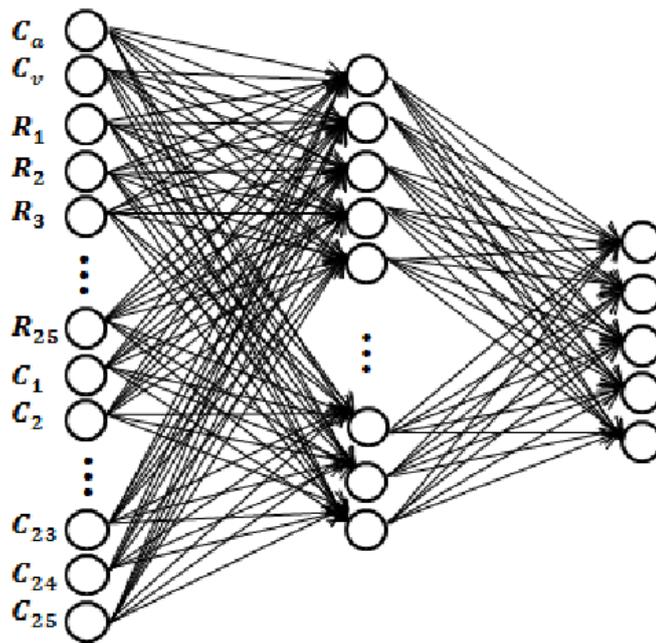


Figura 14: Estrutura da RNA utilizada na identificação dos objetos

Fonte: [Cruz et al., 2013]

relativas as cores do objeto, já os números estão relacionados à sua forma.

Os 5 neurônios da camada de saída representam um resultado binário da rede. Cada combinação de binários representa um determinado objeto associado durante a fase de treinamento. Na fase de treinamento da rede são informados os conjuntos de entrada juntamente com as respectivas saídas desejadas.

Por fim a fase de interpretação consiste em associar cada saída binária aos objetos utilizados durante o treinamento. Segundo o trabalho, os testes realizados mostraram que o método é eficaz mesmo quando os objetos são rotacionados em relação à posição utilizada no durante o treinamento.

Já em [Wahi et al., 2008] é apresentado um método híbrido de extração de características para reconhecimento de objetos utilizando RNA. No método proposto, as imagens são pré-processadas e são calculadas a Transformada Discreta de Fourier (do inglês, *Discrete Fourier Transform* - DFT) e a combinação da DFT com a Transformada Discreta de *Wavelet* (do inglês, *Discrete Wavelet Transform* - DWT). Os valores obtidos são utilizados como entrada para a RNA, Perceptron de Múltiplas Camadas treinada com o algoritmo *Backpropagation*. A RNA classifica os objetos como pertencentes ou não de uma determinada categoria.

O processo é dividido em quatro etapas: pré-processamento, extração de características, rotação dos objetos, treinamento da rede e classificação.

Na etapa de pré-processamento são realizadas as operações de conversão da imagem para escala de cinza e binarização. Caso a imagem contenha ruídos, podem ser aplicados filtros como o da mediana. Ainda nessa etapa, a imagem é cortada e apenas os objetos são extraídos. Após isso, é feito um escaneamento horizontal e vertical na imagem, de forma a extrair apenas as bordas dos objetos.

Na etapa seguinte são extraídas características dos objetos através do cálculo da 2-D DFT e da 2-D DWT. No método proposto, a imagem é redimensionada para 128x128 e aplicada a transformada de Fourier. Da magnitude resultante são calculadas a média e o desvio padrão. Esse cálculo é feito para cada objeto e todos são rotacionados de 0 a 360° divididos em intervalos de 5°.

Os dados obtidos são utilizados como entrada para a RNA que é treinada de forma supervisionada através do algoritmo *backpropagation*. Após isso, a rede deve ser capaz de classificar as classes dos objetos, mesmo quando rotacionados.

A Figura 15 mostra alguns objetos utilizados na simulações do trabalho. Na Figura 15a tem-se os objetos utilizados na simulação, essa é a imagem original após convertida para escala de cinza. A Figura 15b mostra o resultado final do processo, após realizadas todas as operações, treinamento e classificação dos objetos. Já a Figura 15c traz um exemplo do primeiro objeto rotacionado em várias posições.

4.3 Técnicas para rastreamento de objetos

Rastrear objetos é uma tarefa árdua, as dificuldades envolvidas podem surgir devido a características como: movimento abrupto dos objetos, mudança de aparência tanto do objeto quando da cena, estrutura do objetos não rígidas, oclusões objeto à objeto e objeto à cena e movimento da câmera [Yilmaz et al., 2006].

O rastreamento de objetos é uma etapa fundamental no processo de contagem. A contagem automática de objetos é feita a partir de uma sequência de imagens (vídeo) como entrada e o processamento dessas imagens é realizado *frame a frame*. Logo, é necessária uma forma de armazenar os estados dos objetos ao longo do tempo e isso é feito através do rastreamento. Dessa forma, é possível correlacionar objetos entre imagens e evitar que um mesmo objeto



(a) Imagem original



(b) Imagem final



(c) Exemplo de objeto rotacionado

Figura 15: Exemplo de objetos utilizados na simulação

Fonte: 15

seja contado mais de uma vez.

O rastreamento pode ser realizado por predição, quando os objetos já foram previamente detectados e se tem conhecimento prévio de algumas das suas características. O Filtro de Kalman [Kalman, 1960] é um dos filtros preditivos utilizados para estimar a posição em que o objeto de interesse irá ocorrer no próximo quadro e com isso diminuir o custo computacional da detecção. Outra técnica desenvolvida para o rastreamento visual de objetos, denominada *Condensation* [Nummiaro et al., 2003], é tipicamente usada para o rastreamento de contornos [de Moraes, 2005]. A seguir serão abordados essas duas técnicas de rastreamento.

4.3.1 Filtro de Kalman

Em 1960, R. E. Kalman desenvolveu uma solução recursiva para o problema de filtragem linear de dados discretos. Desde então, o filtro de Kalman tem sido alvo de extensos estudos e aplicações. O filtro de Kalman é amplamente utilizado em aplicações da engenharia, con-

troles modernos, navegação e robótica móvel tendo como foco o rastreamento de objetos em movimento [da Silva, 2008].

Teoricamente, o filtro é um estimador para o problema Gaussiano-linear-quadrático que é o problema da estimação dos estados instantâneos de um sistema linear dinâmico perturbado por um ruído Gaussiano branco usando-se medições linearmente relacionadas aos estados e também corrompidas por ruído branco. Quando o sistema é linear e variáveis aleatórias Gaussianas são representações apropriadas para os estados do sistema, o Filtro de Kalman é uma ótima escolha para a estimação dos estados [de Moraes, 2005].

O filtro de Kalman possui duas etapas principais: predição e correção. A etapa de predição consiste em prever a posição de determinado objeto em um instante t , conhecendo-se sua posição no instante $t-1$. A cada incremento de tempo discreto, um operador linear é aplicado ao estado para gerar um novo estado, com algum ruído agregado. Na etapa de correção são utilizadas as informações no instante atual para atualizar a nova posição do objeto.

Embora esse filtro tenha sido aplicado originalmente em sistemas lineares, ele pode ser estendido para aplicações de sistemas não lineares por meio de uma expansão da série de Taylor. A expansão lineariza as funções em torno da estimação corrente usando as derivadas parciais do processo e das funções de medição para calcular as estimações. O filtro resultante é chamado de Filtro de Kalman Estendido [de Moraes, 2005].

Mais informações sobre o filtro de Kalman podem ser encontradas em [Welch & Bishop, 2007], [Simon et al., 2010] e em [Faragher et al., 2012].

4.3.2 CONDENSATION (*Conditional Density Propagation*)

Outros métodos utilizados no rastreamento de objetos são os filtros de partículas (ou filtração *bootstrap*). O algoritmo CONDENSATION [Isard & Blake, 1998] é um filtro de partículas desenvolvido para o rastreamento visual bastante utilizado em problemas relacionados à Visão Computacional.

O algoritmo CONDENSATION é uma abordagem probabilística utilizada no rastreamento de objetos onde o plano de fundo se assemelha ao objetos de interesse, em cenas conhecidas como ambíguas [de Moraes, 2005].

O CONDENSATION consegue representar múltiplas hipóteses simultâneas propagando Funções de Distribuição de Probabilidades não Gaussianas e potencialmente multimodais

[Isard & Blake, 1998]. Para isso, o algoritmo se baseia na técnica chamada de *factored sampling* utilizada para localizar objetos conhecidos em imagens estáticas.

Essa técnica gera um conjunto aleatório de N possíveis configurações da imagem chamadas de partículas, onde são descritos os parâmetros de objetos hipotéticos como localização, forma e orientação. Cada partícula possui uma probabilidade de representar a imagem, tornando o conjunto uma aproximação da distribuição [Isard & Blake, 1998]. A estratégia é utilizada de forma semelhante, sendo que para cada quadro da sequência, ou seja, cada instante de tempo t , um novo conjunto de N partículas é criado com base em amostras do conjunto do instante anterior e levando-se em conta o movimento dos objetos rastreados.

É importante ressaltar que o algoritmo CONDENSATION originalmente não é capaz de detectar novos objetos que apareçam na cena, rastreando apenas aqueles presentes no passo de inicialização. Esse fato ocorre porque o passo de medição é utilizado apenas para calcular as probabilidades de cada partícula e não afeta os objetos rastreados diretamente. Então, se um novo objeto entrar na cena, ele não poderá ser rastreado [de Moraes, 2005].

4.3.3 Trabalhos relacionados

Em [Patel & Thakore, 2013] foi desenvolvido um sistema de detecção e rastreamento de objetos utilizando o filtro de Kalman. Segundo o autor, o sistema é capaz de rastrear qualquer objeto, no entanto para inicialização do sistema são necessários alguns pré-requisitos: o usuário deve informar em qual quadro se deseja iniciar o rastreamento do objeto e selecionar uma máscara (conjunto de *pixels*) onde ele se encontra. Ou seja, uma das limitações do trabalho é que apenas o objeto selecionado é rastreado, nenhum outro que surja posteriormente na cena será considerado.

O algoritmo funciona da seguinte forma: após a etapa de inicialização, o plano de fundo da cena é subtraído a fim de identificar as regiões de movimento do *frame*. Para cada região encontrada, é calculada a menor distância entre essas mesmas regiões em relação ao quadro anterior. Após selecionadas as regiões, seus centróides são calculados desenhado uma linha ligando-os a fim de representar a trajetória do objeto no vídeo.

Para validar o trabalho, foram realizados experimentos com conjuntos de vídeos contendo carros e pessoas, dois tipos de objetos bastante utilizados em trabalhos de pesquisa.

Já em [Lu et al., 2012] é proposto um *framework* para rastreamento de objetos baseado

em um filtro de partículas de fusão de múltiplas características. No modelo proposto, o rastreamento é realizado através da fusão do histograma de cores com o detector de bordas de Harris-Stephens.

O *framework* pode ser dividido em duas partes: o rastreamento inicial do objeto a partir das informações de cores e bordas e o rastreamento exato com o filtro de partículas.

Na primeira etapa são extraídas informações sobre o histograma de cores e inicialização do vetor de características das bordas, através da aplicação do operador de Canny. Após isso, os contornos do objeto inicial são utilizados como referência para guiar a propagação das partículas no *frame* atual. Então, informações sobre características de cores e bordas são integradas por filtros bayesianos para produzir uma representação do objeto de interesse.

A segunda etapa consiste em utilizar o filtro de partículas para rastrear o objeto definido na etapa anterior. Os contornos da etapa inicial servem para orientar a propagação das partículas, o que possibilita uma redistribuição mais efetiva.

4.4 Conclusão

Deteção e rastreamento são etapas complementares e fazem parte do processo de contagem automática de objetos. Existem diferentes maneiras de deteção e rastreamento, de maneira que o que definirá a melhor forma de associar essas etapas para obter sucesso na contagem são as características associadas aos objetos e cena em questão.

No presente trabalho onde o estudo de caso destina-se a desenvolver um sistema capaz de contar telhas de forma automática, algumas características do problema foram determinantes na escolha dos algoritmos utilizados.

Na etapa de deteção, o fato da cor do fundo da cena (tonalidades escuras) se diferenciar de forma significativa das cores das telhas (tonalidades avermelhadas), optou-se por utilizar uma segmentação com base em uma binarização. O algoritmo específico escolhido será explicado em detalhes posteriormente.

Já no rastreamento foi levado em consideração a homogeneidade do movimento relativo das telhas em relação à câmera. Aspectos referentes às cores das telhas em relação ao fundo também foram considerados no desenvolvimento do algoritmo.

5 Sistema proposto

Neste capítulo é apresentado um sistema automático de contagem de telhas, realizada através do processamento de imagens capturadas por um dispositivo móvel. O dispositivo captura as imagens, processa as informações e informa a quantidade total de telhas.

O trabalho desenvolvido possui um caráter inovador na indústria da cerâmica vermelha, uma vez que os mecanismos atuais de contagem de telhas se limitam ao controle de produção, ou seja, a contagem é feita ainda durante a etapa de fabricação. O problema nesse tipo de abordagem é que a quantidade de telhas fabricadas difere significativamente da quantidade de telhas ao final do processo, devido as quebras. Para esclarecer melhor essa diferença, a seguir é apresentada uma visão geral do processo de fabricação de telhas.

O processo de fabricação de telhas é dividido em quatro etapas principais: modelagem, corte, secamento e cozimento. A etapa de modelagem consiste em dar forma à matéria prima (a argila). Para isso, a argila é aguada, misturada e atravessa uma máquina denominada de extrusadora. Essa máquina é responsável por empurrar a argila através de uma boquilha, para que ela adquira a forma semi-circular característica da telha.

O barro extrusado é levado por uma esteira até a máquina de corte, conhecida como cortadeira transversal de telhas. Essa máquina realiza os cortes laterais e inferiores proporcionando à telha suas dimensões finais. É comum ocorrer erros durante esse processo, pois a lâmina utilizada no corte se desgasta e se rompe periodicamente. Logo, caso a contagem das telhas seja feita durante essa etapa, não contabilizará o total de telhas produzidas corretamente. Ainda na etapa de corte, é comum ocorrer quebras de telhas na transferência da esteira para as grades, pois o processo é feito manualmente por operários.

Após modeladas e cortadas as telhas precisam ser secadas antes do cozimento. Para isso, as telhas são colocadas em estufas ou dispostas ao sol durante um período de dois a três dias. Semelhante à etapa anterior, nesse processo é comum ocorrer quebras de telhas durante o transporte entre as grades e os fornos.

A última etapa do processo é o cozimento. As telhas secas são transportadas para os fornos onde são cozidas durante um determinado tempo, sob o controle de medidores de temperatura (pirômetros). Após cozidas, as telhas estão prontas para serem transportadas pelos caminhões.

Como explicado, durante o processo de fabricação ocorre com frequência a quebra do material. Por isso, um mecanismo capaz de contar telhas após toda a etapa de fabricação, ou seja, quando já empilhadas e prontas para transporte, permite calcular de maneira precisa a produção real do material.

5.1 Visão geral do sistema

O sistema desenvolvido é composto pelos seguintes processos: aquisição das imagens, pré-processamento, segmentação, reconhecimento, rastreamento e contagem. A Figura 16 mostra o fluxo dos processos.

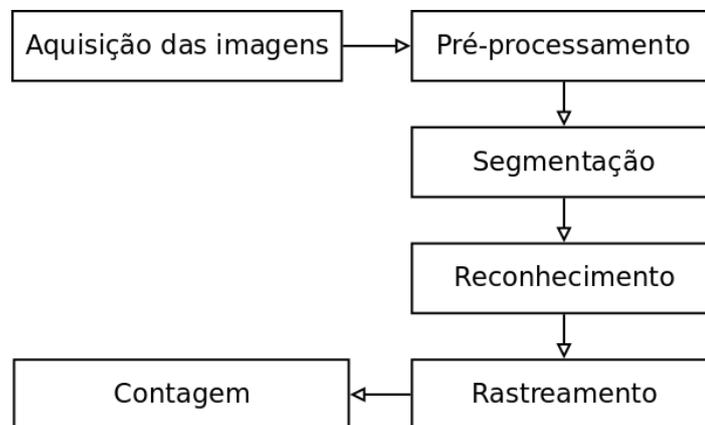


Figura 16: Fluxo de processos do sistema contador de telhas

Na etapa de aquisição, as imagens são capturadas pela câmera de um dispositivo móvel, que é deslocado ao longo de uma fileira de telhas, a uma determinada distância. Em seguida, as imagens são pré-processadas e é realizada uma segmentação, onde ocorre a separação do fundo e objetos presentes na imagem. No reconhecimento, é feita a classificação dos objetos segmentados a fim de identificar quais representam telhas.

As operações descritas são processadas *frame a frame*, ou seja, em cada uma das imagens capturadas, separadamente. Logo, é necessário fazer uma correlação entre os objetos detectados nas imagens, tornando possível a identificação de uma mesma telha ao longo da sequência de imagens. Esse procedimento é realizado através do rastreamento dos objetos.

A etapa de rastreamento é fundamental para evitar que uma mesma telha seja contada mais de uma vez. Quando uma nova telha surge no vídeo, ela é rastreada até o momento em que desaparece totalmente. Nesse instante, o contador de telhas é incrementado em uma

unidade. No final do processamento, o contador informa a quantidade total de telhas. As seções seguintes mostram em detalhes cada um dos processos descritos.

5.2 Aquisição das imagens

O sistema de contagem desenvolvido nesse trabalho foi projetado para contar telhas após o processo de produção, mais especificamente na hora do carregamento dos caminhões. Nas cerâmicas (ou olarias) o carregamento dos caminhões é feito camada por camada. Em cada camada, são formadas várias fileiras de telhas até que se preencha toda a área do caminhão. Após isso, as camadas vão sendo empilhadas umas sobre as outras até atingir a altura máxima disponível no veículo. A Figura 17 mostra um exemplo de caminhão carregado com telhas.



Figura 17: Caminhão carregado com telhas

As imagens são capturadas através da câmera de um dispositivo móvel, deslocado ao longo de cada fileira de telhas presente em cada camada. O dispositivo deve ser posicionado paralelamente à fileira, de modo que sua câmera capture a vista superior das telhas, formando um ângulo de aproximadamente 180° com elas. A Figura 18 mostra um exemplo de imagem capturada pela câmera do dispositivo.

O dispositivo deve ser descolado em um único sentido, do início ao fim de cada fileira e mantido a uma distância das telhas de modo a enquadrar apenas uma fileira por vez. Após a captura de cada fileira, há uma pausa na contagem para que o dispositivo seja posicionado na fileira seguinte.



Figura 18: Exemplo de imagem capturada pela câmera

5.3 Pré-processamento

No pré-processamento são realizadas operações para extrair apenas as informações úteis a serem utilizadas no processo de segmentação. No caso da contagem de telhas, essa etapa tem como objetivo descartar os *pixels* referentes ao fundo da imagem e filtrar objetos, com base em suas componentes de cores.

A primeira operação realizada é a conversão da imagem em escala de cinza, conforme discutido na Seção 2.2.1. A Figura 19 mostra um exemplo de aplicação dessa operação em uma imagem. Uma das vantagens da conversão da imagem em escala de cinza é reduzir o custo computacional necessário para manipular essas imagens.

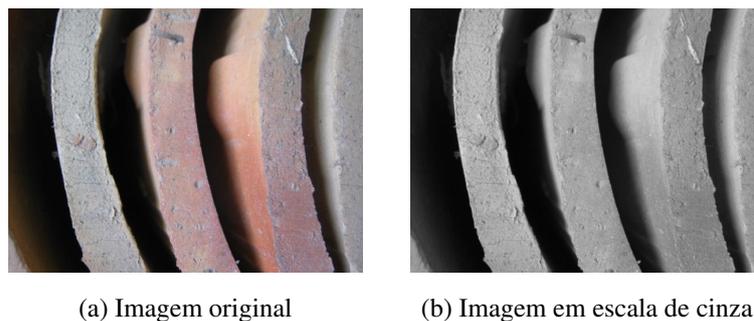


Figura 19: Conversão em escala de cinza

Após a conversão da imagem para tons de cinza, é realizada uma limiarização (ou binarização). Essa operação é responsável por filtrar os objetos presentes na imagem de acordo com um limiar estabelecido. O limiar utilizado foi calculado com base na característica de cor avermelhada das telhas. Após vários testes realizados, os limiares que geraram resulta-

dos satisfatórios foram os valores 90 para o limite inferior e 255 para o superior. A Figura 20 mostra uma imagem limiarizada com esses valores.

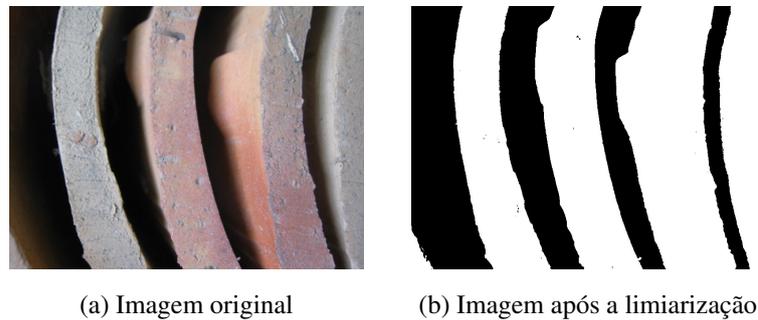


Figura 20: Exemplo de limiarização

Após a binarização, é comum que as telhas apresentem pequenos pontos pretos em sua superfície. Isso ocorre devido a textura da telha apresentar certa rugosidade. Apesar de aparentemente lisa, a superfície da telha contém pequenos furos, alguns dos quais se tornam mais evidentes após o processo de binarização.

Problemas durante a aquisição da imagem também podem gerar pequenos pontos nos objetos. Para eliminar esses ruídos, são utilizadas as operações de dilatação e erosão da imagem.

A primeira operação aplicada é a dilatação. Ela tem como objetivo preencher esses furos e assim eliminar o ruído. A Figura 21 mostra um exemplo de imagem após a aplicação da dilatação.

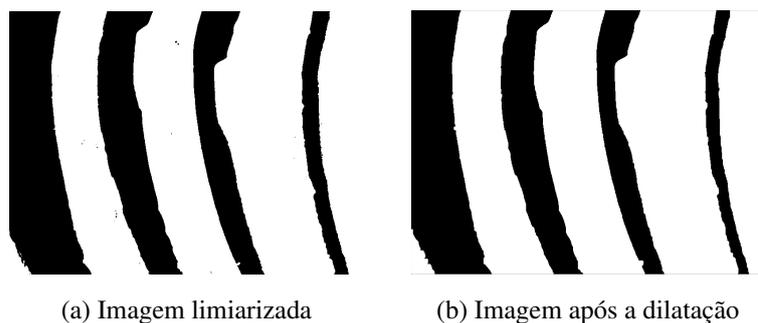


Figura 21: Exemplo de dilatação

A dilatação resulta em um pequeno aumento na área das telhas, e para evitar que telhas que estejam bem próximas umas das outras se toquem, é aplicada uma erosão, logo após a dilatação. A erosão compensa o aumento da área da telha por parte da dilatação, diminuindo

a área ao seu tamanho anterior. Isso aumenta o espaço entre as telhas, garantindo que elas não se toquem e prejudiquem a segmentação.

5.4 Segmentação

Após o pré-processamento a imagem resultante possui apenas duas componentes de cores. A cor preta e branca, representando o fundo e os objetos, respectivamente.

Na etapa de segmentação é feita a separação entre os objetos e o fundo da imagem, e armazenadas as coordenadas dos pontos de cada objeto em estruturas de dados. Como a imagem limiarizada não apresenta variação nas cores de preenchimento dos objetos, pois todos são representados pela cor branca, somente as coordenadas das bordas dos objetos são armazenadas.

O algoritmo utilizado para a extração das bordas dos objetos foi o desenvolvido em [Suzuki & Abe, 1985]. O artigo apresenta duas versões do algoritmo: a primeira realiza a extração das bordas criando uma hierarquia, onde é possível identificar objetos dentro de outros objetos. Já segunda versão extrai apenas as bordas externas os objetos, ignorando o conteúdo dentro dele. No caso das telhas, a utilização da segunda versão é mais apropriada, pois evita que sejam detectados possíveis buracos que persistam nas telhas mesmo após o pré-processamento.

O algoritmo realiza uma varredura na imagem binarizada, detecta as bordas externas das telhas, atribui um rótulo único a cada uma e armazena as coordenadas das bordas de cada telha em um vetor. Cada índice do vetor representa uma telha e é formado por uma lista contendo os *pixels* da borda. A Figura 22 mostra um exemplo de segmentação das bordas das telhas. As linhas azuis representam o resultado da segmentação.

5.5 Reconhecimento

A etapa de reconhecimento é responsável por selecionar os objetos que são telhas e descartar os demais objetos que possam surgir na imagem. Diferentes características podem ser utilizadas na seleção dos objetos. No presente trabalho, foram utilizados o perímetro do objeto e a largura do menor retângulo circunscrito ao objeto.

A telha convencional, denominada de telha colonial, possui um tamanho padrão: 57cm

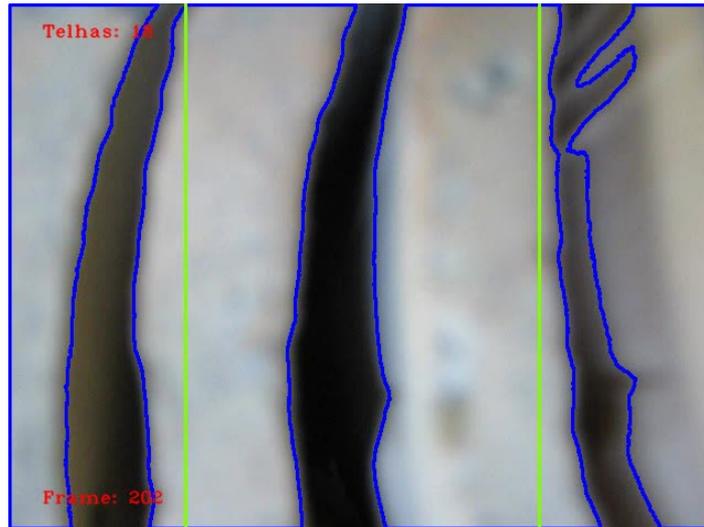


Figura 22: Segmentação das bordas das telhas

de comprimento, 21cm de largura na ponta superior e 18cm na ponta inferior. A Figura 23 mostra o desenho de uma telha colonial com suas respectivas dimensões.

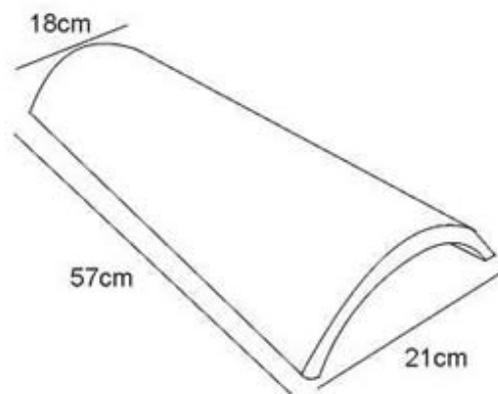


Figura 23: Dimensões da telha colonial

As telhas são empilhadas com a ponta menor para baixo e a maior para cima. Foi considerada uma distância média da câmera do dispositivo móvel, até o topo das telhas, de 5cm. Com base nisso, estabeleceu-se uma faixa de valores para a variação do perímetro do topo da telha. Os limites inferiores e superiores adotados foram, respectivamente, 800 *pixels* (21cm) e 1200 *pixels* (31cm). Isso significa que, qualquer objeto com perímetro fora dessa faixa é

automaticamente descartado.

O segundo critério adotado foi a largura do menor retângulo capaz de envolver o objeto. O retângulo é calculado a partir dos pontos mais extremos da telha, ou seja, a partir dos pontos mais a esquerda, direita, acima e abaixo, traça-se um retângulo circunscrito ao objeto. A largura do retângulo é utilizada como características de seleção da telha. As Figuras 24a e 24b mostram um exemplo dos seletores utilizados, perímetro do topo da telha e largura do retângulo, respectivamente.

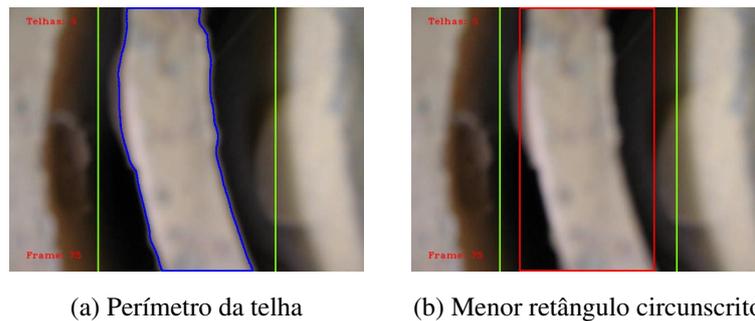


Figura 24: Seletores utilizados no reconhecimento

5.6 Rastreamento

A etapa de rastreamento consiste em identificar cada uma das telhas ao longo da sequência de imagens, ou seja, dada uma sequência de *frames*, identificar o deslocamento de uma determinada telha nessa sequência.

Como descrito na Seção 5.2, o deslocamento da câmera captura as imagens em apenas um sentido. Com base nisso, e visando reduzir a complexidade computacional do rastreamento, foi desenvolvido um algoritmo capaz de correlacionar telhas entre *frames* consecutivos, ou seja, identificar o deslocamento de cada telha *frame a frame* e atualizar as informações de seus posicionamentos em relação ao plano geral.

Em sistemas de contagem automática de objetos por processamento de imagens, é comum a determinação de uma região de contagem (RC). Trata-se de uma área específica da imagem onde os objetos são rastreados. Essa área também serve de referência para determinar a entrada e saída dos objetos e assim realizar a contagem. O trabalho desenvolvido utiliza uma RC delimitada por duas linhas verticais ao longo de toda a altura da imagem. A RC consiste na área entre as duas linhas. A Figura 25 mostra a delimitação dessa região.

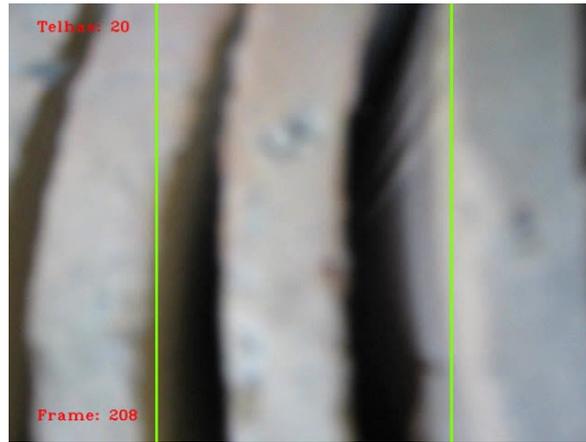


Figura 25: Região de contagem

O sistema de coordenadas adotado no trabalho é orientado da esquerda para a direita e de cima para baixo. De maneira que o ponto (0,0) se localiza na extremidade superior esquerda da imagem e os valores crescem conforme à orientação definida. Desse modo, as telhas que estiverem mais à esquerda do vídeo possuem coordenadas de menor valor.

A câmera é movimentada da esquerda para a direita, portanto o movimento relativo das telhas ocorre no sentido contrário, ou seja, da direita para esquerda. Conforme a câmera se desloca, as coordenadas das telhas diminuem a cada *frame*. Esses aspectos são relevantes para o funcionamento do algoritmo.

O algoritmo desenvolvido foi dividido em duas partes: identificação das telhas a serem rastreadas e atualização do posicionamento das telhas. A Figura 26 mostra um fluxograma da primeira parte do algoritmo.

O algoritmo inicia recebendo como entrada um vetor contendo todas as telhas detectadas na fase de reconhecimento, descrita na Seção 5.5. O algoritmo percorre o vetor e a cada iteração, ou seja, para cada telha identificada, são realizadas as seguintes operações: cálculo do menor retângulo circunscrito (MRC) capaz de envolver todo o objeto, calculado a partir das extremidades das bordas da telha e cálculo da linha central que divide o retângulo ao meio, na direção vertical (LCMRC). É feito então um teste com a posição da LCMRC, caso ela esteja dentro da região de contagem, a telha é incluída no vetor de telhas presentes na imagem. A Figura 27 mostra um exemplo de imagem contendo os elementos descritos. As duas linhas verdes representam a RC, as linhas vermelhas o MRC, as azuis as bordas da telha e as pretas a LCMRC.

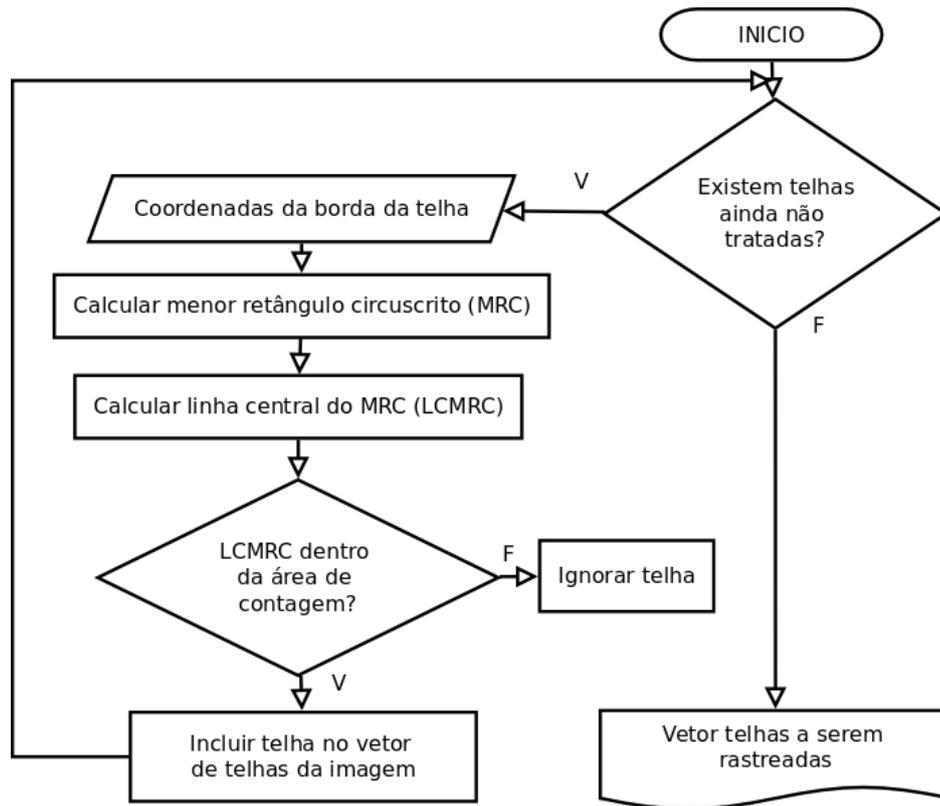


Figura 26: Fluxograma - telhas a serem rastreadas

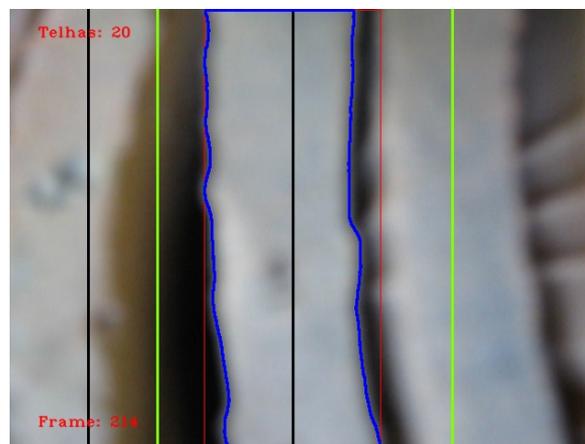


Figura 27: Elementos do algoritmo de contagem

Na segunda parte do algoritmo é feito o rastreamento e atualização do posicionamento das telhas. O resultado da primeira parte do algoritmo é um vetor com as telhas a serem rastreadas, ou seja, as telhas que estejam dentro da região de contagem. Além desse vetor, utiliza-se um outro para armazenar as telhas que já estão sendo rastreadas. A segunda parte do algoritmo, baseia-se na manipulação desses dois vetores. O Algoritmo 1 mostra o pseudo-

código.

Algoritmo 1 Rastreamento e contagem de telhas

Entrada: Vetor de telhas a serem rastreadas (TAR)

Vetor de telhas sendo rastreadas (TER)

Saída: Quantidade total de telhas

início

```

para cada Telha sendo rastreada (TER) faça
  menorQue = 0;
  para cada Telha a ser rastreada (TAR) faça
    se LC de TER < LC de TAR então
      menorQue++;
    senão
      Saia do laço
    fim
  fim
  se menorQue == Tamanho(TAR) então
    Incrementar total de telhas em uma unidade;
  senão
    Incluir telha em TER
  fim
fim
fim
  
```

A ideia geral do algoritmo é que, a cada iteração, seja feita uma comparação entre os objetos que estão sendo rastreados e os objetos presentes no *frame* atual. Esse procedimento é executado pelo encadeamento dos laços de repetição presentes no Algoritmo 1.

O vetor de telhas a serem rastreadas tem escopo local, ou seja, é declarado dentro do laço de repetição. Por isso, a cada *frame* ele é inicializado com o resultado da etapa de reconhecimento. Já o vetor de telhas que estão sendo rastreadas tem escopo global, pois permanece durante todo processo de contagem. Durante toda a descrição do algoritmo, considera-se que a posição da telha é representada sempre pela posição da linha central de seu retângulo circunscrito, ou seja, a LCMRC. Para o algoritmo, cada telha passa então a ser

apenas uma linha vertical, sua posição é dada pela coordenada do eixo x .

A Figura 28 mostra um esquema ilustrando a movimentação de uma telha de um *frame* para o outro. A imagem apresenta os seguintes elementos: o retângulo representa o *frame*; as linhas externas a RC; as linhas mais grossas representam as telhas, a pontilhada é a posição no *frame* anterior e a cheia a posição no *frame* atual.

Como dito anteriormente, a movimentação relativa das telhas ocorre da direita para esquerda. Logo, a cada *frame*, a posição das telhas (coordenada x da LCMRC) diminui.

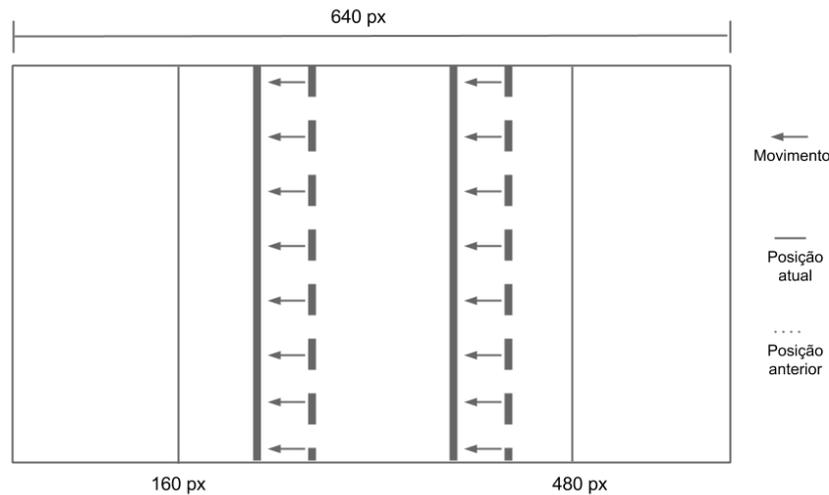


Figura 28: Movimentação das telhas

O algoritmo realiza então um teste, para cada telha sendo rastreada, identificar se sua posição é menor que a de todas as telhas presentes no *frame* atual. Caso ela seja menor de todas, significa que ela telha saiu da região de contagem, pois como o movimento relativo das telhas decrementa suas posições a cada *frame*, deveria haver pelo menos uma telha à sua esquerda, que no caso, seria ela própria, na posição atual. Quando isso ocorre, o contador total de telhas é incrementado em uma unidade, pois a telha atravessou a região de contagem. A Figura 29 ilustra esse caso.

A figura mostra três telhas denominadas de TE1, TE2 e TE3. No *frame* anterior, TE1 e TE2 estavam sendo rastreados, pois estavam dentro da RC. Já TE3 ainda não havia surgido para o algoritmo, pois estava fora da RC. No *frame* seguinte, as telhas se moveram para suas novas posições, TE1 deixou a RC, TE3 entrou na região e TE2 continuou dentro. O algoritmo varre então o vetor de telhas sendo rastreadas (TER) e, ao testar a posição de TE1, constata-

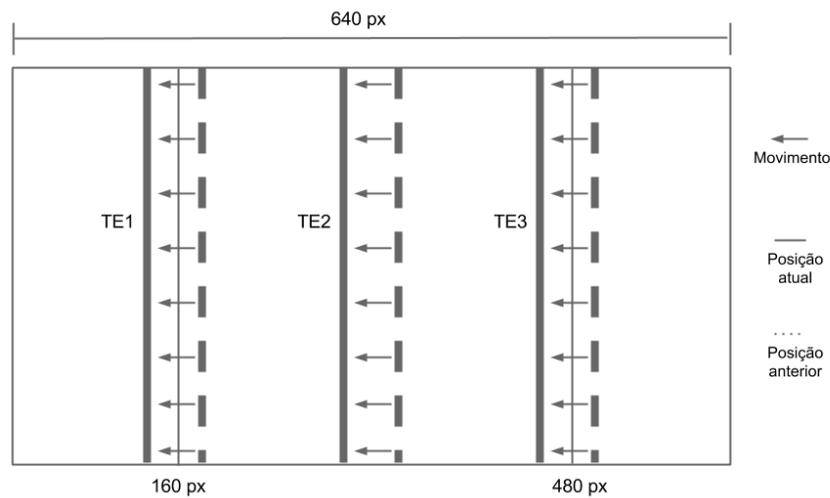


Figura 29: Exemplo de movimentação das telhas

se que é menor que todas as posições das telhas do *frame* atual (TAR), pois a posição atual de TE1 encontra-se fora da RC, logo, não é considerada. O algoritmo incrementa o contador em uma unidade, pois TE1 entrou e saiu da RC. As demais telhas TE2 e TE3 continuam no vetor de telha sendo rastreadas, a posição de TE2 é atualizada no vetor e TE3 é adicionado a ele.

5.7 Contagem

A etapa de rastreamento incrementa um contador cada vez que uma telha atravessa a região de contagem. A contagem é feita verificando o valor dessa variável no final do processamento da sequência de imagens.

6 Testes e Resultados

Nesse capítulo são apresentadas informações sobre o *hardware* e o *software* utilizados nos testes e obtenção dos resultados.

6.1 Aplicativo desenvolvido

O sistema foi primeiramente implementado para plataforma *desktop* e posteriormente migrado para a plataforma móvel. A versão *desktop* foi desenvolvida utilizando a linguagem de programação C++, e a versão para dispositivos móveis, na plataforma Android (Java). Em ambas as versões, foi utilizada a biblioteca OpenCV para auxiliar as operações envolvendo processamento digital de imagens.

OpenCV é uma biblioteca de Visão Computacional de código aberto, escrita em C e C++, disponível nas plataformas Linux, Windows, Mac e Android. Um dos objetivos da biblioteca é fornecer uma infra-estrutura de visão de computador simples de usar que ajude as pessoas a construir aplicações de visão bastante sofisticadas rapidamente. Ela contém mais de 500 funções que abrangem diversas áreas de aplicação, incluindo: fábrica inspeção de produtos, imageamento médico, segurança, interface com o usuário, calibração de câmera e robótica [Bradski & Kaehler, 2008].

A captura das imagens foi realizada através da câmera de um *smartphone* Samsung Galaxy S III Duos, que possui as seguintes especificações técnicas: sistema operacional Android 4.1 (Jelly Bean), processador *Dual Core* de 1.2 Ghz, 8 Gb de memória interna e câmera VGA com resolução de 5 *Megapixels*.

Os vídeos foram gravados no formato MPEG-4, com codificação H.264 (MPEG4 AVC), resolução de 640x480 *pixels* e taxa média de 25 *frames* por segundo.

A fim de validar o método desenvolvido e mensurar a capacidade de contagem do sistema, foram realizados vários testes em um ambiente real de produção. Os testes foram realizados em uma cerâmica (olaria), com as telhas armazenadas no estoque da indústria. No experimento, foram utilizadas fileiras de telhas de diferentes produções, com quantidade, espaçamento e cor diferenciadas. Ao todo, foram utilizados 20 vídeos no experimento. Durante a seleção das amostras, buscou-se utilizar a maior variedade de situações possíveis, ou seja, incluir telhas fora do padrão habitual: com fissuras, pequenas deformações e coloração

escura. A Figura 30 mostra dois exemplos de telhas nessas situações.

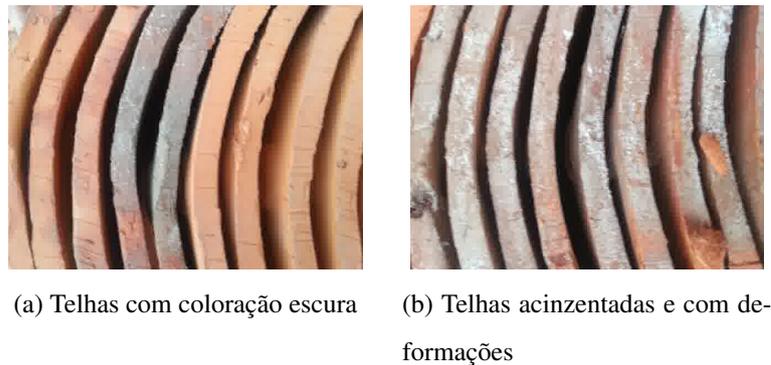


Figura 30: Telhas fora do padrão

6.2 Resultados

A Tabela 1 mostra o resultado da contagem para os vídeos utilizados nos testes. Da esquerda para a direita, as colunas representam, respectivamente: identificador do vídeo, total de telhas calculadas pelo contador, total real de telhas presente no vídeo, telhas detectadas pelo contador que não estão presentes no total real, telhas não detectadas pelo contador e diferença entre a quantidade real de telhas e as contabilizadas pelo sistema.

A diferença entre o número de telhas real presente nas fileiras e o número calculado pelo sistema ocorre de duas maneiras: quando o contador detecta telhas a mais que a quantidade real, denominado de falso positivo; e quando ele não detecta algumas telhas, chamado de falso negativo.

Durante o experimento, constatou-se que ambas as ocasiões se originam no processo de segmentação. A primeira ocorre geralmente quando há fissuras nas telhas, fazendo com que uma mesma telha seja segmentada em duas e gere um falso positivo. Já a segunda situação ocorre quando as telhas estão muito próximas umas das outras, de maneira que o sistema não consegue segmentá-las separadamente, o que gera um falso negativo.

No geral, o algoritmo apresentou um resultado positivo. Os vídeos utilizados nos testes totalizaram 933 telhas, das quais 916 foram detectadas corretamente pelo sistema. Logo, apenas 17 telhas não foram contabilizadas, o que representa uma taxa de acerto de 98,1%.

ID	Estimado	Real	Falso positivo	Falso negativo	Diferença
1	56	56	0	0	0
2	49	55	0	6	6
3	58	59	0	1	1
4	56	58	0	2	2
5	63	61	2	0	-2
6	51	55	0	4	4
7	31	30	1	0	-1
8	39	43	0	4	4
9	28	28	0	0	0
10	47	52	0	5	5
11	63	60	3	0	-3
12	55	54	1	0	-1
13	35	33	2	0	-2
14	21	20	1	0	-1
15	20	22	0	2	2
16	38	36	2	0	-2
17	21	24	0	3	-3
18	62	63	0	1	1
19	57	58	0	1	1
20	66	66	0	0	0

Tabela 1: Resultado da contagem - comparação entre quantidade de telhas real e calculada pelo sistema

A Figura 31 mostra um gráfico comparativo entre a quantidade de telhas real e contabilizada pelo sistema para cada iteração.

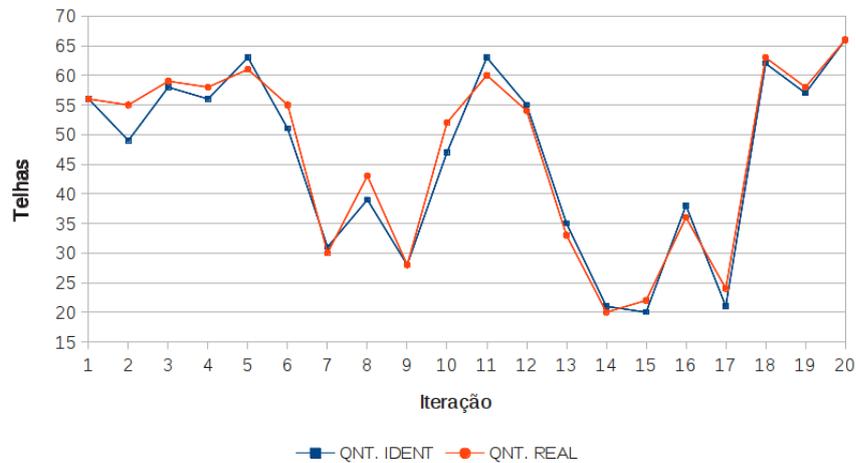


Figura 31: Gráfico comparativo resultado da contagem

Na figura, observa-se que a diferença entre os valores reais e os obtidos na contagem é pequena. Em módulo, as diferenças mínima e máxima apresentadas são de 0 e 6, respectivamente. Em três iterações (1^a, 9^a e 20^a) o sistema contabilizou a quantidade de telhas de forma exata. Nas demais, a menor taxa de acerto ocorreu na 7^a iteração, no valor de 87,5%.

7 Conclusões e trabalhos futuros

A proposta inicial do trabalho foi desenvolver um sistema automático de contagem de telhas, utilizando processamento de imagens em dispositivos móveis. A ideia era utilizar o sistema para realizar a contagem diretamente no local do transporte, na hora do carregamento do caminhão. Isso possibilita contabilizar a quantidade real de telhas que serão vendidas, diferentemente dos métodos convencionais de contagem feitas durante a etapa de fabricação do material.

O objetivo foi alcançado, foram desenvolvidas duas versões do sistema: uma versão inicial, voltada para a plataforma *desktop* que posteriormente foi migrada para a plataforma *Android*, utilizada em dispositivos móveis como *smartphones* e *tablets*. Através do sistema desenvolvido, foi possível realizar a contagem das telhas no ambiente de transporte, conforme realizado durante os testes experimentais.

O sistema se mostrou eficaz, devido a sua baixa taxa de erro, o que justifica sua utilização em alternativa à contagem manual. Além de uma maior precisão na contagem é possível obter o resultado mais rapidamente, o que possibilita uma maior capacidade de escoamento da produção do material.

7.1 Trabalhos futuros

Embora o sistema tenha se mostrado eficaz, algumas melhorias podem ser realizadas a fim de aprimorar os resultados e torná-lo mais robusto à falhas.

Na fase de reconhecimento, por exemplo, a utilização de mais critérios de seleção dos objetos, ou mesmo a utilização de técnicas de inteligência artificial como Redes Neurais poderia aumentar o poder de classificação do algoritmo, e assim garantir que somente telhas reconhecidas durante essa fase.

Outro ponto a ser explorado é a etapa de segmentação. A utilização de outras características além da cor das telhas tornaria o sistema mais eficaz na identificação e separação das telhas do fundo da imagem. Melhoraria por exemplo, a sensibilidade do sistema à variação de luminosidade da cena onde é feita a contagem.

A implementação dessas e outras melhorias a serem incorporadas ao sistema fazem parte dos trabalhos futuros a serem realizados.

Referências

- [Banon & Barrera, 1998] Banon, G. J. F. & Barrera, J. (1998). *Bases da morfologia matemática para a análise de imagens binárias*. São José dos Campos: MCT/INPE; depositado no acervo da URLib., segunda edition. Primeira edição publicada em 1994 pela IX escola de computação. This work has been supported by CNPq under contract 300966/90-3.
- [Bhargava et al., 2011] Bhargava, B., Angin, P., & Duan, L. (2011). A Mobile-Cloud Pedestrian Crossing Guide for the Blind. In *International Conference on Advances in Computing & Communication*.
- [Bradski & Kaehler, 2008] Bradski, G. & Kaehler, A. (2008). *Learning OpenCV: Computer Vision with the OpenCV Library*. O'Reilly Media.
- [Cruz et al., 2013] Cruz, J., Dimaala, M., Francisco, L., Franco, E., Bandala, A., & Dadios, E. (2013). Object recognition and detection by shape and color pattern recognition utilizing Artificial Neural Networks. In *Information and Communication Technology (ICoICT), 2013 International Conference of* (pp. 140–144).
- [da Silva, 1994] da Silva, F. V. (1994). MODELO DE EQUIPAMENTO PARA CORTE TRANSVERSAL DE TELHA CERÂMICA.
- [da Silva, 2008] da Silva, L. S. (2008). Sistema computacional para contagem automática de pessoas baseado em análise de sequências de imagens. Master's thesis, Centro Federal de Educação Tecnológica de Minas Gerais.
- [da Silva & Consolo, 2007] da Silva, M. d. G. M. & Consolo, A. T. (2007). Uso de dispositivos móveis na educação.
- [da Silva, 2010] da Silva, S. S. (2010). Sistema automático de reconhecimento e contagem de ovos da dengue. Master's thesis, Universidade Federal de Pernambuco.
- [de Moraes, 2005] de Moraes, E. F. (2005). Rastreamento e contagem de peixes utilizando filtro preditivo . Master's thesis, Universidade Federal de Minas Gerais.

- [Faragher et al., 2012] Faragher, R. et al. (2012). Understanding the Basis of the Kalman Filter Via a Simple and Intuitive Derivation. *Signal Processing Magazine*, 29(5), 128–132.
- [Feitosa, 2012] Feitosa, F. C. C. (2012). Um estudo prático para contagem volumétrica automática de veículos usando Visão Computacional. Master's thesis, Universidade Federal de Goiás.
- [Filho & Neto, 1999] Filho, O. & Neto, H. (1999). *Processamento digital de imagens*. Série Acadêmica. BRASPORT.
- [Gonzalez & Woods, 1992] Gonzalez, R. C. & Woods, R. E. (1992). *Processamento Digital de Imagens*. Addison-wesley publishing company edition.
- [Graciano, 2007] Graciano, A. B. V. (2007). Rastreamento de objetos baseado em reconhecimento estrutural de padrões. Master's thesis, Instituto de Matemática e Estatística.
- [Haykin, 1999] Haykin, S. (1999). *Redes Neurais - Princípios e prática*. 2 edition.
- [He et al., 2011] He, L., Wang, H., & Zhang, H. (2011). Object detection by parts using appearance, structural and shape features. In *International Conference on Mechatronics and Automation (ICMA)* (pp. 489–494).
- [Isard & Blake, 1998] Isard, M. & Blake, A. (1998). Icondensation: Unifying low-level and high-level tracking in a stochastic framework. In H. Burkhardt & B. Neumann (Eds.), *Computer Vision — ECCV'98*, volume 1406 of *Lecture Notes in Computer Science* (pp. 893–908). Springer Berlin Heidelberg.
- [Jung, 2012] Jung, K. (2012). Object recognition on mobile devices. In *Consumer Electronics - Berlin (ICCE-Berlin), 2012 IEEE International Conference on* (pp. 258–262).
- [Kalman, 1960] Kalman, R. E. (1960). A New Approach to Linear Filtering and Prediction Problems. *ASME Journal of Basic Engineering*.
- [Lameira et al., 2012] Lameira, A., Jesus, R., & Correia, N. (2012). Local object detection and recognition in mobile devices. In *Systems, Signals and Image Processing (IWSSIP), 2012 19th International Conference on* (pp. 193–196).

- [Lane et al., 2010] Lane, N., Miluzzo, E., Lu, H., Peebles, D., Choudhury, T., & Campbell, A. (2010). A survey of mobile phone sensing. *Communications Magazine, IEEE*, 48(9), 140–150.
- [Lopes, 2009] Lopes, D. C. (2009). *Implementação e Avaliação de Máquinas de Comitê em um Ambiente com Múltiplos Processadores Embarcados em um Único Chip*. PhD thesis, Universidade Federal do Rio Grande do Norte.
- [Lu et al., 2012] Lu, X. et al. (2012). Object contour tracking using multi-feature fusion based particle filter. In *Industrial Electronics and Applications (ICIEA)* (pp. 237–242).
- [Marcal et al., 2005] Marcal et al. (2005). Aprendizagem utilizando Dispositivos Móveis com Sistemas de Realidade Virtual. *Revista novas tecnologias na educação*, 3.
- [Mendel & McLaren, 1970] Mendel, J. M. & McLaren, R. W. (1970). *Reinforcement Learning control and pattern recognition systems*, volume 66.
- [Nummiaro et al., 2003] Nummiaro, K., Koller-Meier, E., & Gool, L. V. (2003). An adaptive color-based particle filter. *Image and Vision Computing*, 21(1), 99–110.
- [Patel & Thakore, 2013] Patel, H. A. & Thakore, D. G. (2013). Moving Object Tracking Using Kalman Filter. *International Journal of Computer Science and Mobile Computing (IJCSMC)*, 29(4), 326 – 332.
- [Simon et al., 2010] Simon, D. et al. (2010). Kalman filtering with state constraints: a survey of linear and nonlinear algorithms. *Control Theory & Applications, IET*, 4(8), 1303–1318.
- [Sousa & Marengoni, 2012] Sousa, K. & Marengoni, M. (2012). Uso de Visão Computacional em Dispositivos Móveis para o Reconhecimento de Faixa de Pedestres.
- [Suzuki & Abe, 1985] Suzuki, S. & Abe, K. (1985). Topological structural analysis of digitized binary images by border following. *Computer Vision, Graphics, and Image Processing*, 30(1), 32–46.

- [Wahi et al., 2008] Wahi, A. et al. (2008). A hybrid feature extraction method-based object recognition by neural network. In *Computing, Communication and Networking, 2008 (ICCCn)* (pp. 1–5).
- [Welch & Bishop, 2007] Welch, G. & Bishop, G. (2007). *An Introduction to the Kalman Filter*. Technical report, University of North Carolina at Chapel Hill.
- [Yang et al., 2011] Yang, J., 0002, Y. W., Sowmya, A., Li, Z., Zhang, B., & Xu, J. (2011). Feature fusion for vehicle detection and tracking with low-angle cameras. In *WACV* (pp. 382–388).: IEEE Computer Society.
- [Yilmaz et al., 2006] Yilmaz, A., Javed, O., & Shah, M. (2006). Object Tracking: A Survey. *ACM Comput. Surv.*, 38(4).