



Dissertação de Mestrado

**Um sistema oportunista para detecção de vagas de
estacionamento utilizando placas inteligentes
acopladas em câmeras de segurança**

David Henrique de Souza Lima
dhs.lima@gmail.com

Maceió, junho de 2014

David Henrique de Souza Lima

Um sistema oportunista para detecção de vagas de estacionamento utilizando placas inteligentes acopladas em câmeras de segurança

Dissertação apresentada como requisito parcial para obtenção do grau de Mestre pelo Programa de Pós-Graduação em Informática do Instituto de Computação da Universidade Federal de Alagoas.

Orientador: Prof. Dr. André Luiz Lins de Aquino

Orientadora: Profa. Dra. Eliana Silva de Almeida

Catálogo na fonte
Universidade Federal de Alagoas
Biblioteca Central
Divisão de Tratamento Técnico
Bibliotecário: Valter dos Santos Andrade

L732s Lima, David Henrique de Souza.
Um sistema para detecção de vagas de estacionamento utilizando placas
Acopladas em câmeras de segurança / David Henrique de Souza Lima. – 2014.
63 f. : il.

Orientador: André Luiz Lins de Aquino.

Orientadora: Eliana Silva de Almeida.

Dissertação (Mestrado em Informática) – Universidade Federal de Alagoas.
Instituto de Computação. Programa de Pós-Graduação em Informática.
Maceió, 2014.

Bibliografia: f. 62-63.

1. Sistemas embarcados (Computadores). 2. Processamento de imagens.
3. Placas inteligentes. 4. Câmeras de vídeos. 5. Cidades inteligentes.
I. Título.

CDU: 004.891



Membros da Comissão Julgadora da Dissertação de Mestrado de David Henrique de Souza Lima, intitulada: “Um Sistema Oportunista para Detecção de Vagas de Estacionamento Utilizando Placas Inteligentes Acopladas em Câmeras de Segurança”, apresentada ao Programa de Pós-Graduação em Informática da Universidade Federal de Alagoas em 10 de junho de 2014, às 10h, no Miniauditório do Instituto de Computação da UFAL.

COMISSÃO JULGADORA

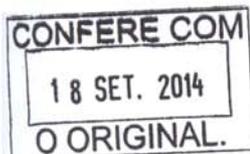
Prof. Dr. André Luiz Lins de Aquino
UFAL – Instituto de Computação
Orientador

Profa. Dra. Eliana Silva de Almeida
UFAL – Instituto de Computação
Orientadora

Prof. Dr. Leandro Dias da Silva
Instituto de Computação – UFAL
Examinador

Prof. Dr. Heitor Soares Ramos Filho
Instituto de Computação – UFAL
Examinador

Prof. Dr. Antônio Alfredo Ferreira Loureiro
Universidade Federal de Minas Gerais - UFMG
Examinador



Floripes Teixeira Santos
Secretaria - IC/UFAL
Mat. SIAPE 1673751

AGRADECIMENTOS

Agradeço a minha família, em especial, a minha mãe Amélia por toda educação, paciência e amor que me foi proporcionado. Gostaria de dedicar este trabalho a minha avó Denair de Souza Lima e meu tio Eduardo de Souza Lima que infelizmente não estão aqui para presenciar este momento, sempre amarei vocês.

Agradecimento especial para a minha noiva Hyllane Salgueiro, agradeço a você simplesmente por tudo! A paciência, compreensão, apoio, incentivo, ajuda, presença, críticas, dedicação, amor e carinho... Sem você nunca teria conseguido chegar até onde cheguei, obrigado por existir em minha vida! Amo você!

A minha eterna orientadora Eliana Almeida, por me aguentar desde a graduação até o mestrado. Obrigado Profa por toda paciência, incentivo, conselhos e por sempre acreditar em meu potencial. Obrigado por me dar a honra de ser seu orientando, devo muito de minha formação a você.

Ao meu orientador André L. L. Aquino, por todo apoio e incentivo dado durante este trabalho. Aprendi muito com você durante esses dois anos. Você se tornou além de meu orientador, um grande amigo.

Aos amigos do LaCCAN, por todos os momentos vividos durante todo esse tempo, agradecimento especial para Tamer, Evellyn, Rian, Victor, Anderson, Matheus e Leonildo, que estiveram comigo durante toda esta jornada. Aos amigos do Panda F. C. por todas as sociais, clubes da caixa e resenhas nesses anos de amizade. Aos amigos do IFAL por todo apoio. A Valve por ter criado o CS.

A mim por ter conseguido chegar até o fim.

Agradeço a todos que contribuíram de forma direta ou indireta neste trabalho.

David Henrique de Souza Lima

“Fear is just a feeling. You feel hot. You feel hungry. You feel angry. You feel afraid. Fear can never kill you.”

Chiun

RESUMO

Este trabalho apresenta um sistema para detecção de vagas de estacionamento utilizando câmeras de forma oportunista. A ideia central é a utilização de imagens provenientes de câmeras já instaladas para verificar a existência de vagas de estacionamento. É importante ressaltar que questões legais, de segurança e privacidade no uso de câmeras para detecção de vagas de estacionamento estão fora do escopo desta dissertação. Neste trabalho é abordado o problema de como utilizar câmeras de forma oportunista para detecção de vagas de estacionamento. A ideia para a solução é que por intermédio de sistemas embarcados, adaptado-os às câmeras já instaladas na cidade, podemos conceber satisfatoriamente a detecção de vagas. O objetivo principal deste trabalho é apresentar a viabilidade da utilização de sistemas embarcados para detecção de vagas de estacionamento. Para tal são propostas três técnicas de processamento de imagens para resolução do problema e como principal contribuição temos a apresentação de uma arquitetura distribuída para utilização de câmeras de forma oportunista para obtenção das imagens. As principais contribuições deste trabalho são a avaliação de três técnicas de processamento de imagens para detecção de veículos; proposta de uma arquitetura distribuída e oportunista utilizando câmeras para detecção de vagas de estacionamento; uma avaliação comparativa entre a arquitetura proposta e outras duas arquiteturas existentes; e um algoritmo de roteamento para encaminhamento das requisições na arquitetura *Embutida*. Inicialmente foram avaliadas três técnicas propostas de processamento de imagens para detecção de vagas de estacionamento, sendo a melhor delas a *Dilatação após Detecção de Borda* com uma eficácia de aproximadamente 100%. Essa avaliação serviu para escolhermos a técnica usada no restante do trabalho. Após a escolha da técnica, realizamos um experimento utilizando a arquitetura proposta e outras duas que serviram para comparação. O resultado do experimento demonstrou a viabilidade da utilização da arquitetura *Embutida*. Como realizar experimentos de grande porte é uma tarefa complexa, foi desenvolvida uma simulação utilizando o simulador de redes Sinalgo para analisar quatro variáveis no sistema (raio de comunicação, quantidade de câmeras auxiliares, quantidade de requisições simultâneas e percentual de falhas nas câmeras). Com o resultado da simulação pode-se concluir que os fatores com maior influência na variação dos tempos médios das requisições são o tamanho do raio de comunicação e a quantidade de câmeras auxiliares.

Palavras-chave: Cidades inteligentes, Placas inteligentes, Câmeras oportunistas, Sistemas embarcados, Processamento de imagens

ABSTRACT

This work aims to present a system for parking spaces detection using opportunistic cameras. The central idea is to use images from cameras that are already installed to identify possible parking spaces. It is important to highlight that legal, privacy and security issues in using cameras for parking detection is out of scope. This work addressed the problem of how to use cameras opportunistically to detect free parking spaces. The idea for the solution is that by means of embedded systems, adapted them to the cameras already installed in the city, we can successfully design the detection of vacancies. The main objective of this work is to present the feasibility of using embedded systems to detect parking spaces, for such, it is proposed three image processing techniques to solve the problem and as main contribution it is presented a distributed architecture for use of cameras opportunistically to obtain the images. The major contributions of this work were the development of three image processing techniques for vehicle detection; a distributed and opportunistic architecture proposal using cameras to detect parking spaces; comparative evaluation of the proposed architecture and two other existing architectures; a routing algorithm for routing requests in the *Embedded* architecture. Initially, it was evaluated three proposed image processing techniques to detect parking spaces, the best technique was the *Dilation after Border Detection* with a success rate close to 100%, this evaluation was performed to choose the technique used in the rest of this work. After choose the best technique, it was performed an experiment using the proposed architecture and two others that are used for comparison, the result of the experiment demonstrated the feasibility of use the *Embedded* architecture. To conduct large experiments is a complex task, because of this a simulation was developed using the network simulator Sinalgo to analyze four variables in the system (communication radius, number of auxiliary cameras, amount of simultaneous requests and percentage of failed cameras). The simulation result shows that the factors with the greatest influence on the variation of the average times of the requests are the size of the communication radius and the amount of auxiliary cameras.

Keywords: Smart cities, Intelligent boards, Opportunistic cameras, Embedded systems, Image processing

LISTA DE FIGURAS

Figura 3.1	Telas da aplicação Cliente	19
Figura 3.2	Dispositivos utilizados no sistema	20
Figura 3.3	Funcionamento da abordagem <i>Embutida</i>	21
Figura 3.4	Fluxograma de funcionamento do sistema	22
Figura 3.5	Exemplo de imagem em tons de cinza com a área demarcada	24
Figura 3.6	Imagens utilizadas como exemplo nas técnicas de processamento de imagens	25
Figura 3.7	Imagens convertidas para tons de cinza	25
Figura 3.8	Resultado da aplicação da operação de subtração entre as imagens	27
Figura 3.9	Resultado da aplicação da operação de binarização	28
Figura 3.10	Resultado da busca por vaga de estacionamento	28
Figura 3.11	Resultado da aplicação da operação de equalização de histogramas	29
Figura 3.12	Resultado da aplicação da operação de detecção de bordas	31
Figura 3.13	Resultado da aplicação da operação de binarização	32
Figura 3.14	Resultado da aplicação da operação de dilatação	32
Figura 4.1	Exemplo de uma linha temporal	36
Figura 4.2	Espaço ROC para as três técnicas apresentadas	40
Figura 4.3	Avaliação dos tempos de execução da pior e melhor câmera	42
Figura 4.4	Avaliação do tempo de execução para todas as câmeras	43
Figura 4.5	Gráfico de densidade para cada arquitetura avaliada	44
Figura 4.6	Análise detalhada de cada arquitetura	45
Figura 4.7	Disposição dos elementos no experimento realizado	48
Figura 4.8	Simulação com todas arquiteturas	52
Figura 4.9	Simulação com quantidade de nós auxiliares = 200	54
Figura 4.10	Simulação com percentual de falhas = 5	55
Figura 4.11	Simulação com raio de comunicação = 600 m	56
Figura 4.12	Simulação com quantidade de requisições = 500	57

LISTA DE TABELAS

Tabela 4.1	Uma ilustração geral de uma matriz de confusão	37
Tabela 4.2	Matriz de confusão para cada técnica	38
Tabela 4.3	Valores dos parâmetros utilizados na simulação	50

SUMÁRIO

1	Introdução	9
2	Trabalhos relacionados	13
3	Sistema oportunista para detecção de vagas de estacionamento	18
3.1	Aplicação para detecção de vagas de estacionamento	18
3.2	Arquitetura <i>Embutida</i>	20
3.3	Técnicas de processamento de imagens	23
3.3.1	Subtração	25
3.3.2	Equalização com Subtração	29
3.3.3	Dilatação após Detecção de Bordas	30
4	Avaliações e resultados obtidos	34
4.1	Especificidade dos experimentos	34
4.2	Avaliação das técnicas de processamento de imagens	36
4.3	Avaliação de desempenho via experimentação	40
4.4	Avaliação de desempenho via simulação	46
5	Considerações finais	58
	REFERÊNCIAS	60

1 Introdução

Atualmente, as cidades vêm enfrentando diversos desafios sócio-econômicos devido ao crescimento e migração urbana. O crescimento urbano, que em sua maioria, é realizado sem algum tipo de planejamento, insere um significativo *stress* nas infraestruturas das cidades demandando novos esforços no suprimento de água e energia, transporte, saúde, educação e segurança (Naphade et al., 2011). Nesse cenário, a tecnologia da informação e comunicação tem um papel fundamental para melhorar a qualidade de vida dos cidadãos que habitam em grandes cidades. Muitas aplicações interessantes podem surgir em um futuro próximo que irão criar cidades inteligentes, uma nova tendência que envolve inovação no planejamento, gerenciamento e operação.

Sistemas de Transporte Inteligentes (Ramos et al., 2012) têm favorecido o desenvolvimento de uma grande variedade de novas tecnologia para melhorar a qualidade do serviço de transporte e o trânsito, de aumentar o conforto dos usuários, a segurança e o entretenimento durante um trajeto. Uma problemática recorrente é a busca por vagas de estacionamento livres em cidades inteligentes. Esta tarefa é conhecida por consumir uma significativa quantidade de tempo por parte dos motoristas e aumenta a quantidade de tráfego nas cidades. Carros presos no tráfego geram um consumo desnecessário de combustível fóssil, o que causa um impacto econômico e resulta em um impacto ambiental negativo causado pela poluição do ar e pela emissão significativa de gases do efeito estufa (Waterson et al., 2001). Esse problema geralmente incide em áreas urbanas onde a quantidade de veículos é grande e existe um *déficit* de vagas de estacionamento livres (tanto vagas em estacionamento fechados quanto na rua). Para a comodidade dos motoristas, estacionamentos podem se responsabilizar por notificar a disponibilidade de vagas quando uma nova aparecer. No entanto, se a busca é realizada manualmente, os estacionamentos precisariam de uma quantidade significativa de recursos humanos para funcionarem de forma correta. Atualmente, existem diversos sistemas sem a necessidade de supervisão humana para detecção de vagas de estacionamento (Worldsensing SL, 2014; ParkHelp, 2014; Streetline Inc., 2014; Urbiotica, 2014; VehicleSense, 2014; TKH Group company, 2014).

Apesar disto, é importante salientar que, esses sistemas não foram desenvolvidos para detectar vagas de estacionamento nas ruas, que é considerado um ambiente na qual o usuário não tem controle. Por exemplo, em alguns lugares existe uma grande variação na luminosidade e algumas câmeras podem possuir uma baixa resolução para captar imagens. Por outro lado, em

estacionamentos internos, temos um ambiente controlado, onde pode-se escolher por exemplo, as câmeras que serão utilizadas e tais câmeras serão para uso exclusivo na detecção de vagas. Assim, um sistema oportunista, i.e., sistema que se beneficia de recursos de outros sistemas, pode ser utilizado para realizar esta tarefa. Neste trabalho é apresentado um sistema que utiliza imagens provenientes de câmeras de seguranças que já estão instaladas, logo, não é necessário instalar nenhuma infraestrutura adicional. É importante ressaltar que questões legais, de segurança e privacidade no uso de câmeras de segurança para detecção de vagas de estacionamento estão fora do escopo deste trabalho.

O seguinte problema é abordado neste trabalho:

Qual a viabilidade da utilização de câmeras de maneira oportunista para detecção de vagas de estacionamento em ambientes externos?

As hipóteses consideradas são (i) que por intermédio de sistemas embarcados, adaptado-os às câmeras já instaladas na cidade, podemos conceber satisfatoriamente a detecção de vagas; (ii) a utilização de placas de processamento acopladas às câmeras possuem resultados satisfatórios nos tempos de processamento.

O objetivo principal deste trabalho é analisar a viabilidade da utilização de sistemas embarcados para detecção de vagas de estacionamento. A ideia é utilizar câmeras de forma oportunista para captar e processar imagens. Para realizar o processamento foram acopladas placas de processamento às câmeras, tais placas possuem também a capacidade de comunicação, ou seja, caso a câmera não possua acesso à rede, a placa terá também esta atribuição. As câmeras utilizadas de forma oportunista podem ser câmeras presentes em edifícios ou casas ou câmeras de segurança. Para o funcionamento do sistema é preciso que seja disponibilizado o acesso público às imagens destas câmeras. A vantagem da utilização de câmeras que já estejam previamente instaladas nos ambientes é a diminuição nos custos tanto de implantação quanto de manutenção, visto que, serão utilizadas câmeras que não foram implantadas exclusivamente para a aplicação proposta. É importante frisar que o oportunismo do sistema é relacionado a utilização das câmeras para aquisição das imagens e não para comunicação, como ocorre em redes oportunistas (Pelusi et al., 2006).

O desenvolvimento de uma solução para o problema de busca de vagas de estacionamento se justifica por diversos trabalhos que tratam de problemas gerados pelo processo de localização de vagas de estacionamento pelos motoristas (Polycarpou et al., 2013; Shoup, 2006). Tais problemas são relativos ao aumento da produção de gases tóxicos, aumento no consumo de combustíveis fósseis, além do tempo desperdiçado durante o processo de busca de uma vaga de estacionamento livre. Evenepoel et al. (2014) apresenta um estudo sobre o impacto financeiro da implantação de sistemas para resolução deste problema.

A metodologia adotada na realização deste trabalho é a seguinte, inicialmente foram avaliadas três técnicas de processamento de imagens: *Subtração*, *Equalização com Subtração* e *Dilatação após Detecção de Bordas*. Essas técnicas têm como objetivo a resolução do problema

da detecção de vagas de estacionamento. As técnicas utilizadas são então avaliadas utilizando imagens capturadas de um estacionamento. Essa avaliação teve como objetivo eleger quais das três técnicas obteve uma melhor eficácia, em outras palavras, qual das três técnicas obteve uma maior taxa de acerto. Os resultados revelam que a técnica *Dilatação após Detecção de Borda* apresenta um taxa de detecção com sucesso de aproximadamente 100%.

Após eleger a melhor técnica para detecção de vagas de estacionamento, avaliamos o desempenho do sistema utilizando apenas a técnica *Dilatação após Detecção de Borda*. O desempenho do sistema é dado basicamente pela soma do tempo de processamento e o tempo de comunicação. Esta avaliação de desempenho considera três diferentes arquiteturas:

- I ***Embutida***: é uma arquitetura *ad-hoc* onde apenas as câmeras são responsáveis por todo o processamento de imagens e pela comunicação com os clientes.
- II ***Centralizada***: utiliza uma arquitetura cliente-servidor onde o servidor central é responsável por todo o processamento das imagens e pela comunicação com os clientes. Esta arquitetura serve como um *baseline* para comparação com as demais.
- III ***Híbrida***: é uma arquitetura onde as câmeras lidam com a comunicação com os clientes e um servidor central é responsável por todo o processamento das imagens.

A avaliação das arquiteturas é realizada por duas abordagens: experimentação e simulação. Para os experimentos foram utilizadas sete câmeras monitorando dois estacionamentos. Cada arquitetura foi implantada e executada sobre uma certa quantidade de imagens (foi aplicado o mesmo conjunto de imagens em nas três arquiteturas). Após isto foi quantificado o tempo médio de execução de uma requisição. Foram analisados o tempo total e os tempos fracionados em cada estágio (comunicação e processamento). Os resultados demonstraram que a arquitetura *Embutida* obteve melhores resultados quando comparada à arquitetura *Híbrida*. No entanto, a arquitetura *Centralizada* possui melhores tempos em comparação com as demais. A avaliação por intermédio da simulação foi realizada utilizando o simulador SIGNALGO (Distributed Computing Group, 2008). O objetivo da utilização de simulação foi avaliar o comportamento do sistema em cenários onde não foi possível verificar via experimentação. Foram desenvolvidos cenários na simulação combinando a variação de fatores como o raio de comunicação, quantidade de câmeras oportunistas utilizadas e quantidade de requisições simultâneas recebidas. Os resultados demonstraram que a utilização da arquitetura *Embutida* é viável e não causa grandes perdas comparada às arquiteturas *Centralizada* e *Híbrida*.

As principais contribuições deste trabalho são:

- Avaliação de três técnicas de processamento de imagens para detecção de veículos;
- Avaliação de uma arquitetura distribuída e oportunista utilizando câmeras para detecção de vagas de estacionamento;

- Avaliação comparativa entre arquiteturas existentes;
- Adaptação de um algoritmo de roteamento de redes de sensores sem fio para a utilização na arquitetura *Embutida*.

Tivemos dois artigos completos aceitos em anais de eventos nacionais. As publicações foram:

- Lima, D. H. S., Aquino, A. L. L., Almeida, E. S., Sistema para detecção oportunista de vagas de estacionamento via câmeras de segurança, in V Simpósio Brasileiro de Computação Ubíqua e Pervasiva, Maceió - AL, 2013.
- Oliveira, S., Teixeira, F. A., Macedo, D. F., Aquino, A. L. L., Lima, D. H. S., Silva, C., Silva, R., Sistema de coleta e disseminação de dados de trânsito, in XXXI Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos, Brasília - DF, 2013.

Este trabalho está dividido da seguinte maneira:

Capítulo 2: apresenta os trabalhos relacionados ao problema de detecção de vagas de estacionamento. São apresentadas soluções utilizando redes de sensores sem fio, sensores acoplados, *smartphones*, redes neurais e processamento de imagens;

Capítulo 3: apresenta os conceitos referentes ao funcionamento, desenvolvimento e partes integrantes do sistema proposto. Nesse capítulo são também detalhadas as técnicas de processamento de imagens desenvolvidas neste trabalho, além de apresentar a arquitetura proposta e as utilizadas para comparação e avaliação;

Capítulo 4: apresenta inicialmente a avaliação das técnicas de processamento de imagens utilizadas. Em seguida, são descritos e discutidos tanto o experimento quanto a simulação realizados com o objetivo de avaliar e validar a arquitetura proposta neste trabalho;

Capítulo 5: apresenta uma recapitulação do trabalho realizado, além das considerações finais e dos trabalhos futuros.

2 Trabalhos relacionados

Existem diferentes estratégias para detecção de vagas de estacionamentos. Neste capítulo serão apresentados alguns trabalhos relacionados que possuem como principal problema a detecção de vagas de estacionamento.

Os trabalhos de [Geng & Cassandras \(2012\)](#), [Zhang et al. \(2013\)](#) e [Idris et al. \(2009\)](#) apresentam soluções para o problema de detecção de vagas utilizando redes de sensores sem fios. [Geng & Cassandras \(2012\)](#) propõem uma solução que pode ser utilizada tanto em estacionamentos fechados quanto em estacionamentos nas ruas. Utiliza um sistema luminoso para reserva de vagas quando o sistema funciona em um estacionamento fechado, porém quando tratamos de um estacionamento na rua surge o problema de não poder garantir a vaga. O autor cita a ideia de utilizar obstáculos que entrariam/sairiam do chão e que seriam ativados/desativados por intermédio de troca de mensagens com o usuário que tivesse reservado a vaga. O trabalho citado difere do trabalho proposto em relação a arquitetura utilizada no sistema. No trabalho citado é utilizada uma arquitetura centralizada, com o processamento dos dados ocorrendo em um servidor central, enquanto que o trabalho proposto apresenta uma arquitetura distribuída, com o processamento e a comunicação sendo responsabilidade das câmeras utilizadas no sistema. O trabalho citado não apresentou avaliação da solução apresentada.

[Zhang et al. \(2013\)](#) utilizam sensores que captam o campo magnético, assumindo que os veículos são formados por materiais de metal, para detectar a existência ou não de uma vaga de estacionamento. Além dos sensores, o sistema é formado também por uma estação base, roteadores e um servidor remoto. Cada nó sensor é responsável por monitorar uma vaga de estacionamento específica, este sensor será responsável por informar o estado da vaga (livre ou ocupada). A comunicação entre os componentes do sistema é realizada utilizando o padrão de comunicação ZigBee ([Baronti et al., 2007](#)). Para avaliação da proposta, os autores realizaram um experimento com duração de um ano utilizando 82 nós sensores. Os resultados demonstraram que esta solução alcançou em torno de 98% de acurácia na verificação da existência ou não de vagas de estacionamento. Além de demonstrar a eficácia da solução, o experimento demonstrou que a utilização dos nós sensores é viável, pois seria necessário trocar a bateria dos mesmos a cada cinco anos. A maior diferença entre o trabalho citado e o trabalho proposto é o fato do trabalho citado necessitar realizar um mapeamento de um para um entre os nós sensores e as vagas de estacionamento, isto é, a cada vaga de estacionamento que se deseje monitorar, será

necessário um nó sensor para ser responsável pela mesma, enquanto que o trabalho proposto, ao utilizar técnicas de processamento de imagens consegue-se monitorar quantas vagas estiverem em sua visada.

Idris et al. (2009) apresentam uma solução utilizando uma rede de sensores ultra-sônicos. Este trabalho apresenta um sistema que utiliza uma arquitetura centralizada para gerenciar as requisições dos usuários e o estado das vagas do estacionamento. Para realizar a verificação da existência de vagas de estacionamento são utilizados sensores ultra-sônicos. Os autores utilizam a ideia de analisar a quantidade de energia refletida para calcular a existência da vaga de estacionamento. Como padrão de comunicação é utilizado o ZigBee. O trabalho, além de prover o serviço de detecção de vaga de estacionamento, também apresenta um módulo que tem como objetivo guiar o usuário até uma vaga de estacionamento livre, este caminho é escolhido utilizando o algoritmo A^* (Hart et al., 1968). Por utilizar sensores ultra-sônicos, este sistema é bastante afetado com mudanças na temperatura, enquanto que o trabalho proposto, por utilizar processamento de imagens para detecção de vagas, não possui este problema. Além disto, o foco do trabalho de Idris et al. são estacionamentos fechados, enquanto que o trabalho proposto tem como objetivo a detecção de vagas em qualquer rua/avenida. Outra diferença entre os trabalhos está no fato do trabalho citado utilizar uma arquitetura centralizada, enquanto a proposta deste trabalho é a utilização de uma arquitetura distribuída para resolver o problema da detecção de vagas de estacionamento.

As soluções propostas por Mathur et al. (2010), Alhammad et al. (2012), Caliskan et al. (2006) e Zhao et al. (2012) apresentam alternativas para resolver o problema de detecção de vagas de estacionamento utilizando VANETs (Hartenstein & Laberteaux, 2008) com sensores acoplados. Estas abordagens eliminam a necessidade de instalação de sensores nos estacionamentos. Mathur et al. (2010) utilizam sensores ultra-sônicos acoplados as laterais dos veículos para realizar a detecção de vagas. O trabalho tem como área de atuação estacionamentos públicos localizados em ruas/avenidas. Esta solução utiliza uma arquitetura centralizada, onde os dados captados pelos veículos são enviados para um servidor central que tem como responsabilidade mesclar as informações recebidas e criar um mapa com as vagas de estacionamento que estão disponíveis. Os autores apresentam duas formas distintas para conexão entre os veículos e o servidor central, a primeira consiste na utilização da rede celular, enquanto que a segunda consiste na utilização oportunista de uma conexão wifi. Um dos problemas encontrados nesta abordagem, é o fato do sensor não conseguir distinguir um carro de um objeto qualquer, podendo causar um aumento na quantidade de resultados falsos negativos (inferir que a vaga está ocupada, porém ela está livre). Para avaliação da proposta, os autores realizaram um experimento com duração de dois meses. O experimento ocorreu na cidade de Nova Jersey e foram utilizados três veículos. Ao fim, a proposta obteve aproximadamente 80% de acerto. Os autores sugerem que sua solução possa ser utilizada acoplando os sensores em veículos públicos, como carros de polícia, ônibus, táxis etc. Isto ocorre, pois assume-se que estes veículos, em sua maioria, ficam circulando pela cidade. Desta maneira não existiria a necessidade de acoplar

os sensores em veículos particulares. A principal diferença entre o trabalho citado e o trabalho proposto está na arquitetura utilizada em cada um, enquanto o trabalho citado utiliza uma arquitetura centralizada, com um servidor em cada estacionamento, a contribuição do trabalho proposto é a utilização de uma arquitetura distribuída para resolver o problema de detecção de vagas de estacionamento.

Alhammad et al. (2012) apresentam uma solução utilizando sensores implantados nos para-choques dos veículos. A proposta utiliza o conceito de estações de informações, que exercem o papel de infraestrutura centralizada para armazenar o estado (livre ou ocupado) das vagas de estacionamento. Esta proposta é aplicada tanto em estacionamentos fechados quanto em ruas/avenidas. Para comunicação utiliza o padrão DSRC (*Dedicated Short Range Communication*), esta comunicação pode ser tanto com uma infraestrutura (V2I) quanto com outro veículo (V2V). Um problema que a utilização do DSRC traz a tona é a questão do alcance das mensagens na rede. Os autores utilizam também bancos de dados locais, onde cada participante da rede (seja um veículo ou uma infraestrutura) possui o seu próprio. Estes bancos locais são replicados para o banco de dados da estação de informação central. Para verificar a existência de uma vaga, os autores utilizam um padrão para o tamanho da vaga de estacionamento acrescido de um espaço extra. Os trabalhos se diferem no simples fato da proposta do artigo citado utilizar um servidor central para o armazenamento do banco de dados de todo sistema, enquanto que a contribuição do trabalho proposto é uma solução distribuída para o problema de detecção de vagas de estacionamento.

Caliskan et al. (2006) apresentam uma solução onde ocorre a comunicação entre veículos e parquímetros para informar o estado atual das vagas de estacionamento disponíveis. Esta comunicação é realizada utilizando o padrão IEEE 802.11. Os parquímetros, por intermédio de mensagens de *broadcasts* periódicos, informam aos veículos sobre o estado das vagas de estacionamento e suas localizações (coordenadas GPS). Os veículos realizam o serviço de agregação dos dados recebidos e os encaminham para os outros veículos. Para validação da proposta, os autores realizaram simulações com os simuladores NS-2 e VISSIM. O trabalho citado se difere do trabalho proposto pela necessidade de ter um mapeamento de um para um, sendo um parquímetro responsável por uma vaga, enquanto que o trabalho proposto, ao utilizar técnicas de processamento de imagens consegue monitorar quantas vagas estiverem em sua visada.

Zhao et al. (2012) apresentam uma solução utilizando veículos previamente estacionados para determinar a existência de vagas de estacionamento. Este trabalho oferece dois serviços, um de disseminação de informação de vagas de estacionamento e outro de navegação em estacionamentos fechados. A abordagem inicia com os carros nos estacionamentos formando *clusters* entre si, em cada *cluster* formado um dos participantes deverá ser eleito líder (*cluster head*), para receber os dados dos demais participantes do *cluster*. Ao receber os dados, o líder deverá armazenar os dados recebidos em um banco de dados local. Um desafio para o funcionamento desta proposta é a necessidade dos carros, além de possuírem um módulo de comunicação, devem dispor de um dispositivo GPS e disponibilizá-lo para utilização pelo sistema.

Para determinar a existência de vagas de estacionamento, cada veículo deverá gerar um mapa do seu redor com informações a respeito das vagas (se estão livres ou ocupadas e suas localizações) e enviar tais informações para o *cluster head*. O líder terá a função de construir o mapa do estacionamento a partir de fragmentos enviados pelos carro que compõe o grupo. Diferentemente do trabalho proposto, este trabalho utiliza um servidor central para armazenamento das informações geradas na rede, enquanto o objetivo da proposta é a utilização de uma arquitetura distribuída.

Os trabalhos de Xu et al. (2013) e Nawaz et al. (2013) abordam a solução para detecção de vagas de estacionamento utilizando *smartphones* como dispositivos principais. Xu et. al utilizam os dados provenientes do GPS e do acelerômetro ou do *bluetooth* do dispositivo para inferir a localização da vaga, enquanto que Nawaz et. al utilizam dados provenientes da mudança dos *beacons* da conexão WiFi. Um problema comum aos dois trabalhos é o fato de ambos detectaram que o motorista do veículo deixou o veículo parado, com isto eles assumem que o veículo foi estacionado, porém não tem como afirmar com certeza, por exemplo, que o veículo foi estacionado em um local permitido. Com a volta do motorista ao veículo, e percebendo a movimentação do carro (ao se mover a velocidade do carro é claramente maior do que a de uma pessoa), o sistema assume que onde o veículo estava parado é uma vaga de estacionamento e que no momento ela está livre. Outro problema é a necessidade de todos os usuários possuírem o aplicativo instalado em seus *smartphones*, caso isto não ocorra não será possível, por exemplo, mensurar a quantidade de vagas no local, ou até mesmo, se determinado espaço é uma vaga de estacionamento.

Jermurawong et al. (2012) utilizam uma abordagem com redes neurais para determinar se a vaga de estacionamento está ocupada ou vaga, baseado na extração de características visuais dos estacionamentos. As características extraídas são relacionadas com a luminosidade, bordas, pixels, cores e adicionalmente, os autores incluíram o horário. Para alcançar a solução, os autores tiveram que lidar com três diferentes problemas: variação da intensidade da luz, limitação da luminosidade durante o nascer e o pôr do sol e os diferentes padrões de luminosidade durante a noite. O trabalho apresentou uma taxa de acerto de cerca de 98%.

Os trabalhos de Yusnita et al. (2012), Lixia & Dalin (2012) e Wu et al. (2007) apresentam soluções para o problema de detecção de vagas de estacionamento utilizando processamento de imagens, tal abordagem é também utilizada no trabalho proposto. apresenta uma solução para detecção de vagas de estacionamento utilizando processamento de imagens. Neste contexto, Yusnita et al. (2012) desenvolveu um sistema que possui seis etapas: inicialização do sistema, aquisição da imagem, segmentação da imagem, transformação da imagem em tons de cinza, *thresholding* e detecção de bordas nos objetos. O principal objetivo é realizar detecção na imagem para localizar a quantidade de vagas de estacionamento disponíveis. Para avaliar as técnicas propostas, os autores utilizaram carro de brinquedo em miniatura e realizaram todas as avaliações em um ambiente interno. Em nossa proposta, nós também aplicamos técnicas de processamento de imagem, mas nossos experimentos são realizados em um ambiente real.

Além disto, diferentemente de Yusnita, nossa proposta leva em consideração detecção de vagas de estacionamento nas ruas.

Lixia & Dalin (2012) buscam por vagas de estacionamento tratando-as como se fossem células. Para verificar a disponibilidade de determinada vaga, os autores buscam por características comuns presentes em veículos (vidros, grades, portas etc), com estas características, eles buscam nas células por combinações que podem ser detectadas após o processo de segmentação da imagem. Se alguma combinação for encontrada, a vaga é considerada ocupada, caso contrário a vaga é considerada livre. A principal desvantagem desta proposta é a taxa de falhas quando existir algum objeto encobrindo parte da vaga. Apesar disto, este método obteve uma taxa de acurácia de 97%.

Wu et al. (2007) apresentam uma solução para o problema de detecção de vagas de estacionamento utilizando a combinação de imagens provenientes de câmeras com técnicas de aprendizado de máquinas. Dado o vídeo capturado como entrada, os autores distinguem espaços vagos de espaços ocupados utilizando um classificador máquina de vetor de suporte (SVM) de 8 classes com saídas probabilísticas. Para validação da proposta, foi realizado um experimento capturando imagens de um estacionamento real. A proposta obteve uma taxa de acurácia de aproximadamente 85%.

3 Sistema oportunista para detecção de vagas de estacionamento

Neste capítulo serão abordados conceitos referentes ao funcionamento, desenvolvimento e partes integrantes do sistema desenvolvido. A seção 3.1 apresenta a aplicação cliente desenvolvida para os usuários poderem utilizar o sistema proposto. Na seção 3.2 são apresentadas a arquitetura *Embutida*, proposta por este trabalho e outras duas arquiteturas, chamadas de *Centralizada* e *Híbrida*, utilizadas como base de comparação para avaliação da proposta. Por fim, na seção 3.3 são apresentadas três técnicas de processamento de imagens utilizadas para detectar a existência de vagas de estacionamento. As técnicas são chamadas de *Subtração*, *Equalização com Subtração* e *Dilatação após Detecção de Bordas*, tais técnicas foram desenvolvidas utilizando princípios de processamento de imagens e morfologias matemáticas.

3.1 Aplicação para detecção de vagas de estacionamento

Para a utilização do sistema pelos usuários foi preciso o desenvolvimento de uma aplicação, onde por ela, o usuário pudesse realizar uma requisição para buscar vagas de estacionamento nos locais nos quais ele deseje. A aplicação foi desenvolvida para o sistema operacional ANDROID utilizando a *API* do GOOGLE MAPS para apresentar o mapa da cidade de Maceió com os possíveis locais de estacionamento. Na Figura 3.1, é possível ver o funcionamento da aplicação Cliente.

Na Figura 3.2(a) é apresentada a tela inicial da aplicação cliente. Ao iniciar a aplicação, pode-se observar o mapa da cidade de Maceió. Através deste mapa o usuário poderá escolher o local onde deseja verificar a existência de vagas de estacionamento. Para selecionar uma área escolhida, é preciso que o usuário pressione a tela no local desejado. Após isto, a área será demarcada com um ponto, como pode ser visualizado na Figura 3.2(b). Ao demarcar todas as áreas desejadas, o usuário deverá selecionar o botão "Buscar Vagas", localizado na parte inferior da aplicação. O objetivo deste botão é encaminhar a requisição do usuário para o responsável em analisar e processar as áreas selecionadas. O usuário deverá aguardar a resposta sobre o estado dos locais selecionados, caso existam vagas de estacionamento no local selecionado, a

Figura 3.1 – Telas da aplicação Cliente

Fonte: Autor, 2014

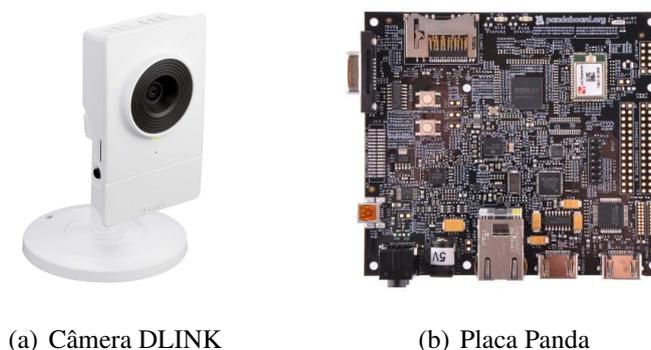
marcação realizada previamente pelo usuário ficará na cor verde (Figura 3.2(c)), caso contrário a marcação deverá ficar na cor vermelha (Figura 3.2(d)). Durante o desenvolvimento da aplicação cliente, existia a possibilidade de obter a localização do usuário através de coordenadas do GPS e com tal informação prover direções para vagas de estacionamentos que estivessem mais próximas a ele, porém esta abordagem foi desconsiderada, pois caso fosse utilizado a localização GPS do usuário, seria possível apenas realizar buscas por vagas de estacionamento que estivessem próximas a ele, enquanto que o objetivo da aplicação é que o usuário possa ser livre para escolher o local onde ele pretende realizar a busca das vagas de estacionamento. Esta característica é útil pois, por exemplo, caso o usuário esteja distante de seu destino, porém, o mesmo queira verificar o estado atual das vagas de estacionamento próximas a seu destino, ele poderá o fazer sem algum problema.

A ideia principal do trabalho é a utilização de câmeras oportunistas para captação das imagens que foram utilizadas para detecção de vagas de estacionamento. Porém, por necessidade de realização de experimentos, as câmeras utilizadas neste trabalho não foram utilizadas de maneira oportunista, ou seja, foi preciso realizar a instalação de algumas câmeras para serem utilizadas para a avaliação das propostas apresentadas, pois não existiam câmeras no local do experimento. Para o experimento realizado foram utilizados sete pontos de monitoramentos, cada um com uma câmera responsável, capturando imagens de duas áreas de estacionamento diferentes. As áreas de estacionamento monitoradas possuíam uma área em formato retangular, a primeira com 100 m^2 e a segunda com 500 m^2 . Cada câmera era responsável por monitorar um ponto diferente de cada área de estacionamento. Todas as câmeras utilizadas no experimento foram instaladas na mesma altura (em torno de 3 m) e com distância uniforme entre elas (em torno de 2 m). Como não foi possível a criação de um cenário para uso de câmeras oportunista, foi também desenvolvido uma simulação para verificar o comportamento do sistema em

cenários onde não foi possível a reprodução no experimento. Foram desenvolvidos cenários na simulação combinando a variação de fatores como o raio de comunicação, quantidade de câmeras oportunistas utilizadas e quantidade de requisições simultâneas recebidas.

As câmeras utilizadas no projeto foram as DCS-2130¹ da marca DLINK (ver Figura 3.3(a)), em alternativa poderiam ser utilizadas quaisquer câmeras *ip*. Estas câmeras são apropriadas para ambientes internos. É importante ressaltar que esta característica não é um fator de impedimento para a execução do experimento. Possuem a capacidade de captar imagens em HD, além de possuir conectividade via *wireless* ou via *ethernet*.

Figura 3.2 – Dispositivos utilizados no sistema



Fonte: Autor, 2014

As placas utilizadas, em conjunto com as câmeras, são as placas Panda². Esta placa possui um processador AMR *dual-core* de 1.2GHz além de 1GB de memória RAM DDR2. As placas Panda possuem um baixo custo de aquisição, com um preço em média de \$178. Na Figura 3.3(b) podemos visualizar a placa Panda.

3.2 Arquitetura *Embutida*

Para realizar a detecção oportunista de vagas de estacionamento utilizando câmeras de segurança, foram consideradas três diferentes arquiteturas para execução do sistema, sendo uma a proposta deste trabalho e as outras são utilizadas para comparação com a arquitetura proposta. A arquitetura proposta é chamada de *Embutida*. A abordagem *Embutida* é um arquitetura *ad-hoc*, onde toda comunicação é realizada entre os clientes e as câmeras de forma direta. Nessa arquitetura, as câmeras possuem placas de processamento acopladas para realizar o processamento. No caso onde a câmera não possui capacidade de comunicação, essa placa pode ser utilizada

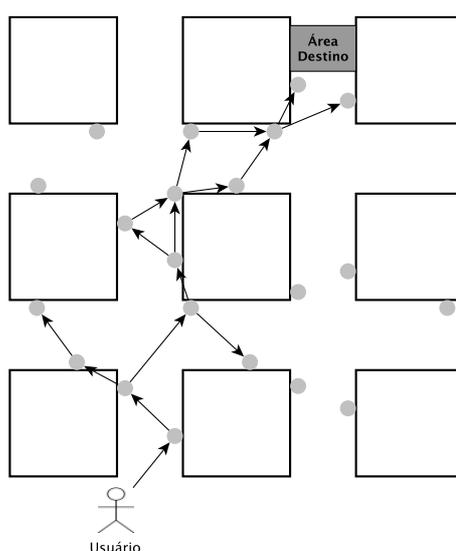
¹<http://www.dlink.com.br/site/index.php/produtos/detalhes/dcs-2130.html>

²<http://pandaboard.org/>

também para esse fim. Quando as câmeras recebem uma requisição, elas processam a imagem e enviam o resultado obtido para o cliente. É importante ressaltar que toda a comunicação cliente-câmera-cliente é efetuada de forma *ad-hoc*.

Na Figura 3.3 podemos ver o funcionamento da arquitetura *Embutida*. Na imagem estão representados o usuário, a área na qual se quer verificar a existência de vagas de estacionamento e as câmeras utilizadas no processo (representadas pelas circunferências). Quando o usuário realiza uma requisição, as câmeras que estiverem ao seu alcance irão receber a requisição e se responsabilizar de encaminhá-la até as câmeras na área onde o usuário queira realizar a busca por vagas de estacionamento. Quando a mensagem alcançar as câmeras que estejam na área destino, elas irão captar a imagem e processá-la, e, em seguida, responder com o resultado do processamento realizado. A abordagem *Embutida* possui algumas das vantagens trazidas pelo uso de sistemas distribuídos (Tanenbaum & Steen, 2006): escalabilidade e disponibilidade do sistema. A escalabilidade trata do aumento na quantidade de câmeras participantes do sistema, na arquitetura *Embutida*, i.e., caso mais câmeras participem, não é necessário realizar nenhuma alteração no funcionamento. O fator da disponibilidade corresponde ao fato de se uma câmera para de funcionar, não irá afetar o funcionamento do sistema como um todo, visto que existirão outras câmeras para substituir o papel da câmera que falhou, porém esta característica é dependente da cobertura do raio de comunicação. Como a ideia do sistema é utilizar câmeras de maneira oportunista, não é possível saber por quanto tempo uma câmera estará disponível para ser utilizada pelo sistema, possuir esta característica de disponibilidade é de suma importância para o funcionamento do sistema proposto.

Figura 3.3 – Funcionamento da abordagem *Embutida*



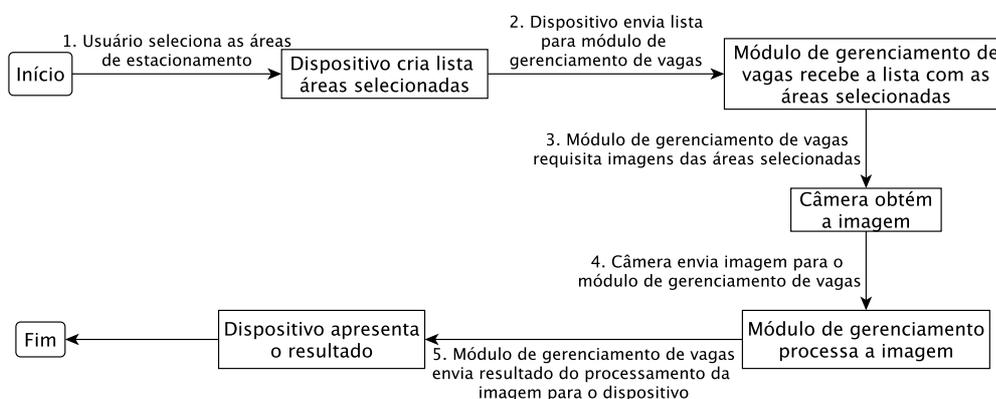
Fonte: Autor, 2014

Além da abordagem proposta, foram utilizadas outras duas abordagens para servirem como comparação da arquitetura *Embutida*. A abordagem *Centralizada* consiste na utilização de uma arquitetura cliente-servidor, na qual, para os testes realizados, foi utilizado um computador pessoal. Nessa proposta o servidor será responsável por receber as requisições enviadas pelo cliente. Quando o servidor receber as requisições do cliente, ele irá requisitar as imagens das câmeras para então realizar o processamento das imagens e enviar a resposta para o cliente. É importante ressaltar que toda a comunicação cliente-servidor-cliente é efetuada por intermédio da *Internet*.

A abordagem *Híbrida* consiste na utilização de um servidor para gerenciar as conexões com os cliente e para encaminhar as requisições para as câmeras. Além disto, esta abordagem utiliza placas de processamento acopladas às câmeras para executar os algoritmos de processamento de imagens. Nessa arquitetura, o servidor recebe a requisição do cliente, então, ele encaminha a requisição para a respectiva câmera que irá processar a imagem utilizando a placa de processamento acoplada a ela e então enviará a resposta para o servidor. É importante ressaltar que toda a comunicação cliente-servidor-câmera-servidor-cliente é efetuada por intermédio da *Internet*.

Os passos do funcionamento do sistema, de uma forma geral, são descritos a seguir e ilustrados na Figura 3.4.

Figura 3.4 – Fluxograma de funcionamento do sistema



Fonte: Autor, 2014

1. Usuário seleciona uma área de estacionamento pública (rua, avenida etc) onde ele/ela deseja verificar se existem vagas de estacionamento disponíveis. O dispositivo Android cria uma lista contendo possíveis estacionamentos na área selecionada.
2. O dispositivo *Android* envia a lista para o módulo de detecção de vagas de estacionamento que irá identificar as câmeras no estacionamento. Este módulo será o servidor na abordagem *Centralizada* ou *Híbrida*; ou as placas de processamento na abordagem *Embutida*.

3. O módulo detector de vagas de estacionamento requisita as imagens das câmeras
4. A câmera obtém a imagem e envia para o módulo detector de vagas de estacionamento (*Centralizada*, *Híbrida* ou *Embutida*). Após receber as imagens das câmeras, o módulo detector de vagas de estacionamento realiza o processamento nas imagens para detectar a existência de vagas de estacionamento livres.
5. Após verificar todos os estacionamentos selecionados, o módulo detector de vagas de estacionamento enviará o resultado para o usuário, que será capaz de visualizar o resultado no dispositivo utilizado.

É importante destacar que se existir uma vaga de estacionamento livre em algum lugar e três motoristas requisitarem por informações de disponibilidade de vagas naquele lugar, todos serão informados que existe uma vaga disponível, mas uma vez que eles cheguem ao local outro usuário (ou até um não usuário) poderá ter estacionado na vaga, desperdiçando tempo e esforço.

As arquiteturas *Centralizada* e *Híbrida* foram utilizadas como base para comparação com a arquitetura *Embutida*, que é a proposta deste trabalho.

3.3 Técnicas de processamento de imagens

Uma das etapas necessárias para o funcionamento do sistema é o processo de detecção de vagas de estacionamento. Para detecção das vagas de estacionamento torna-se necessário o desenvolvimento de técnicas de processamento de imagens. As técnicas utilizadas neste trabalho são resultado de sucessivas aplicações de morfologias matemáticas (Dougherty, 1992b) e de processamento de imagens. Nesta seção são detalhadas as seguintes técnicas:

- *Subtração*: esta técnica possui apenas duas etapas de processamento de imagens. A primeira etapa é a subtração da imagem recebida pela requisição do servidor e uma imagem anterior com o estacionamento vazio. No resultado desta operação é aplicado uma técnica de binarização. A técnica de binarização consiste em transformar uma imagem em tons de cinza para uma imagem em preto e branco.
- *Equalização com Subtração*: esta técnica é composta pela técnica *Subtração*, com o adicional de aplicar a técnica de equalização do histograma em cada imagem antes de executar o processo de subtração. A técnica de equalização de histogramas tem como objetivo balancear o contraste da imagem de acordo com seu histograma.
- *Dilatação após Detecção de Bordas*: esta técnica aplica um método de detecção de bordas na imagem para depois utilizar métodos de binarização e dilatação. A detecção de bordas é uma técnica que tem como objetivo encontrar pontos onde ocorram mudanças repentinas na brilho da imagem. O efeito básico do operador de Dilatação em imagens

binárias é o aumento gradual dos limites das regiões dos *pixels* do primeiro plano (e.g. *pixels* brancos geralmente). Assim, as áreas com *pixels* do primeiro plano aumentam de tamanho, enquanto que os buracos existentes nestas regiões tornam-se menores.

Todas as técnicas foram aplicadas em uma área pré-determinada do estacionamento (Figura 3.5), e.g., é necessário fazer um pré processamento nas imagens para demarcar todas as possíveis áreas onde o veículo poderia ser estacionado. A ideia desse processo é não ter que processar todas as imagens mas apenas os lugares que foram demarcados. Esta área específica deve ser marcada na inicialização do sistema por seu administrador. A aplicação das técnicas em um sistema real possui alguns desafios e limitações comuns a todas, como por exemplo, caso exista uma vaga que esteja encoberta por algum objeto, não será possível aferir a existência ou não da vaga de estacionamento. Esse problema poderá ser solucionado se houver outra câmera captando o mesmo local, porém em um ângulo onde a vaga não seja coberta por nenhum objeto. A etapa de pré processamento de imagens é comum em diferentes técnicas de processamento de imagens, o objetivo é adequar a imagem de entrada aos requisitos da técnica (Prince, 2012).

Figura 3.5 – Exemplo de imagem em tons de cinza com a área demarcada



Fonte: Autor, 2014

Uma imagem é uma representação espacial de um objeto, de uma cena bi ou tri dimensional, ou de outra imagem. Em visão computacional, o termo imagem geralmente significa imagem gravada, como por exemplo, um vídeo, uma imagem digital ou uma fotografia. O conceito de imagem pode ser abstraído como um função contínua I de duas variáveis definidas em uma região limitada e geralmente retangular de um plano. O valor da imagem localizada nas coordenadas espaciais (i, j) é denotado por $I(i, j)$ (Dougherty, 1994).

Para demonstração, em todos os casos, é assumido que uma imagem é uma matriz com i linhas e j colunas e em cada célula temos três valores que se referem aos componentes vermelho, verde e azul (RGB), respectivamente.

Para demonstrar a aplicação das técnicas utilizaremos as imagens exemplo da Figura 3.6. Estas imagens são da Avenida Álvaro Calheiros na cidade de Maceió.

Figura 3.6 – Imagens utilizadas como exemplo nas técnicas de processamento de imagens



(a) Imagem inicial com o estacionamento vazio



(b) Imagem com o estado atual do estacionamento

Fonte: Autor, 2014

3.3.1 Subtração

Esta técnica possui apenas duas etapas de processamento de imagens. A primeira etapa é a subtração da imagem recebida pela requisição do servidor e uma imagem anterior com o estacionamento vazio. No resultado desta operação é aplicada uma técnica de binarização.

Para utilizar a técnica *Subtração* é necessário possuir uma imagem inicial com o estacionamento vazio e com a área de busca previamente demarcada e uma outra imagem do estacionamento com seu estado corrente. A imagem com o estacionamento vazio é utilizada como um estado inicial do estacionamento. Ela é obtida no início da execução, todas as técnicas desenvolvidas consideram esta imagem inicial.

Figura 3.7 – Imagens convertidas para tons de cinza



(a) Imagem inicial com o estacionamento vazio em tons de cinza



(b) Imagem com o estado atual do estacionamento em tons de cinza

Fonte: Autor, 2014

Ambas imagens são convertidas para tons de cinza (Figura 3.7). Basicamente, a conversão para tom de cinza transforma a imagem no sistema RGB para uma escala de cinza. A conversão para tons de cinza é importante pois, utilizando como exemplo o sistema RGB, ao invés de representar um *pixels* por um conjunto de três valores, ele será representado por apenas um valor variando de 0 até 255. O método adotado neste trabalho é o da Luminosidade (Gonzalez & Woods, 1992). Este método foi escolhido pois se aproveita do fato do olho humano não detectar as cores uniformemente. Humanos percebem o verde de forma mais forte do que o vermelho, e o vermelho mais forte que o azul. Por este motivo, este método leva em conta uma soma ponderada,

$$I_{i,j} = R_{i,j} \times 0.3 + G_{i,j} \times 0.59 + B_{i,j} \times 0.11,$$

onde R , G e B são os valores dos componentes vermelho, verde e azul respectivamente.

Com as duas imagens em tons de cinza, é realizada a operação de subtração entre elas (Figura 3.8). O objetivo da utilização da técnica de subtração é separar os elementos presentes na imagem do estado corrente do estacionamento que não estão presentes na imagem do estacionamento vazio. A operação de subtração de *pixels* (Acharya & Ray, 2005) utiliza duas imagens como entrada e produz como saída uma terceira imagem cujo os valores dos *pixels* são simplesmente a subtração dos valores dos *pixels* da primeira imagem com os respectivos valores dos *pixels* na segunda imagem. A subtração de duas imagens (assumindo as imagens como X e Y) é realizada em um único passo, a saída dos valores dos *pixels* é dado por,

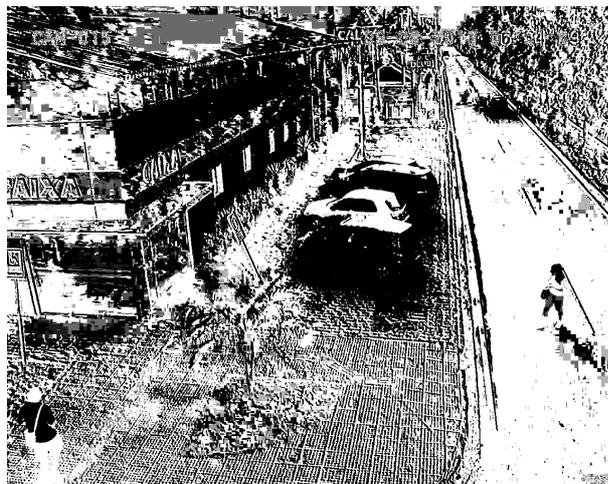
$$I_{i,j} = \begin{cases} X_{i,j} - Y_{i,j}, & \text{se } X_{i,j} - Y_{i,j} \geq 0 \\ 0, & \text{caso contrário} \end{cases}$$

Se o valor da operação $X_{i,j} - Y_{i,j}$ for menor que zero, $P_{i,j}$ assume o valor zero, isto foi feito pois os valores dos *pixels* da imagem não podem assumir valores negativos, é preciso que os valores dos *pixels* estejam entre 0 e 255. Se os valores dos *pixels* das imagens forem vetores ao invés de valores escalares (e.g. imagens coloridas) então os valores dos componentes individuais (e.g. componentes vermelho, verde e azul) são simplesmente subtraídos separadamente e produzem o valor de saída.

Por fim, é aplicada a técnica de binarização no resultado obtido da etapa anterior (Figura 3.9, etapa 3). Binarização (ou *thresholding*) é um método de segmentação de imagens que tem como objetivo transformar uma imagem em tons de cinza para uma imagem binária (dois valores possíveis para o valor do *pixel*), em geral em preto e branco.

Na literatura, pode-se encontrar diversos tipos de técnicas de binarização (Sezgin & Sankur, 2004). A técnica padrão de binarização de uma imagem I , com i linhas e j colunas, consiste na escolha de um valor arbitrário (κ), que servirá como um limite para identificar qual o valor que o *pixel* irá assumir.

Figura 3.8 – Resultado da aplicação da operação de subtração entre as imagens



Fonte: Autor, 2014

$$I_{i,j} = \begin{cases} 1, & \text{se } I_{i,j} \geq \kappa \\ 0, & \text{se } I_{i,j} < \kappa \end{cases}$$

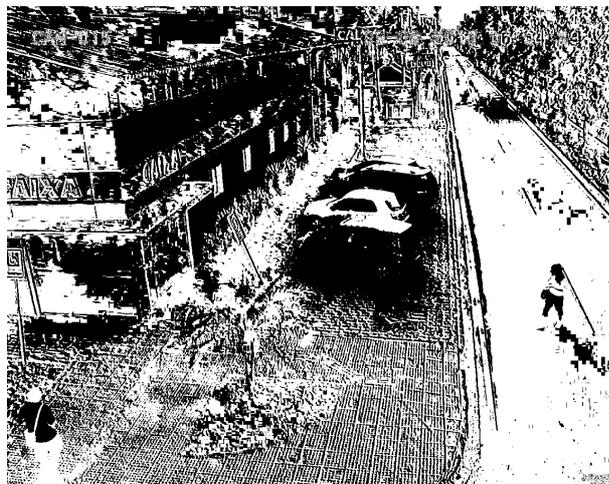
Esta técnica funciona visitando cada *pixel* ($I_{i,j}$) da imagem, e definindo o valor do *pixel* de acordo com o valor do limiar (κ). Neste trabalho utilizamos o método de Otsu (Otsu, 1979), tal método é utilizado para realizar a binarização de imagens utilizando uma abordagem de conjuntos. O Algoritmo 1 assume que a imagem a ser binarizada contém duas classes de *pixels* (primeiro plano e fundo), então ele calcula o limiar ótimo separando estas duas classes. A ideia é escolher um limiar que resulte na minimização da variância intra-classes. Cada classe possui suas próprias características, ou seja, sua média e desvio-padrão.

Algoritmo 1: Pseudo código do Algoritmo de Otsu

- 1 **inicio**
 - 2 Calcule o histograma da imagem em tons de cinza;
 - 3 Calcule o peso w e a variância σ^2 tanto do plano de fundo quanto do primeiro plano;
 - 4 Calcule a variância intra classe σ_w^2 ;
 - 5 Repita para todo possível valor do limiar
 - 6 **fin**
-

A imagem resultante é utilizada para determinar a existência de vagas de estacionamento. Para executar este processo, é realizada uma busca na imagem resultado por conjuntos de *pixels* brancos. O tamanho destes conjuntos são determinados a priori e utilizados para representar uma vaga de estacionamento livre. Assumindo que o conjunto seja um retângulo de tamanho

Figura 3.9 – Resultado da aplicação da operação de binarização



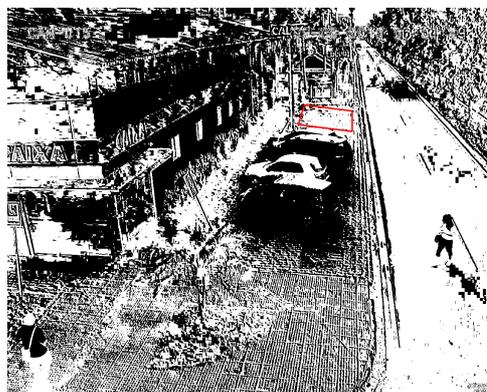
Fonte: Autor, 2014

$i \times j$, e $I_{i,j}$ seja a imagem original, temos que,

$$R = \sum_{m=0}^i \sum_{n=0}^j I_{m,n}.$$

caso R seja igual ao tamanho do conjunto de *pixels* determinado no início da busca, o espaço é identificado como uma vaga disponível. Uma demonstração da busca por vagas de estacionamento pode ser visualizada na Figura 3.10. A área demarcada é considerada como uma vaga na busca realizada. É utilizado o mesmo processo para verificar a existência de vagas de estacionamento em todas as técnicas utilizadas. É importante ressaltar que em todos os algoritmos, ao encontrar uma vaga disponível, a busca é interrompida. A técnica *Subtração* possui uma complexidade computacional na ordem de $O(n^2)$, onde n representa a área da imagem.

Figura 3.10 – Resultado da busca por vaga de estacionamento



Fonte: Autor, 2014

3.3.2 Equalização com Subtração

A segunda técnica desenvolvida é chamada de *Equalização com Subtração*. Esta técnica é composta pela técnica *Subtração* (apresentada anteriormente), com o adicional de aplicar a técnica de equalização do histograma em cada imagem antes de executar o processo de subtração.

Da mesma maneira que a técnica *Subtração*, para executar a técnica *Equalização com Subtração* é necessário utilizar uma imagem com o estacionamento vazio e uma imagem com o estacionamento no estado atual. A imagem com o estacionamento vazio é utilizada como estado inicial do estacionamento, ela é obtida no início da execução. O primeiro passo para execução da técnica é realizar a equalização do histograma em ambas imagens (Figura 3.11).

Figura 3.11 – Resultado da aplicação da operação de equalização de histogramas



(a) Imagem inicial com o estacionamento vazio



(b) Imagem com o estado atual do estacionamento

Fonte: Autor, 2014

Equalização de histogramas (Marques Filho & Vieira Neto, 1999) é uma técnica que realiza o ajuste das intensidades de uma determinada imagem para melhorar o contraste, tal ajuste é realizado de acordo com o histograma desta imagem em tons de cinza. Basicamente, realizar a equalização do histograma significa obter a máxima variância do histograma de uma imagem, obtendo assim uma imagem com um melhor contraste. O contraste é uma medida qualitativa que está relacionada com a distribuição dos tons de cinza em uma imagem. Neste trabalho é utilizado a técnica de equalização global de histograma. Na equalização global o processo é aplicado levando-se em consideração todos os tons de cinza presentes na imagem. O Algoritmo 2 apresenta o processo de equalização do histograma.

Com as imagens equalizadas, é, então, aplicada a técnica *Subtração*, apresentada anteriormente. A principal vantagem em utilizar a equalização de histogramas antes de realizar a subtração das imagens é que este processo realça diferenças sutis entre níveis de tons de cinza próximos e leva, na maioria dos casos, a um aumento substancial no nível de detalhes perceptíveis.

Da mesma maneira, a imagem resultante é utilizada para determinar a existência de vagas de estacionamento livres. Para tal, é realizada a busca por conjuntos de *pixels* brancos na imagem.

Algoritmo 2: Algoritmo para equalização do histograma**Entrada:** Imagem f , Quantidade de tons de cinza L ;**Saída:** Imagem s ;1 **inicio**

2 Obter o histograma normalizado da imagem

$$\rho_n = \frac{\# \text{ pixels com intensidade } n}{\# \text{ total de pixels}}, n = 0, 1 \dots L - 1;$$

3 **para cada** *Nível de cinza no histograma* **hacer**4 Aplicar a transformação de equalização $[(L - 1) \sum_{n=0}^{f_{i,j}} \rho_n]$ Armazenar o valor obtido em $s_{i,j}$ 5 **fin**6 **fin**

A técnica *Equalização com Subtração* possui uma complexidade computacional na ordem de $O(n^2)$, onde n representa a área da imagem.

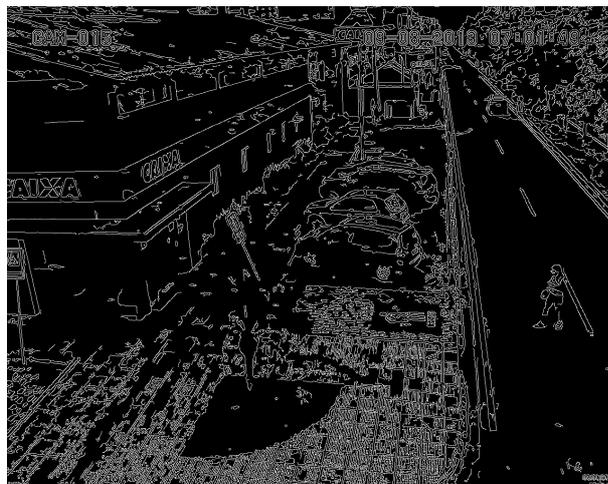
3.3.3 Dilatação após Detecção de Bordas

A última técnica considerada neste trabalho é chamada de *Dilatação após Detecção de Bordas*. Esta técnica aplica um método de detecção de bordas na imagem para depois utilizar métodos de binarização e dilatação.

Diferentemente das outras técnicas, esta abordagem não necessita de uma imagem do estacionamento vazio, pois não existe um processo de subtração entre a imagem atual do estacionamento e a imagem do estacionamento vazio, porém é preciso realizar o pré processamento para demarcar a área onde será realizada a busca por vagas de estacionamento. O funcionamento desta técnica se inicia convertendo a imagem corrente do estacionamento para tom de cinza.

Após isto, é aplicado o processo de detecção de bordas (Figura 3.12). Uma borda pode ser definida como a fronteira entre duas regiões de níveis de cinza diferentes. Em processamento de imagens, a detecção de bordas tem como objetivo encontrar pontos onde ocorram mudanças repentinas no brilho da imagem, estas mudanças são tidas como descontinuidades. A detecção de bordas é essencialmente uma operação de identificação de mudanças locais significativas nos tons de uma imagem. A vantagem da utilização da técnica de detecção de bordas é demarcar os contornos dos objetos presentes na imagem, em nosso problema, obteremos o contorno dos veículos presentes.

Para a realizar a detecção de bordas, uma das abordagens existentes é a aplicação de máscaras de convolução, também conhecidas como operadores 2×2 ou 3×3 , dependendo de seu tamanho. Na literatura, existem diversos métodos para detecção de borda (Ziou & Tabbone, 1998), neste trabalho é aplicado o operador de Prewitt (Prewitt, 1970). O operador de Prewitt funciona com dois *kernels* de convolução (gradientes), um que percorre a imagem na horizontal e outro que percorre a imagem na vertical (G_x e G_y) respectivamente.

Figura 3.12 – Resultado da aplicação da operação de detecção de bordas**Fonte: Autor, 2014**

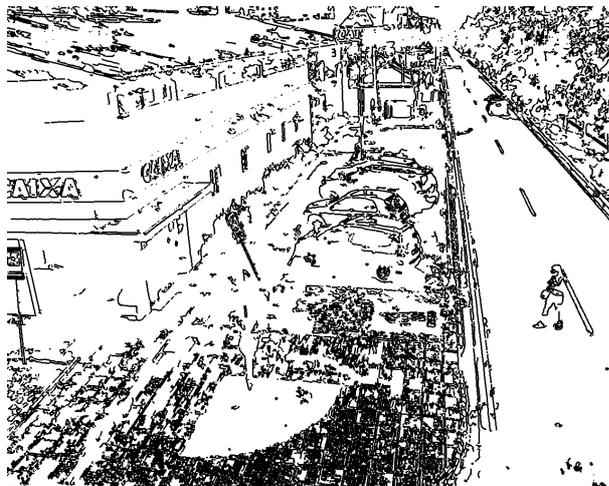
$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -1 & 0 & +1 \\ -1 & 0 & +1 \end{bmatrix} \quad G_y = \begin{bmatrix} +1 & +1 & +1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

Estes *kernels* podem ser aplicados separadamente na imagem de entrada, produzindo medições do gradiente em cada orientação. Assumindo uma imagem de entrada I , estes *kernels* são aplicados pela operação $I * G_x$ e $I * G_y$, para geração do gradiente horizontal e do gradiente vertical respectivamente. Estas operações devem ser aplicadas em toda imagem I . Eles podem ser combinados para encontrar o gradiente magnitude, dado por

$$|\nabla G| = \sqrt{G_x + G_y}$$

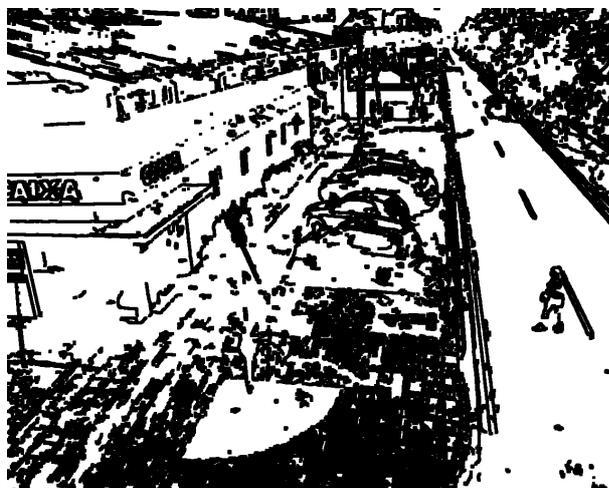
Na imagem resultante da aplicação da detecção de bordas, é aplicada a técnica de binarização (Figura 3.13), novamente é utilizado o método de Otsu. A ideia da aplicação da binarização após a detecção de bordas é aumentar o destaque das bordas que foram detectadas no processo anterior.

Por fim, é aplicada a técnica de dilatação (Figura 3.14). Dilatação (Dougherty, 1992a) é um dos operadores básicos na área de morfologias matemáticas. É tipicamente aplicado em imagens binárias, porém atualmente existem versões que trabalham com imagens em tons de cinza e coloridas. O efeito básico deste operador em imagens binárias é o aumento gradual dos limites das regiões dos *pixels* do primeiro plano (e.g. *pixels* brancos geralmente). Assim, as áreas com *pixels* do primeiro plano aumentam de tamanho, enquanto que os buracos existentes nestas regiões tornam-se menores. O objetivo na utilização da dilatação nesta etapa, é diminuir a quantidade de buracos criados após a aplicação da binarização, com isto os objetos encontrados

Figura 3.13 – Resultado da aplicação da operação de binarização

Fonte: Autor, 2014

na imagem deverão ser mais visíveis.

Figura 3.14 – Resultado da aplicação da operação de dilatação

Fonte: Autor, 2014

O operador de dilatação precisa de dois dados como entrada. O primeiro é a imagem que sofrerá o processo de dilatação, enquanto que o segundo é um *kernel* (elemento estrutural). Este *kernel*, em geral pequeno, é o elemento estrutural que determina o efeito da dilatação na imagem de entrada, quanto maior o *kernel*, maior será o efeito produzido pelo operador. Matematicamente, a dilatação pode ser demonstrada como a seguinte relação

$$A \oplus B = \{c \mid c = a + b, a \in A, b \in B\},$$

onde A representa a imagem sendo operada e B representa o elemento estrutural.

Como nas técnicas apresentadas anteriormente, a imagem resultante é utilizada para determinar a existência de vagas de estacionamento. Da mesma maneira, é utilizada a busca por conjuntos de *pixels* brancos que representam a vaga de estacionamento vazio. A técnica *Dilatação após Detecção de Bordas* possui uma complexidade computacional na ordem de $O(n^2)$, onde n representa a área da imagem.

4 Avaliações e resultados obtidos

Neste capítulo será apresentada a avaliação da qualidade das técnicas de processamento de imagens aplicadas para detectar vagas de estacionamento, considerando as especificidades aqui apresentadas. Afim de identificar a melhor técnica de processamento de imagens, avaliamos a quantidade de acertos de cada imagem em um conjunto de imagens obtidas em um experimento. Em seguida será avaliado, via experimentação, o desempenho do sistema proposto utilizando apenas a melhor técnica de processamento de imagem dentre as três técnicas propostas. Por último, é apresentada uma avaliação via simulação para validação da proposta em cenários de maiores magnitudes. O objetivo da simulação é identificar o comportamento da arquitetura proposta de acordo com variações em certas características.

4.1 Especificidade dos experimentos

As câmeras utilizadas no experimento foram as DCS-2130 da empresa DLINK. No entanto, assumindo o processamento e a comunicação é realizada por uma placa embutida, qualquer tipo de câmera pode ser utilizada. As câmeras DCS-2130 capturam imagem em alta resolução (HD) e possuem conectividade *WIFI* ou *ethernet*. O principal problema em sua utilização nos experimentos é o fato dela não possui um controle de brilho manual, onde não é possível configurar a qualidade das imagens quando estiverem contra a luminosidade do sol.

As placas embutidas nas câmeras foram as placas Panda. Tais placas possuem um processador ARM *dual-core* com 1.2 GHz de *clock*, 1 GB de memória RAM e com conectividade *WIFI* ou *ethernet*. As placas Panda tem funcionamento semelhante a um computador pessoal, para tal é preciso a escolha de um sistema operacional, neste trabalho foi escolhido instalar o sistema operacional Ubuntu-arm 12.04. Por ser uma placa com um custo relativamente baixo e com uma capacidade de processamento aceitável, a placa Panda se torna uma escolha apropriada para este tipo de aplicação. Por exemplo, comparadas a computadores pessoais, as placas de processamento equivalem até 1/20 do valor destes computadores, isto é, ao invés de comprar um computador pessoal, poderíamos comprar em torno de 20 placas de processamento. Se for considerado a utilização de um servidor robusto, esta razão pode superar 1/1000, ou seja, com o valor gasto no servidor poderíamos adquirir em torno de 1000 placas de processamento. Como será visto na seção 4.4, em nosso trabalho são utilizadas no máximo 454 câmeras, isto significa

que seriam necessárias 454 placas de processamento, isto equivale a uma economia de aproximadamente 55% no custo da aplicação, comparado a utilização de um servidor central. Neste trabalho é utilizado o mapeamento de uma câmera para uma placa de processamento, porém é possível desenvolver uma abordagem onde uma placa de processamento gerencie, por exemplo, todas as câmeras de uma única residência.

Como descrito anteriormente, consideramos três arquiteturas distintas para a detecção de vagas de estacionamento:

Embutida: é uma arquitetura *ad-hoc* onde apenas as câmeras são responsáveis por todo o processamento de imagens e pela comunicação com os clientes.

Centralizada: utiliza uma arquitetura cliente-servidor onde o servidor central é responsável por todo o processamento das imagens e pela comunicação com os clientes.

Híbrida: é uma arquitetura onde as câmeras lidam com a comunicação com os clientes e um servidor central é responsável por todo o processamento das imagens.

Nas propostas *Centralizada* e *Híbrida*, foi utilizado um servidor com um processador Intel Core2Duo P8600 com 2.4 GHz de *clock*, 4 GB de memória RAM e com o sistema operacional Ubuntu 12.04 64 bits. Todas as técnicas de processamento de imagens foram implementadas utilizando uma biblioteca de visão computacional conhecida como OPENCV¹, escrita na linguagem de programação C++.

Foram utilizados sete pontos de monitoramentos, cada um com uma câmera responsável, capturando imagens de duas áreas de estacionamento diferentes, na Universidade Federal de Alagoas, durante 4 horas contínuas (das 8:00 até as 12:00). No momento do experimento o tempo estava ensolarado e com poucas nuvens no céu. A informação a respeito do clima no momento do experimento é importante pois estas variações causam impacto nos resultados do processamento das imagens. As áreas de estacionamento monitoradas possuíam uma área em formato retangular, a primeira com 100 m^2 e a segunda com 500 m^2 . Cada câmera era responsável por monitorar um ponto diferente de cada área de estacionamento. Todas as câmeras utilizadas no experimento foram instaladas na mesma altura (em torno de 3 m) e com distância uniforme entre elas (em torno de 2 m). Como cada câmera monitora diferentes espaços das áreas de estacionamento, isto resulta em imagens com diferentes características, como por exemplo, algumas imagens apresentam sombras e outras uma alta luminosidade.

As câmeras foram configuradas para capturar e salvar as imagens em um repositório em uma máquina local a cada movimento detectado em sua visada. Pode acontecer das câmeras capturem imagens de quando pessoas cruzam a sua visada, estas imagens não foram descartadas em nosso experimento. Quando as câmeras salvam uma imagem no repositório, elas também salvam o horário na qual a imagem foi capturada. Com isso, foi criado um repositório de imagens que é utilizado nas avaliações deste trabalho. Neste repositório, cada imagem é indexada

¹<http://opencv.org/>

em uma linha do tempo (Figura 4.1) e cada câmera possui a sua própria linha do tempo. As imagens são associadas ao tempo na qual elas foram capturadas. A ideia da utilização desta linha temporal é para podermos simular requisições em determinada câmera em determinado horário. O repositório criado por este experimento possui 3320 imagens, cada uma relacionada com o instante que foi capturada (hh:mm:ss).

Figura 4.1 – Exemplo de uma linha temporal



Fonte: Autor, 2014

A seguir será apresentada a avaliação da qualidade das técnicas de processamento de imagens aplicadas para detectar vagas de estacionamento, considerando as especificidades aqui apresentadas. Em seguida será avaliado, via experimentação, o desempenho do sistema proposto utilizando apenas a melhor técnica de processamento de imagem. Por último, é apresentado uma avaliação via simulação para validação da proposta em cenários de maiores magnitudes.

4.2 Avaliação das técnicas de processamento de imagens

Antes de apresentar a avaliação do tempo de execução das abordagens propostas, é necessário, em primeiro lugar, escolher a técnica de processamento de imagem mais adequada para o problema da detecção de vagas de estacionamento. Para realizar isto, é considerado uma avaliação baseada em matriz de confusão (Stehman, 1997). Matriz de confusão é um modelo específico de tabela que permite a visualização do desempenho de um algoritmo, tipicamente de detecção. Cada linha da matriz representa uma instância da classe de predição (valor inferido), enquanto que cada coluna representa uma instância da classe real. Ao analisar o resultado da aplicação do algoritmo avaliado, deve-se combinar as instâncias das classes de valores reais e inferidos e completar a matriz com a quantidade de valores obtidos. Uma ilustração da matriz de confusão considerada neste trabalho é apresentada na Tabela 4.1. São avaliadas as três técnicas de processamento de imagens propostas, porém apenas a com melhor resultado para a aplicação deste trabalho será utilizada.

Assumindo que existem duas classes para inferir, vagas *livres* e \sim *livres*, onde *livre* significa

Tabela 4.1 – Uma ilustração geral de uma matriz de confusão

		TÉCNICA		
		Valores Reais		
		<i>livre</i>	<i>~livre</i>	
Inferido	<i>livre</i>	<i>TP</i>	<i>FP</i>	$PPV \cong \frac{\sum TP}{\sum TOP}$
	<i>~livre</i>	<i>FN</i>	<i>TN</i>	$NPV \cong \frac{\sum TN}{\sum TON}$
		$TPR \cong \frac{\sum TP}{\sum CP}$	$SPC \cong \frac{\sum TN}{\sum CN}$	$ACC \cong \frac{\sum TP + \sum TN}{\sum CP + \sum CN}$

Fonte: Autor, 2014

que existem vagas disponíveis e *~livre* significa que não existem vagas disponíveis, a informação dada pela matriz de confusão será:

- Quantidade de verdadeiros positivos (*TP*), no caso deste trabalho ocorre se o valor inferido e o valor real forem ambos *livre*.
- Quantidade de falsos positivos (*FP*), no caso deste trabalho ocorre se o valor inferido for *livre* e o valor real for *~livre*.
- Quantidade de falsos negativos (*FN*), no caso deste trabalho ocorre se o valor inferido for *~livre* e o valor real é *livre*.
- Quantidade de verdadeiros negativos (*TN*), no caso deste trabalho ocorre se o valor inferido e o valor real forem ambos *~livre*.
- O valor da taxa de verdadeiros positivos (*TPR*), ou sensibilidade, é conhecida como a taxa de acerto, e.g., quantos valores *livre* o sistema inferiu que são realmente *livre*. O valor $\sum CP$ representa o total de valores que são positivos ($TP + FN$). O melhor resultado para esta taxa é 1, quando o sistema acerta todas as imagens positivas.
- O valor da especificidade (*SPC*), ou taxa de verdadeiros negativos, é relativa a quantidade de valores que o sistema inferiu serem *~livre* que são realmente *~livre*. O melhor resultado para esta taxa é 1, quando o sistema acerta todas as imagens negativas.
- O valor da precisão ou valor da inferência positiva (*PPV*), é relativa a todos os testes que foram inferidos como *livre*, quantos era realmente *livre*. O valor $\sum TOP$ é o total de valores positivos inferidos ($TP + FP$).
- O valor da inferência negativa (*NPV*), é relacionada a: de todos os testes que foram inferidos como *~livre*, quantos são realmente *~livre*. O valor $\sum TON$ é a quantidade total de valores negativos inferidos ($TN + FN$).

- O valor da acurácia (*ACC*) é a proporção de valores verdadeiros (tanto verdadeiros positivos quanto verdadeiros negativos) na população. Uma acurácia de valor 1 significa que os valores obtidos são exatamente os mesmo que os valores dados.

Para criar a matriz de confusão para os resultados obtidos neste trabalho, foi preciso identificar manualmente em cada imagem capturada (3320 imagens) a presença de vagas de estacionamento vazias. Para ser considerada uma vaga de estacionamento, o espaço deve possuir 2,3 m de largura e 5,5 m de comprimento. Estes valores foram assumidos de acordo com o padrão adotado pela Companhia de Engenharia de Tráfego da cidade de São Paulo (Cidade de São Paulo, 2014). Para cada técnica proposta foi quantificado a quantidade de acertos e falhas (falsos positivos e falsos negativos) para indicar vagas de estacionamento livres. A Tabela 4.2 apresenta a matriz de confusão completa para cada técnica avaliada considerando o resultado utilizando todas as câmeras do experimento.

Tabela 4.2 – Matriz de confusão para cada técnica

SUBTRAÇÃO

		Valores reais		
		<i>livre</i>	<i>~livre</i>	
Inferido	<i>livre</i>	TP = 1075	FP = 182	PPV \cong 0,855
	<i>~livre</i>	FN = 1966	TN = 97	NPV \cong 0,047
		TPR \cong 0,353	SPC \cong 0,347	ACC \cong 0,353

EQUALIZAÇÃO COM SUBTRAÇÃO

		Valores reais		
		<i>livre</i>	<i>~livre</i>	
Inferido	<i>livre</i>	TP = 427	FP = 154	PPV \cong 0,735
	<i>~livre</i>	FN = 2614	TN = 125	NPV \cong 0,045
		TPR \cong 0,140	SPC \cong 0,448	ACC \cong 0,166

DILATAÇÃO APÓS DETECÇÃO DE BORDAS

		Valores reais		
		<i>livre</i>	<i>~livre</i>	
Inferido	<i>livre</i>	TP = 2939	FP = 128	PPV \cong 0,958
	<i>~livre</i>	FN = 102	TN = 151	NPV \cong 0,597
		TPR \cong 0,966	SPC \cong 0,541	ACC \cong 0,931

Fonte: Autor, 2014

Na Tabela 4.2 é possível notar que a técnica mais apropriada, para os experimentos realizados neste trabalho, é a técnica *Dilatação após Detecção de Bordas*. Esta técnica possui os maiores valores para a acurácia (*ACC*) e a sensibilidade (*TPR*), ambos próximos de 1. Isto ocorre porque, diferentemente de outras técnicas, a forma dos veículos são levadas em consideração e, conseqüentemente, elas são detectadas com maior facilidade. A alta taxa do valor positivo

inferido ($PPV = 0,958$) na técnica *Dilatação após Detecção de Bordas* indica que muitos dos resultados positivos provenientes dos testes são valores verdadeiros positivos. Por outro lado, a pior técnica foi a *Equalização com Subtração* com ACC e TPR próximos de zero (0,166 e 0,140 respectivamente), i.e., a cada seis imagens processadas o algoritmo acertava apenas uma. Isto ocorria devido a variação da luminosidade nas imagens utilizadas. O processo de equalização do histograma, utilizado nesta técnica, em geral aumenta o contraste global das imagens. Se a imagem possui um grande brilho, após a aplicação da equalização do histograma ela perde a maioria de suas características. Isto ocorre, por exemplo, quando o sol reflete nos carros estacionados. Neste caso o algoritmo infere que existe uma vaga livre naquele espaço, causando um erro (falso positivo).

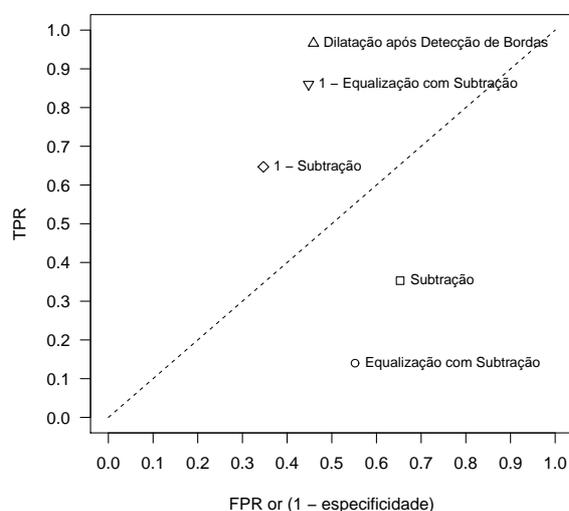
Em ambas técnicas, *Subtração* e *Equalização com Subtração*, os valores PPV são aceitáveis (0,855 e 0,735 respectivamente), mas seus NPV são próximos de zero (0,047 e 0,045 respectivamente). Esta métrica é relativa aos valores negativos, quando é próxima de zero significa que existem mais valores falsos negativos do que valores verdadeiros positivos.

O valor de SPC (especificidade) na técnica *Dilatação após Detecção de Bordas* é 0,541. Este resultado ocorre devido a variação de luminosidade nas imagens. Quando o processamento de imagens é executado, imagens com grande quantidade de brilho criam grandes conjuntos de *pixels* brancos, este fato dificulta a busca por vagas de estacionamento vazias, causando assim um aumento no número de resultados falsos positivos.

O espaço ROC é uma técnica que tem como objetivo facilitar na avaliação e comparação entre algoritmos de classificadores binários (Spackman, 1989). No eixo x temos a taxa de falsos positivos, enquanto que no eixo y temos o valor da taxa de verdadeiros positivos. O ponto (0,0) representa uma estratégia que nunca gera uma classificação positiva, ou seja, não comete falsos positivos mas também não classifica nenhum verdadeiro positivo. A melhor técnica de processamento de imagens está no ponto no canto superior esquerdo (na coordenada [0,1]) do espaço ROC , representando 100% de sensibilidade (sem falsos negativos) e 100% de especificidade (sem falsos positivos). Um mecanismo aleatório, que segue uma distribuição uniforme, de busca por vagas de estacionamento livre daria um ponto qualquer na linha diagonal. Pontos sobre a diagonal representam bons resultados de classificação (melhores que o aleatório), pontos abaixo da linha diagonal representam classificadores ruins (piores que aleatório). Note que se existir um classificador pior que o aleatório, basta simplesmente inverter seu parâmetro de detecção para obter um bom classificador. O inverso de cada técnica significa que quando o algoritmo retornar *livre* para uma determinada vaga, a resposta considerada será \sim *livre* e vice-versa. Na Figura 4.2, podemos ver o espaço ROC (Metz, 2006) para cada técnica.

Adicionalmente, a Figura 4.2 apresenta dois pontos extras (*1 - Subtração* e *1 - Equalização com Subtração*), representando o inverso das técnicas *Subtração* e *Equalização com Subtração*. Este processo foi realizado para identificar se essas técnicas com seus parâmetros de detecção invertidos poderiam ser melhores do que a técnica *Dilatação após Detecção de Bordas*. Para

Figura 4.2 – Espaço ROC para as três técnicas apresentadas



Fonte: Autor, 2014

tal, foi calculado a distância euclidiana dos pontos *Dilatação após Detecção de Bordas*, *1 - Subtração* e *1 - Equalização com Subtração* até a coordenada (0,1) (classificação perfeita) e o resultado obtido foi 0,460, 0,480 e 0,495 respectivamente. Este resultado mostra que a técnica *Dilatação após Detecção de Bordas* permanece mais próxima da coordenada (0,1) mesmo quando os parâmetros de detecção das outras técnicas são invertidos, isto significa que a técnica mais apropriada para os cenários apresentados é a *Dilatação após Detecção de Bordas*.

4.3 Avaliação de desempenho via experimentação

A avaliação de desempenho das abordagens propostas considera o tempo médio gasto para executar uma requisição do usuário completa em cada proposta: *Embutida*, *Híbrida* e *Centralizada*. A requisição é formada pela soma do tempo de comunicação e o tempo de processamento. Nesta avaliação, a técnica de processamento de imagens utilizada é a *Dilatação após Detecção de Bordas*. Esta técnica foi escolhida baseada na avaliação apresentada na seção 4.2.

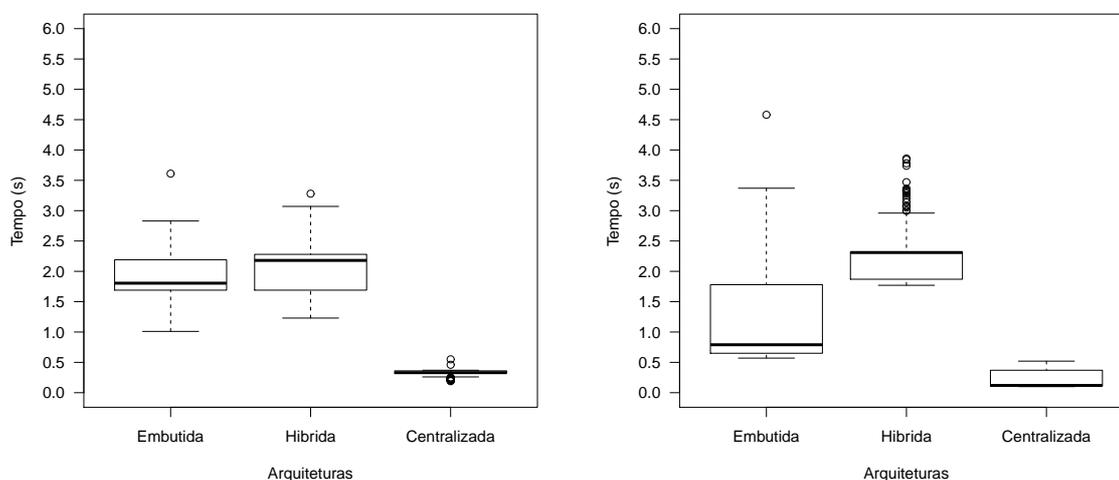
Como mencionado anteriormente, foram consideradas sete câmeras DCS-2130 para compôr o repositório de imagens. Sendo o repositório de imagens dado por $R = C_1, \dots, C_7$, onde C_i representa a câmera i . A quantidade de imagens em cada câmera é, respectivamente, 267, 496, 1294, 527, 314, 236, 186, totalizando $|R| = 3320$ imagens. Foi calculado os tempos médios gastos para executar uma requisição completa do usuário para cada C_i e para todo o repositório R . Considerando as câmeras separadamente, são apresentados os resultados apenas para a melhor (C_4) e pior (C_7) câmeras (Figuras 4.4(a) e 4.4(b) respectivamente). Adicionalmente, é apresentado o resultado considerando o repositório R por completo (Figura 4.4). É importante

ressaltar que o algoritmo *Dilatação após Detecção de Bordas* ao encontrar uma vaga disponível encerra o processamento da imagem, como o processamento ocorre na direção esquerda para direita, ele será mais veloz em imagens onde as vagas estiverem no início da imagem (mais a esquerda) do que em imagens onde as vagas estiverem no fim do estacionamento (mais a direita). Nas figuras 4.3 e 4.4 é utilizado o diagrama de caixa (McGill et al., 1978), este tipo de gráfico foi escolhido por possuir diversas informações agregadas, o diagrama de caixa apresenta onde estão localizados 50% dos valores mais prováveis, a mediana e os valores extremos (tanto inferior quanto superior). O eixo x dos gráficos apresenta as arquiteturas utilizadas na avaliação do sistema, enquanto que o eixo y possui o tempo médio das requisições em segundos.

Figura 4.4(a) apresenta o tempo médio gasto para executar uma requisição completa do usuário utilizando apenas as imagens capturadas por C_7 . A arquitetura *Embutida* obteve resultados um pouco melhores do que a abordagem *Híbrida*, apresentando uma variação de 1,01 s até 3,61 s. A abordagem *Híbrida* teve os resultados variando de 1,23 s até 3,28 s, com a maior concentração dos valores no intervalo [1,692 s, 2,277 s]. Os melhores resultados obtidos foram para a arquitetura *Centralizada* variando de 0,19 s até 0,55 s, porém a maioria dos valores estão concentrados no intervalo [0,32 s, 0,36 s]. Adicionalmente, foi calculado separadamente a média para cada arquitetura resultando em 1,94 s com intervalo de confiança [1,889 s, 1,991 s] e $\sigma = 0,35$; 2,01 s com intervalo de confiança [1,959 s, 2,060 s] e $\sigma = 0,35$, e 0,33 s com intervalo de confiança [0,323 s, 0,336 s] e $\sigma = 0,04$, respectivamente. Isto ocorre pois, na arquitetura *Centralizada* é utilizado um servidor para gerenciar as requisições e processar as imagens, obtendo assim uma vantagem com relação as arquiteturas *Híbrida* e *Embutida*. A arquitetura *Embutida* possui a vantagem de possuir um estágio a menos de comunicação do que a arquitetura *Híbrida*, justificando assim seus melhores resultados.

Figura 4.4(b) apresenta o tempo médio gasto para executar uma requisição completa do usuário utilizando apenas as imagens capturadas por C_4 . A arquitetura *Embutida* obteve resultados um pouco melhores do que a abordagem *Híbrida*, apresentando uma variação de 0,57 s até 4,58 s, com os valores concentrados no intervalo [0,65 s, 1,78 s]. A abordagem *Híbrida* teve os resultados variando de 1,77 s até 3,86 s, porém a maioria de seus resultados estão concentrados no intervalo [1,87 s, 2,32 s]. Os melhores resultados obtidos foram para a arquitetura *Centralizada* variando de 0,1 s até 0,52 s, porém a maioria dos valores estão concentrados no intervalo [0,11 s, 0,37 s]. Novamente, foram calculados separadamente a média para cada arquitetura resultando em 1,165 s com intervalo de confiança [1,128 s, 1,200 s] e $\sigma = 0,66$; 2,261 s com intervalo de confiança [2,189 s, 2,331 s] e $\sigma = 0,65$, e 0,2032 s com intervalo de confiança [0,196 s, 0,209 s] e $\sigma = 0,12$, respectivamente. Isto ocorre pois, na arquitetura *Centralizada* é utilizada um servidor para realizar gerenciar as requisições e processar as imagens, obtendo assim uma vantagem com relação as arquiteturas *Híbrida* e *Embutida*. A arquitetura *Embutida* possui a vantagem de possuir um estágio a menos de comunicação do que a arquitetura *Híbrida*, justificando assim seus melhores resultados quando comparadas.

Figura 4.3 – Avaliação dos tempos de execução da pior e melhor câmera



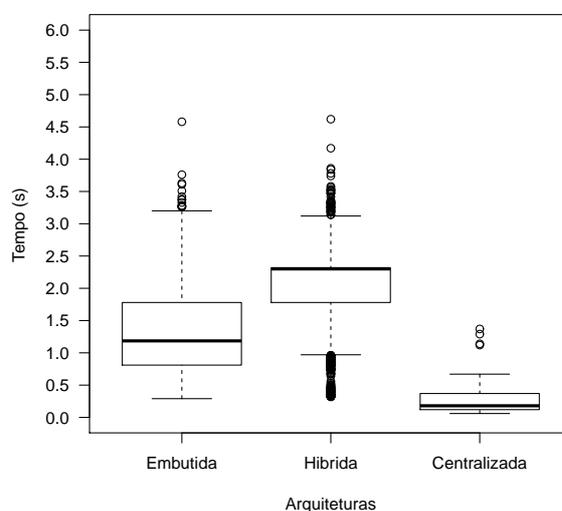
(a) Avaliação do tempo de execução da pior câmera (b) Avaliação do tempo de execução da melhor câmera

Fonte: Autor, 2014

Assumindo que todas as câmeras apresentam um comportamento similar (Figuras 4.4(a) e 4.4(b)), é possível avaliar o tempo médio gasto para executar uma requisição completa do usuário com todas as imagens do repositório (Figura 4.4). Similar as anteriores, a arquitetura *Embutida* tem resultados um pouco melhores do que a *Híbrida*, apresentando uma variação de 0,29 s até 4,58 s. A abordagem *Híbrida* tem resultados variando de 0,32 s até 4,17 s. Os melhores resultados foram na arquitetura *Centralizada* com valores variando de 0,06 s até 1,37 s, porém a maioria dos valores estão concentrados no intervalo [0,12 s, 0,37 s]. Novamente, foram calculados separadamente a média para cada arquitetura resultando em 1,357 s com intervalo de confiança [1,336 s, 1,379 s] e $\sigma = 0,62$; 2,03 s com intervalo de confiança [2,000 s, 2,065 s] e $\sigma = 0,52$, e 0,24 s com intervalo de confiança [0,236 s, 0,245 s] e $\sigma = 0,12$, respectivamente. Isto ocorre pois, na arquitetura *Centralizada* é utilizada um servidor para gerenciar as requisições e processar as imagens, obtendo assim uma vantagem com relação as arquiteturas *Híbrida* e *Embutida*. Novamente, por possuir um estágio a menos de comunicação, a arquitetura *Embutida* obteve melhores resultados quando comparada com a arquitetura *Híbrida*.

A fim de entender melhor o comportamento dos resultados previamente apresentados, a Figura 4.5 apresenta o gráfico de densidade para cada arquitetura proposta, i.e., o tempo de execução (eixo x) pela aproximação do número de requisições (eixo y). A Estimativa de Densidade do *Kernel* (Scott, 1992) é um método não paramétrico para estimar a densidade de curvas onde cada observação é ponderada pela distância a um valor central, conhecido como *kernel*. A ideia é centralizar cada observação x onde estimamos a densidade. Assumindo (x_1, x_2, \dots, x_n) como uma amostra distribuída de forma independente e idêntica desenhada a partir de uma distribui-

Figura 4.4 – Avaliação do tempo de execução para todas as câmeras



Fonte: Autor, 2014

ção com uma densidade desconhecida f . Estamos interessados em estimar a forma desta função f . Seu estimador de densidade do *kernel* é:

$$\hat{f}_h(x) = \frac{1}{n} \sum_{i=1}^n K_h(x - x_i) = \frac{1}{nh} \sum_{i=1}^n K\left(\frac{x - x_i}{h}\right),$$

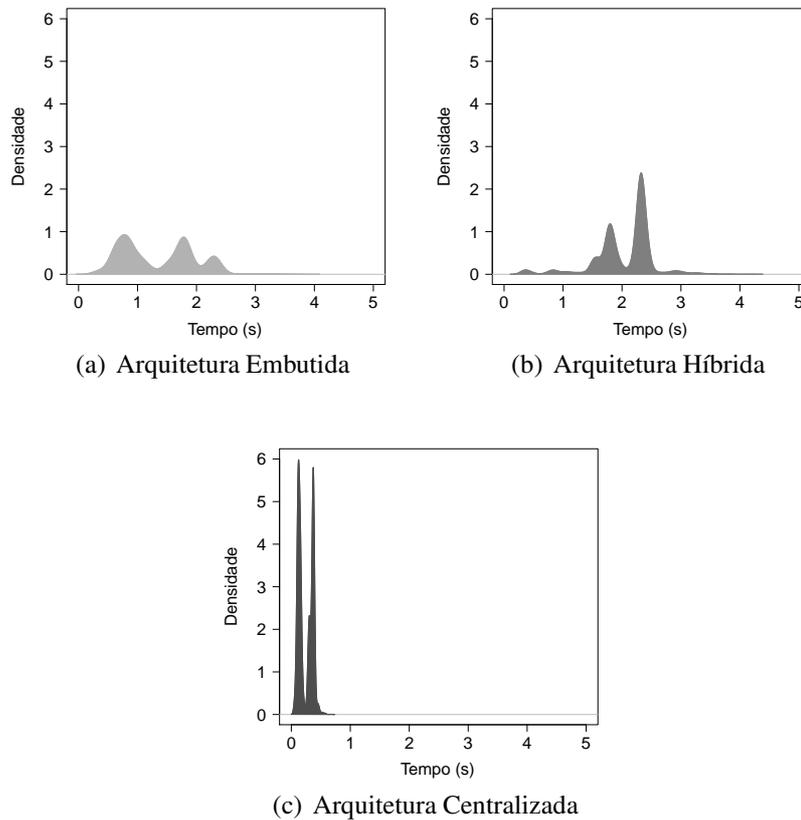
onde K é o *kernel* e $h > 0$ é um parâmetro de suavização chamado de largura de banda. Neste trabalho utilizamos o *kernel* Gaussiano dado por

$$K(t) = \frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}}$$

A Figura 4.5 apresenta a densidade para as arquiteturas *Centralizada*, *Híbrida* e *Embutida* levando em consideração o tempo médio de requisição de todas as câmeras. No eixo x temos o tempo médio das requisições em segundos e no eixo y o valor das densidades. É possível notar que em cada arquitetura existe duas modas distintas. Isto ocorre pois na base de dados as imagens apresentam duas características distintas: vagas livres de estacionamento no (i) início ou no (ii) fim do estacionamento. Basicamente, os algoritmos apresentados processam a imagem da direita para a esquerda, no início significa que existem vagas de estacionamento livres localizadas no lado esquerdo da imagem, conseqüentemente, no fim corresponde a vagas de estacionamento livres localizadas mais a direita da imagem. Assumindo que estas vagas de estacionamento estão no início, os algoritmos levam menos tempo para realizar o processamento, pois quando uma vaga livre é encontrada, o algoritmo finaliza o processamento. Da mesma maneira, quando uma vaga de estacionamento está no fim da imagem, o algoritmo leva mais

tempo pois é preciso processar mais partes da imagem.

Figura 4.5 – Gráfico de densidade para cada arquitetura avaliada



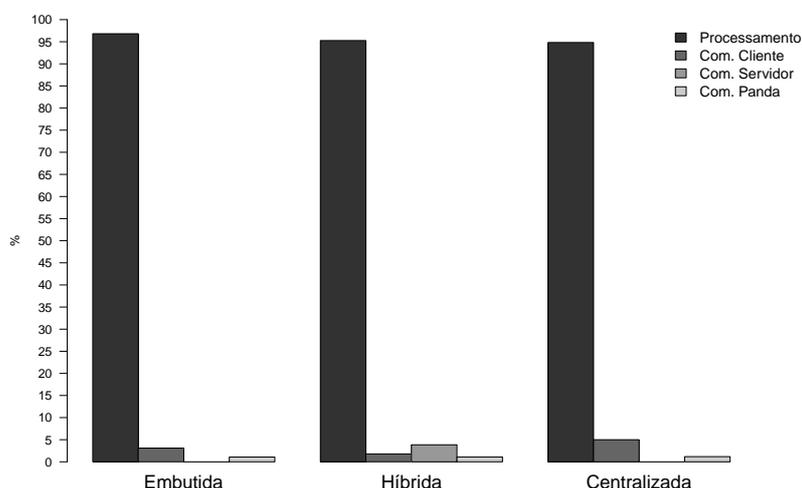
Fonte: Autor, 2014

A abordagem *Embutida* é mais homogênea do que a *Híbrida*, porém, ela é mais influenciada pelo processo de comunicação quando comparada com a abordagem *Centralizada* (Figura 4.6(a)). A abordagem *Híbrida* é a mais afetada pelo processo de comunicação, visto que ela possui um estágio a mais que as outras (Figura 4.6(b)). Neste caso, a comunicação justifica o aumento nos valores da segunda classe, por combinar o maior tempo de processamento e a comunicação realizada. Por fim, é perceptível que a abordagem *Centralizada* é a mais homogênea e a mais rápida dentre todas, pois suas modas possuem a mesma densidade (Figura 4.6(c)).

Na Figura 4.6, é apresentada uma análise mais detalhada dos tempos, considerando o tempo gasto em cada etapa de processamento específica em todas as abordagens. Primeiramente é apresentado o tempo para cada etapa na abordagem *Embutida*. Comunicação Panda é relativo ao tempo em que a placa Panda leva para receber uma requisição e enviar a resposta para o cliente. Comunicação do Cliente é relativo ao tempo gasto no lado do cliente, é dado pelo tempo total menos gasto na placa Panda. Processamento é o tempo que leva para aplicar a técnica de processamento de imagem (*Dilatação após Detecção de Bordas*). Aproximadamente 95 % do

tempo de execução é gasto na fase de processamento.

Figura 4.6 – Análise detalhada de cada arquitetura



Fonte: Autor, 2014

Da mesma forma, é apresentado o tempo para cada etapa na abordagem *Híbrida*. Aqui, a Comunicação do Servidor é relativo ao tempo que o servidor leva para enviar/receber a requisição da placa Panda. Cerca de 94 % do tempo de execução é gasto na fase de processamento.

Por último, temos o tempo para cada etapa na abordagem *Centralizada*. Comunicação do Servidor é relativo ao tempo que o servidor leva para receber uma requisição e enviar a resposta para o cliente. Aproximadamente 95 % do tempo de execução é gasto na fase de processamento, como era esperado.

Na arquitetura *Embutida* os valores são um pouco diferentes, a porcentagem para o tempo de processamento é maior que na arquitetura *Centralizada*, isto ocorre devido ao servidor possuir maior capacidade de processamento do que a placa Panda.

Em todas as arquiteturas propostas, o tempo gasto no local onde ocorre o processamento (servidor ou placa Panda) é basicamente utilizado na fase de processamento.

É importante ressaltar que a abordagem *Embutida* pode ser utilizada para reduzir o tráfego na rede, quando requisições simultâneas são utilizadas. Cada imagem nos cenários apresentados possuem 200 KB, assumindo que ocorram 1000 requisições simultâneas, teremos 200 MB de tráfego constante no servidor da abordagem *Centralizada*. Entretanto, na abordagem *Embutida* os 200 MB de tráfego são distribuídos entre as câmeras requisitadas reduzindo o tráfego na rede. Isto ocorre pois na abordagem *Embutida* não é necessário transmitir as imagens já que o processamento ocorre nas câmeras, enquanto que na abordagem *Centralizada*, todas as imagens tem que ser transmitidas para o servidor central.

4.4 Avaliação de desempenho via simulação

Nesta seção é apresentada a avaliação da arquitetura *Embutida*, que foi proposta neste trabalho, sendo comparada com outras duas arquiteturas, *Centralizada* e *Híbrida*, considerando uma simulação em larga escala utilizando os três cenários com requisições simultâneas. No experimento apresentado anteriormente, ocorria apenas uma requisição por vez. Por essa razão, surge a necessidade de realizar avaliações onde fosse possível elevar o nível de carga do sistema para analisar melhor o comportamento de cada arquitetura.

Utilizamos o simulador de redes SINALGO ([Distributed Computing Group, 2008](#)). O SINALGO é uma *framework* para testes e validações de algoritmos de rede. Diferentemente de outros simuladores que gastam boa parte do tempo simulando diferentes camadas da pilha de rede, o SINALGO foca nos algoritmos de rede e abstrai as camadas mais baixas. O SINALGO foi projetado para as redes sem-fio, mas não é limitado apenas a estas. O SINALGO foi desenvolvido utilizando a linguagem Java. Pode-se criar modelos 2D ou 3D das simulações realizadas. O SINALGO é um projeto *open source*, sendo disponibilizado sob a licença BSD.

O desenvolvimento e crescimento das cidades, em termos de quantidade populacional, implica no aumento de problemas que a tempos atrás não eram considerados críticos. A frota de veículos no Brasil cresceu em torno de 139% entre os anos de 2000 e 2010 ([Rodrigues, 2013](#)). Atualmente, existem cerca de um veículo para cada quatro habitantes no Brasil. Em 2012, a cidade de Maceió, de acordo com o IBGE (Instituto Brasileiro de Geografia e Estatística), possuía 152.399 automóveis² para uma população de 953.393 habitantes, nos dando uma razão de aproximadamente um carro para cada 6 habitantes. Mesmo sendo considerada uma cidade de pequeno porte, Maceió possui os mesmos problemas relacionados a trânsito do que grandes cidades como São Paulo e Rio de Janeiro. Dentre estes problemas podemos citar a dificuldade em encontrar locais para estacionar e o trânsito formado em suas principais avenidas em horários de pico. Por este motivo, utilizaremos a cidade de Maceió como ambiente para as simulações realizadas neste trabalho.

Para a simulação utilizaremos, como exemplo, um evento cultural ocorrido na orla de Maceió durante o ano de 2014. Este evento ocorria durante o final de semana. Criamos três situações distintas:

- Na primeira situação, utilizamos 100 requisições simultâneas. Esta situação pode ser comparada ao período prévio do evento, como por exemplo entre 2 e 3 horas antes do início, neste momento poucas pessoas estão se dirigindo ao evento, conseqüentemente o sistema irá receber menos requisições.
- Na segunda situação, utilizamos 500 requisições simultâneas. Esta situação é relativa ao período que antecede o evento, como por exemplo em torno de uma hora antes de seu

²<http://cod.ibge.gov.br/27FR>

início, neste momento o fluxo de chegada de pessoas ao evento começa a aumentar e conseqüentemente a quantidade de requisições irão aumentar.

- Na terceira situação, utilizamos 1000 requisições simultâneas. Esta situação caracteriza o período em torno de 30 minutos antes e durante o início do evento, este é o momento com o maior fluxo de pessoas chegando. Conseqüentemente a busca por vagas de estacionamento se torna a cada momento uma tarefa mais árdua, logo a tendência é o aumento na quantidade de requisições ao sistema.

Na simulação existem quatro elementos principais, sendo dois fixos e dois variáveis. São ditos elementos fixos, aqueles elementos no qual suas posições não se alteram durante toda a simulação. Um dos elementos fixos na simulação são 14 pontos, chamados neste trabalho de postes, interligados entre si. Estes postes podem se comunicar diretamente, facilitando o encaminhamento das mensagens dentro da rede. Além destes postes, foram fixados também o posicionamento de 54 câmeras. O posicionamento destas câmeras foram escolhidas de acordo com a localização das câmeras no projeto de vídeo monitoramento da cidade de Maceió. Este projeto foi desenvolvido pelo governo do estado de Alagoas e teve início em novembro de 2012. Inicialmente contava com 27 câmeras distribuídas pela cidade de Maceió, atualmente conta com cerca de 82 câmeras. As câmeras utilizadas no projeto são do modelo AXIS Q6034-E, captam imagens em HD e possuem *zoom* entre 500 e 800 m, algumas destas câmeras possuem uma visada panorâmica, podendo girar 360° em torno de seu eixo, todas estas câmeras são conectadas com o servidor central por fibra óptica. Este projeto também conta com uma sala de monitoramento que funciona durante 24 horas por dia. A ideia inicial do projeto era combater a criminalidade na cidade de Maceió, porém atualmente, além disto, é também utilizada como apoio para gerenciamento de questões de trânsito. Neste trabalho levamos em consideração o estado do projeto em Outubro de 2013 que contava com 54 câmeras. A disposição destes elementos podem ser vistos na Figura 4.7, onde os quadrados representam as câmeras do projeto de vídeo monitoramento e as circunferências representam os postes. Esta imagem foi retirada do simulador SINALGO.

Além dos elementos fixos apresentados anteriormente, existem também elementos variáveis presentes na simulação. São ditos elementos variáveis, aqueles elementos no qual suas posições se alteram antes de cada repetição da simulação. Um dos elementos variáveis inseridos na simulação são câmeras ditas auxiliares, que tem como objetivo serem utilizadas de forma oportunista. É importante lembrar que em um ambiente real, estas câmeras não precisariam ser instaladas para uso exclusivo da aplicação proposta, bastaria apenas seu acesso ser disponibilizado. Estas câmeras poderiam estar em edifícios ou residências, por exemplo. O outro elemento variável presente na simulação são os locais de onde partem as requisições realizadas pelos usuários do sistema.

Nas arquiteturas *Centralizada* e *Híbrida* existem um servidor cuja responsabilidade é tratar

Figura 4.7 – Disposição dos elementos no experimento realizado



Fonte: Autor, 2014

as requisições entre os usuários e as câmeras, além disto, no caso da arquitetura *Centralizada*, este servidor também realiza o processamento das imagens. Sua posição é indiferente para o resultado da simulação, visto que a comunicação entre os clientes e o servidor é realizada de forma direta, ou seja, em apenas um salto. É importante lembrar que, na arquitetura *Centralizada*, a comunicação entre o cliente e as câmeras e o processamento das imagens é realizado pelo servidor central. Enquanto que na arquitetura *Híbrida* o processamento das imagens ocorre nas câmeras, porém as requisições são tratadas pelo servidor central.

A arquitetura *Embutida* possui algumas diferenças das demais. O processamento das imagens é realizado diretamente na câmera, a comunicação ocorre diretamente entre os usuário e as câmeras, no entanto, enquanto a comunicação nas demais arquiteturas é realizada de forma direta (em apenas um salto), na abordagem *Embutida* esta comunicação é realizada em multi saltos. Como as câmeras encaminham as mensagens até seus destinatários, é preciso utilizar um algoritmo de roteamento para encaminhar as mensagens. Neste trabalho utilizamos uma abordagem desenvolvida por [Maia et al. \(2013\)](#). O algoritmo proposto por Maia et al. tem como ideia verificar se determinado nó se encontra dentro de um arco com determinado ângulo (escolhido *a priori*). Esta verificação é realizada levando em consideração os coeficientes lineares de duas retas formadas respectivamente pelos nó destino e o nó remetente e pelo nó candidato (nó vizi-

nho ao nó remetente) e o nó remetente. Além disso, a principal motivação do trabalho de Maia et al. é a utilização de redes complexas como modelo de propagação dos dados. A utilização do algoritmo de Maia et al. traz a vantagem de diminuir a quantidade de mensagens enviadas na rede, isto é feito criando *links* lógicos entre nós sensores com maior capacidade. Estes *links* servem como um atalho para encaminhamento da mensagem, outra funcionalidade é que estes *links* evitam desconexões na rede, por exemplo, caso falte energia em uma parte central da cidade, o encaminhamento da mensagem por esta área não será possível, porém por intermédio destes *links* as mensagens poderão ser encaminhadas para outras áreas mais próximas do destino, não sendo assim perdidas. Em nosso trabalho não é possível a utilização de algoritmos de roteamento com tabelas fixas, visto que são utilizadas câmeras de maneira oportunista. Um dos problemas da utilização de câmeras de forma oportunista é não conseguir prever por quanto tempo a utilização das câmeras estará disponível.

Para adequar as necessidades deste trabalho, o algoritmo proposto por Maia et al. precisou ser adaptado. A primeira adaptação necessária é que o algoritmo original não trata a perda de mensagens durante um encaminhamento, por exemplo, caso um nó queira encaminhar uma mensagem porém nenhum dos seus vizinhos esteja apto a receber a mensagem, a mesma será perdida. Em nossa aplicação, este ponto é algo que não deve ocorrer, visto que todas as requisições devem ser atendidas. Para solucionar este problema alteramos o algoritmo para o caso em que não encontre nenhum nó candidato a receber a mensagem, aumente o ângulo gradativamente (utilizamos uma taxa de θ , onde θ é o ângulo inicial) até o limite de 180° , onde basicamente a mensagem seria enviada para todos os vizinhos do nó remetente. A segunda adaptação necessária é que o algoritmo original utiliza a relação entre coeficientes lineares para determinar se um nó deve receber ou não a mensagem. O problema no uso de coeficiente linear está no fato de que se os nós estiverem na mesma horizontal (suas coordenadas possuem o mesmo valor no eixo x), como o coeficiente linear de uma reta é dado por $m = \frac{y_2 - y_1}{x_2 - x_1}$, caso os valores de x sejam iguais, o coeficiente linear não existirá, impossibilitando assim a utilização da relação. Para solucionar este problema, substituímos a necessidade da utilização dos coeficientes lineares pela relação do cálculo do ângulo entre dois vetores, dado por:

$$\phi = \arccos \left(\frac{u \cdot v}{\|u\| \|v\|} \right),$$

onde u é o vetor do nó candidato e o nó remetente e v é o vetor do nó destino e o nó remetente. O pseudocódigo do algoritmo adaptado pode ser visualizado no Algoritmo 3.

Para a simulação são levados em consideração os parâmetros apresentados na Tabela 4.3. O parâmetro *Raio* é relativo ao raio de alcance das câmeras utilizadas na simulação. Realizamos esta variação para avaliarmos o comportamento da arquitetura proposta com diferentes raios de comunicação. Como as câmeras utilizam wifi para comunicação, escolhemos três alcances distintos: 300 m, relativo ao padrão 802.11b ([Institute of Electrical and Electronics Engineers](#),

Algoritmo 3: Algoritmo de roteamento utilizado na abordagem *Embutida*

Entrada: Nó r , Nó d , Ângulo θ

```

1 inicio
2   Nó  $r$  calcula vetor  $u$  em direção ao nó  $d$ ;
3   para cada  $c$  vizinho de  $r$  fazer
4     Nó  $r$  calcula o vetor  $v$  em direção ao nó  $c$ ;
5      $\phi = \arccos\left(\frac{u \cdot v}{\|u\| \|v\|}\right)$ ;
6     se  $\phi < \theta$  então
7        $r$  envia mensagem para  $c$ ;
8     fim
9   fim
10  Se o nó  $r$  não conseguir enviar a mensagem para nenhum nó  $c$ , incremente  $\theta$  e tente novamente;
11 fin

```

2012), 600 m e 900 m, os dois últimos valores foram escolhidos apenas para avaliar o comportamento em casos que a tecnologia de comunicação utilizada possua alcances maiores. O parâmetro *Requisições* é relativo a quantidade de requisições realizadas simultaneamente, pode assumir os valores 100, 500 e 1000, tais valores foram escolhidos para simular possíveis situações na cidade, como exemplificado anteriormente. O parâmetro *Auxiliares* é relativo a quantidade de câmeras que não fazem parte do conjunto de câmeras do vídeo monitoramento do governo, podendo ser, por exemplo, câmeras de segurança de um edifício. Queremos observar com esta variação como a arquitetura *Embutida* se comporta de acordo com a quantidade de câmeras envolvidas no processo. Este parâmetro pode assumir os valores 100, 200 e 400. O parâmetro *falha* é relativo ao percentual de câmeras que irão falhar na rede, ou seja, irão desaparecer durante a simulação. Isto poderá ocorrer, por exemplo, quando uma câmera parar de disponibilizar suas imagens para uso na aplicação. Este parâmetro é utilizado apenas na abordagem *Embutida*. Isto ocorre pois nas abordagens *Centralizada* e *Híbrida* a comunicação é feita de forma direta entre o usuário e o servidor e entre o servidor e as câmeras, este parâmetro não irá influenciar no funcionamento das abordagens.

Tabela 4.3 – Valores dos parâmetros utilizados na simulação

Parâmetros	Valor 1	Valor 2	Valor 3
Raio	300 m	600 m	900 m
Requisições	100	500	1000
Câmeras Auxiliares	100	200	400
Falhas	5%	10%	25%

Fonte: Autor, 2014

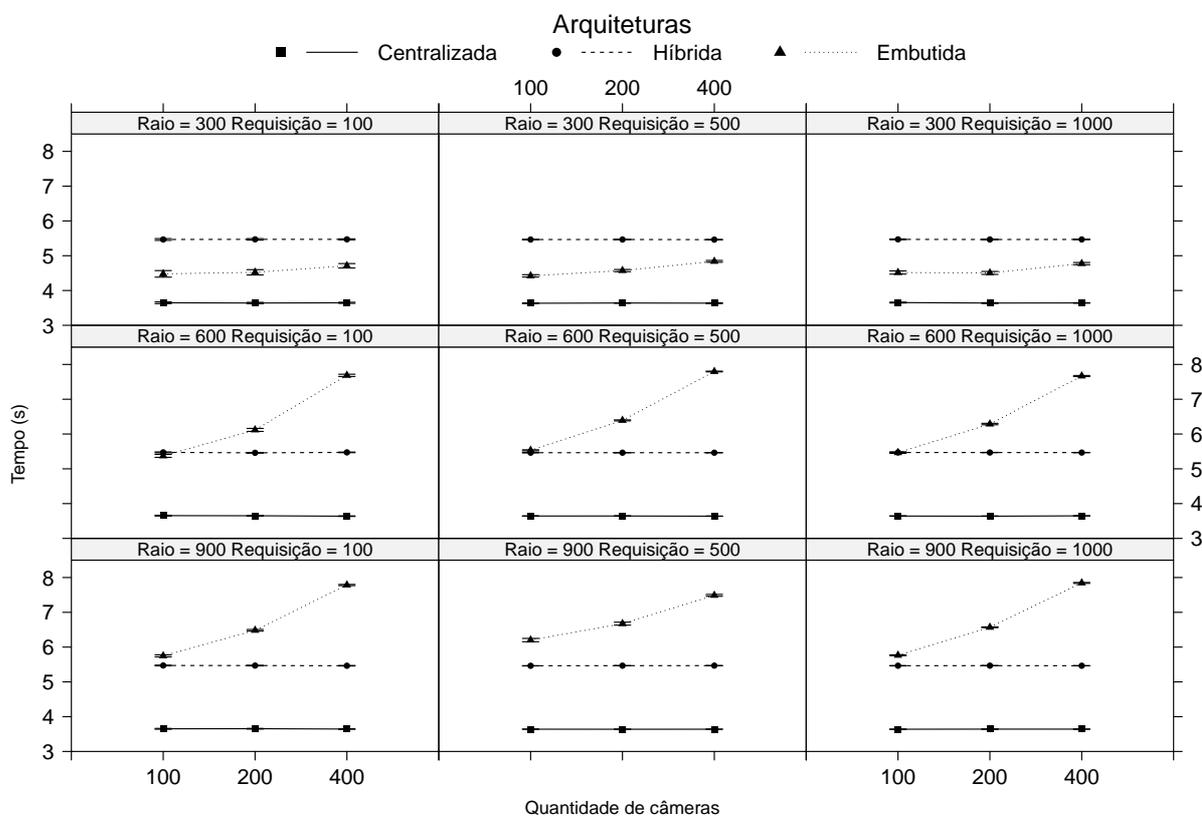
As simulações foram realizadas utilizando um computador com sistema operacional Fedora

20 64 bits processador Intel Xenon E5 com 32 cores de 2,6 Ghz e 128 GB de memória RAM.

Na Figura 4.8 é apresentado o resultado da simulação para as três abordagens utilizadas. No eixo x temos a quantidade de câmeras auxiliares utilizadas. No eixo y temos a média dos tempos (em segundos) das requisições e o intervalo de 95% de confiança destas médias. Como os intervalos de confiança ficaram muito parecidos com as médias não é possível visualiza-los facilmente no gráfico. Como dito anteriormente, neste cenário não é considerado o parâmetro de falhas, que é relativo a quantidade de nós auxiliares que falham durante a requisição, como as arquiteturas *Centralizada* e *Híbrida* não possuem nós auxiliares, não é possível analisar este parâmetro. É possível notar que a abordagem *Embutida*, no cenário com $Raio = 300$, consegue ter melhores tempos do que a abordagem *Híbrida*. Isto ocorre devido ao tamanho do raio de comunicação utilizado. Quanto menor o raio utilizado menor será o tempo consumido pela arquitetura *Embutida*, pois se o raio for grande, a câmera remetente terá mais câmeras vizinhas candidatas a receber a mensagem. A abordagem *Centralizada* possui os melhores tempos em todos os testes, isto ocorre pois o tempo de processamento das imagens no servidor é mais rápido do que o processamento nas placas utilizadas nas arquiteturas *Híbrida* e *Embutida*. Tanto a abordagem *Centralizada* quanto a abordagem *Híbrida* possuem comportamentos semelhantes, se mantendo praticamente constantes durante os testes. A abordagem *Embutida* possui um comportamento diferente das demais. Ao aumentar a quantidade de câmeras auxiliares, o tempo gasto aumenta. Isto ocorre pelo fato de quanto mais câmeras auxiliares no sistema, maior a quantidade de mensagens enviadas/recebidas na rede, aumentando assim o atraso na rede. Podemos notar que, em nenhum teste, a quantidade de requisições influenciou no resultado obtido, pois em todas as arquiteturas sempre será garantido o encaminhamento da requisição até o destino.

Ao aumentarmos o valor do parâmetro $Raio$ para 600, podemos observar a mudança nos resultados. Diferentemente dos cenários com $Raio = 300$, a abordagem *Embutida* se torna mais lenta, na maioria das combinações dos parâmetros, do que a abordagem *Híbrida*, novamente a abordagem *Centralizada* é a mais rápida. Quando a quantidade de câmeras auxiliares é pequena (nesta simulação, assumindo sua quantidade como 100) a abordagem *Embutida* obtêm seus melhores resultados, tendo tempos similares a abordagem *Híbrida*, porém quando a quantidade de câmeras auxiliares aumenta, o tempo gasto pela abordagem *Embutida* aumenta consideravelmente, enquanto as demais abordagens possuem um comportamento constante. Isto se justifica pelo fato das arquiteturas *Centralizada* e *Híbrida* não precisarem encaminhar mensagens dentro da rede, toda sua comunicação é realizada de forma direta, enquanto que na arquitetura *Embutida* as mensagens são encaminhadas de câmera em câmera. Além disto, ao aumentar o tamanho do raio de comunicação das câmeras, aumentará a quantidade de câmeras vizinhas candidatas a receber mensagens. O comportamento da abordagem *Embutida* está diretamente ligada a quantidade de câmeras auxiliares, quanto maior a quantidade destas câmeras, maior será o tempo gasto. Podemos deduzir que a quantidade de mensagens na rede será maior na ar-

Figura 4.8 – Simulação com todas arquiteturas



Fonte: Autor, 2014

quitetura *Embutida* devido ao aumento na quantidade de câmeras vizinhas candidatas em cada câmera remetente. Podemos notar que, em nenhum teste, a quantidade de requisições influenciou no resultado obtido, pois em todas as arquiteturas sempre será garantida o encaminhamento da requisição até o destino.

Com o parâmetro *Raio* = 900, o resultado se assemelha a simulação com *Raio* = 600. Porém, diferentemente da simulação anterior, a abordagem *Embutida* se torna sempre mais lenta do que a abordagem *Híbrida*, novamente a abordagem *Centralizada* é a mais rápida dentre as três. O aumento no tamanho do raio de comunicação das câmeras, aumentará a quantidade de câmeras vizinhas candidatas a receber mensagens, aumentando o tempo gasto para o envio destas mensagens. Quando a quantidade de câmeras auxiliares é pequena (nesta caso assumindo sua quantidade como 100) a abordagem *Embutida* obtêm seus melhores resultados, tendo tempos similares a abordagem *Híbrida*, porém quanto a quantidade de câmeras auxiliares aumenta, o tempo gasto pela abordagem *Embutida* aumenta consideravelmente, enquanto as demais abordagens possuem um comportamento constante. Este comportamento da arquitetura *Embutida* se justifica pois quanto menos câmeras auxiliares, menor será a quantidade de câmeras vizinhas e conseqüentemente menor será a quantidade de mensagens enviadas pela câmera remetente. O comportamento da abordagem *Embutida* está diretamente ligada a quantidade de câmeras auxi-

liares, quanto maior a quantidade destas câmeras, maior será o tempo gasto. Podemos deduzir que a quantidade de mensagens na rede será maior na arquitetura *Embutida* devido ao aumento na quantidade de câmeras vizinhas candidatas em cada câmera remetente. Podemos notar que, em nenhum teste, a quantidade de requisições influenciou no resultado obtido, pois em todas as arquiteturas sempre será garantida o encaminhamento da requisição até o destino.

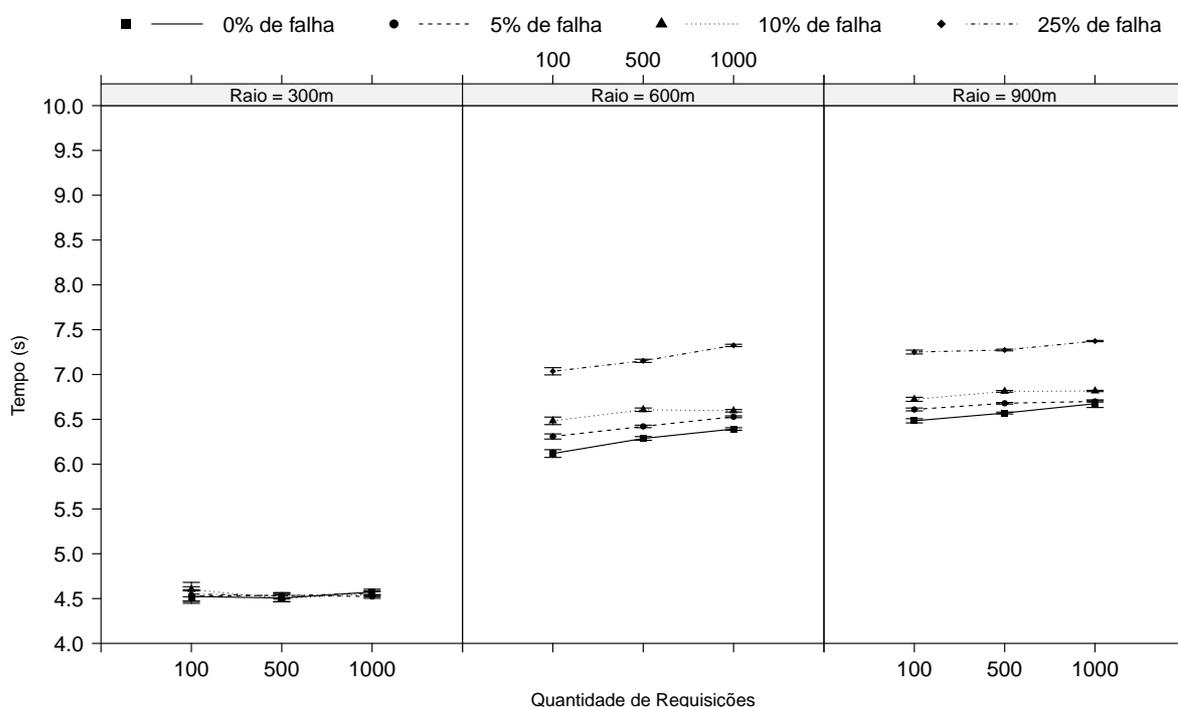
As principais conclusões destas simulações é o fato de a quantidade de requisições não influencia no resultado obtido, visto que variando a quantidade de requisições realizadas o comportamento das curvas são similares. Os principais fatores que influenciam o resultado são a quantidade de câmeras auxiliares e o aumento do raio de alcance das câmeras. Ambos fatores fazem com que a câmera remetente tenha mais vizinhos e conseqüentemente necessite enviar mais mensagens na rede, quanto maior o número de mensagens na rede, pior será o desempenho do sistema na arquitetura *Embutida*.

Além da análise comparativa entre a abordagem *Embutida* e as abordagens *Centralizada* e *Híbrida*, foram realizadas simulações para verificar o comportamento da abordagem proposta em maiores detalhes. Analisamos adicionalmente o parâmetro Falhas, em conjunto com os demais.

Na Figura 4.9 temos a simulação da abordagem *Embutida* fixando o valor do parâmetro quantidade de nós auxiliares em 200. No eixo x temos a quantidade de requisições realizadas simultaneamente. No eixo y temos a média dos tempos das requisições (em segundos) e seus respectivos intervalos de confiança. Cada reta no gráfico corresponde a um diferente percentual de falhas nas câmeras. Podemos verificar que com o $Raio = 300$, o percentual de falhas não interfere nos tempos das requisições, sendo praticamente iguais. Isto ocorre pois com uma menor quantidade de câmeras candidatas a receber as mensagens, menor será a chance de uma câmera que falhe receber a mensagem, conseqüentemente menor será a quantidade de tempo desperdiçado. Ao aumentar o tamanho do raio de comunicação das câmeras, vemos que quanto maior o percentual de falhas, maior será o tempo utilizado nas requisições. Isto ocorre pois quanto maior o raio de comunicação, maior será a área na qual a câmera remetente terá acesso e conseqüentemente maior será a probabilidade dela possuir mais vizinhos. Com uma maior quantidade de vizinhos, maior será a quantidade de mensagens enviadas. Ao aumentar o percentual de falhas nas câmeras, as câmeras vizinhas receberão as mensagens, porém tais mensagens poderão ser desperdiçadas, caso a câmera falhe. Como a quantidade de câmeras auxiliares foi fixada neste cenário da simulação, pode-se concluir que o fator influenciador no comportamento da simulação é o tamanho do raio de comunicação.

Na Figura 4.10 temos o cenário de simulação com o percentual de falhas fixado em 5%. No eixo x temos a quantidade de requisições realizadas simultaneamente. No eixo y temos a média dos tempos das requisições (em segundos) e seus respectivos intervalos de confiança. Cada reta no gráfico corresponde a uma diferente quantidade de câmeras auxiliares. Ao fixar o percentual de falhas, não temos o problema relativo ao cenário anterior de uma configuração possuir mais

Figura 4.9 – Simulação com quantidade de nós auxiliares = 200

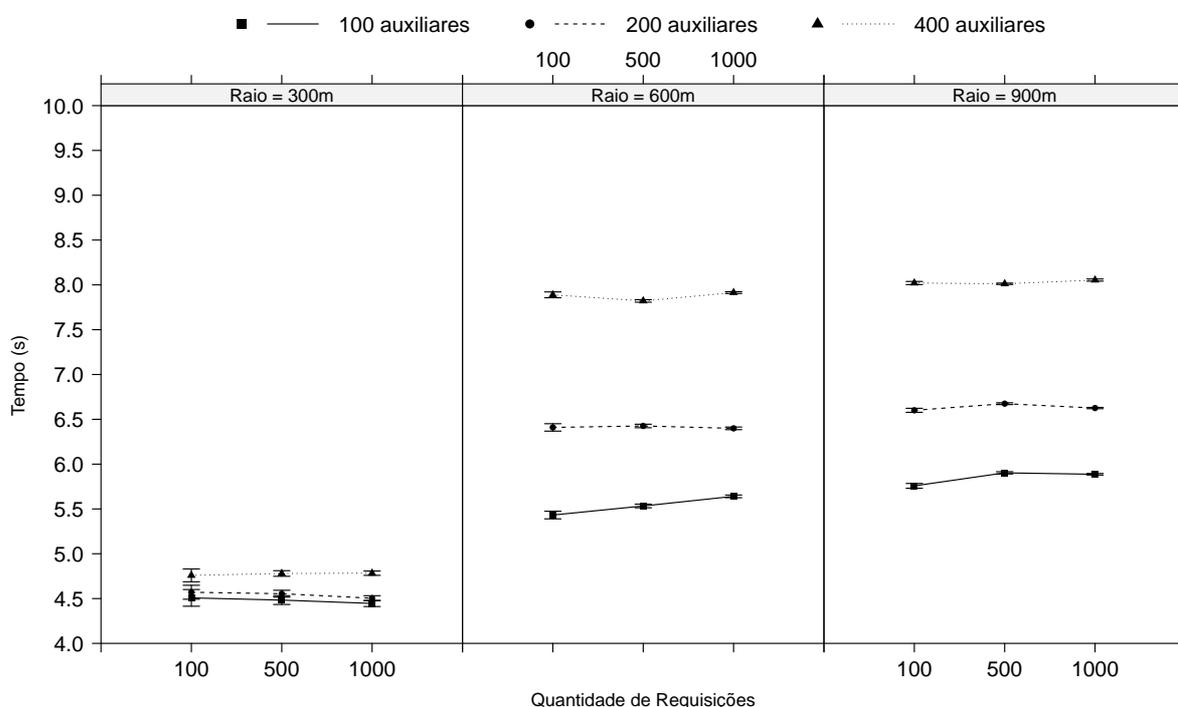


Fonte: Autor, 2014

desperdício de mensagens do que as outras, pois a probabilidade das câmeras falharem será a mesma em todos os cenários. Podemos notar que em todas as combinações dos parâmetros, quanto menor for a quantidade de câmeras auxiliares utilizadas, menor será o tempo médio gasto pelas requisições dos usuários. Além disto, é também notável que ao aumentar o raio de comunicação das câmeras, o tempo médio de requisição também aumenta. Estes fatos se justificam pois, quanto maior a quantidade de câmeras auxiliares, maior será a probabilidade de uma câmera possuir uma maior quantidade de vizinhos, e quanto maior a quantidade de câmeras vizinhas, maior será o tempo gasto para o envio de mensagens para cada vizinho.

Na Figura 4.11 temos o cenário de simulação com o raio de comunicação fixado em 600 m. No eixo x temos a quantidade de requisições realizadas simultaneamente. No eixo y temos a média dos tempos das requisições (em segundos) e seus respectivos intervalos de confiança. Cada reta no gráfico corresponde a um diferente percentual de falhas nas câmeras. Ao fixarmos o raio de comunicação não teremos o problema do aumento na quantidade de vizinhos devido ao alcance do envio das mensagens. Podemos notar que em todas as combinações dos parâmetros neste cenário, quanto menor o percentual de falhas, menor o tempo médio das requisições. Isto ocorre pois quanto maior o percentual de falhas, maior a probabilidade de desperdiçar

Figura 4.10 – Simulação com percentual de falhas = 5

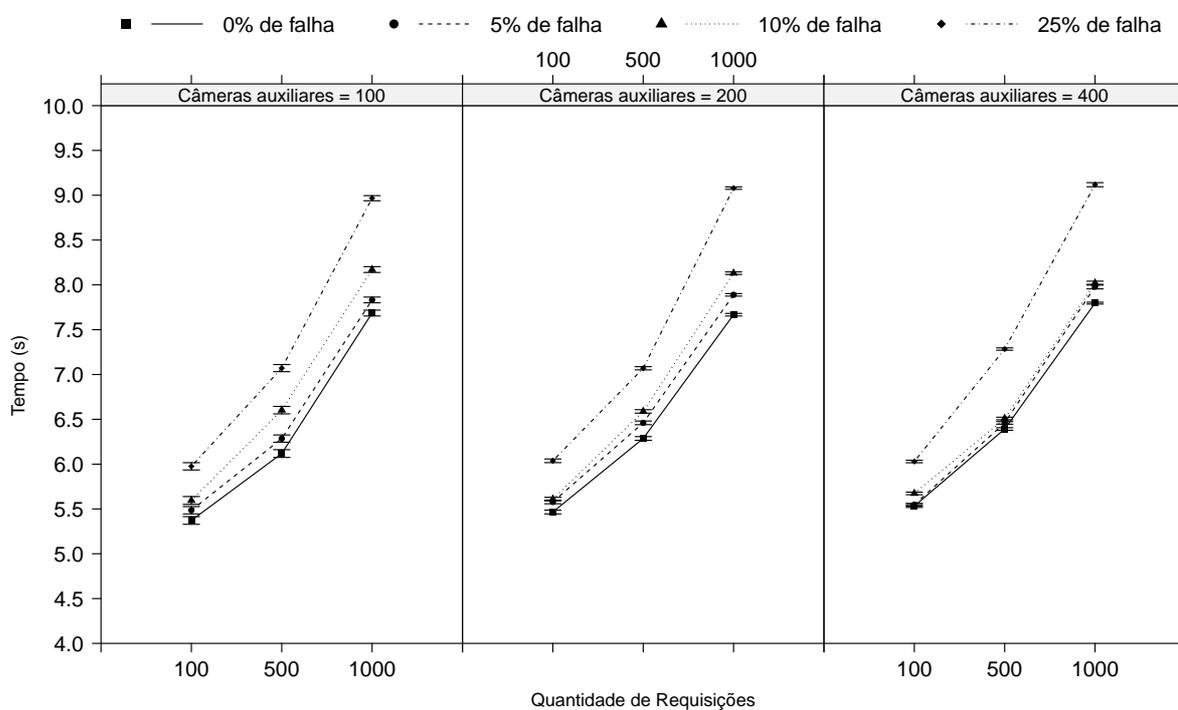


Fonte: Autor, 2014

tempo, isto é, enviar a mensagem para uma câmera que irá falhar. Além disto, podemos notar que a quantidade de requisições influencia na variação do tempo médio da requisição, ao aumentarmos a quantidade de requisições, o tempo médio também aumenta, porém ao variar apenas a quantidade de câmeras auxiliares, o tempo médio das requisições se torna praticamente constante. Isto ocorre pois quanto mais requisições realizadas simultaneamente, maior será a quantidade de mensagens na rede e conseqüentemente maior será o tempo gasto no envio das mensagens.

Na Figura 4.12 temos o cenário de simulação com quantidade de requisições simultâneas fixada em 500. No eixo *x* temos a quantidade de câmeras auxiliares. No eixo *y* temos a média dos tempos das requisições (em segundos) e seus respectivos intervalos de confiança. Cada reta no gráfico corresponde a um diferente percentual de falhas nas câmeras. Analisando este cenário por partes temos que, quando *Raio* = 300 a variação da quantidade de câmeras auxiliares em relação ao percentual de falhas nas câmeras é mínima, porém ao aumentarmos o tamanho do raio de comunicação, vemos que o tempo médio das requisições aumenta, por exemplo, na configuração (*Raio* = 300 *Auxiliares* = 400 *Falha* = 25%), o tempo médio da requisição é em torno de 5 s, quando temos a mesma configuração porém com *Raio* = 600, o valor do

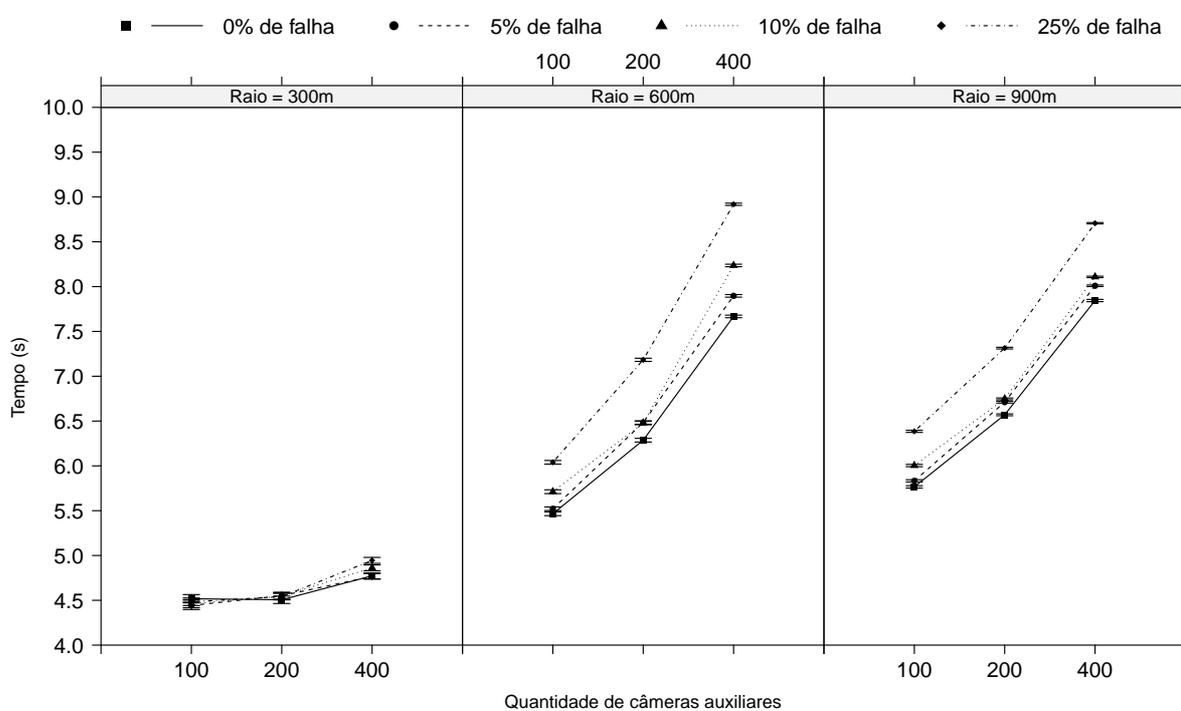
Figura 4.11 – Simulação com raio de comunicação = 600 m



Fonte: Autor, 2014

tempo médio aumenta para aproximadamente 9 s. Vemos que ao aumentar o tamanho do raio de comunicação, aumentará a quantidade de câmeras vizinhas, conseqüentemente aumentando a quantidade de mensagens enviadas na rede. Isto demonstra que o Raio é o fator com maior influência na variação nos resultados obtidos.

Figura 4.12 – Simulação com quantidade de requisições = 500



Fonte: Autor, 2014

5 Considerações finais

Este trabalho apresentou um sistema oportunista para detecção de vagas de estacionamento utilizando placas acopladas à câmeras de segurança. Dentre as principais contribuições deste trabalho tivemos:

- Avaliação de três técnicas de processamento de imagens para detecção de veículos;
- Proposta de uma arquitetura distribuída e oportunista utilizando câmeras para detecção de vagas de estacionamento;
- Avaliação comparativa entre arquiteturas existentes;
- Adaptação de um algoritmo de roteamento de redes de sensores sem fio para a utilização no problema apresentado com a arquitetura *Embutida*.

Inicialmente foram avaliadas três técnicas de processamento de imagens com o objetivo de realizarem a detecção da existência ou não de vagas de estacionamento nas imagens. Como resultado das avaliações destas técnicas, a técnica *Dilatação após Detecção de Bordas* obteve aproximadamente 100% de eficácia, sendo assim escolhida para a continuidade do trabalho.

Em seguida, foi realizado um experimento para verificar o comportamento da abordagem *Embutida* em comparação com outras duas abordagens existentes (*Centralizada* e *Híbrida*). Estes experimentos foram realizados com sete câmeras recebendo uma requisição por vez e utilizando a técnica de processamento de imagens *Dilatação após Detecção de Bordas*. Os resultados obtidos demonstraram que o tempo de execução utilizando a placa de processamento embutida não é afetado caso seja utilizado um servidor para intermediar o recebimento/repasso das requisições.

Adicionalmente, foi realizada uma simulação com diversos cenários levando em consideração quatro variáveis (raio de comunicação, quantidade de câmeras auxiliares, quantidade de requisições simultâneas e percentual de falhas nas câmeras). A realização destas simulações tiveram como objetivo verificar e avaliar o comportamento da abordagem em cenários onde, por intermédio de experimentação, seria uma tarefa árdua. Com o resultado da simulação pode-se verificar que os fatores com maior influência na variação dos tempos médios das requisições são o tamanho do raio de comunicação e a quantidade de câmeras auxiliares.

Tivemos dois artigos completo aceitos em anais de eventos nacionais. As publicações foram:

- Lima, D. H. S., Aquino, A. L. L., Almeida, E. S., Sistema para detecção oportunista de vagas de estacionamento via câmeras de segurança, no V Simpósio Brasileiro de Computação Ubíqua e Pervasiva, Maceió - AL, 2013.
- Oliveira, S., Teixeira, F. A., Macedo, D. F., Aquino, A. L. L., Lima, D. H. S., Silva, C., Silva, R., Sistema de coleta e disseminação de dados de trânsito, no XXXI Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos, Brasília - DF, 2013.

Como trabalhos futuros temos:

- Utilização de mais características presentes nas imagens para obter melhores resultados nas técnicas utilizadas
- Desenvolvimento e avaliação de novas técnicas para detecção de vagas de estacionamento;
- Desenvolvimento de técnicas para testes em imagens de baixa resolução;
- Desenvolvimento de um algoritmo de roteamento voltado para a aplicação com câmeras oportunista;
- Executar experimento com maior quantidade de câmeras;
- Adaptar a solução para uso em VANETs;
- Realizar estratégia para balanceamento de carga com relação as vagas de estacionamento disponíveis;
- Desenvolvimento de uma solução pro-ativa;
- Utilização de séries temporais para recomendação das melhores horas para se encontrar vagas em determinados lugares.

REFERÊNCIAS

- Acharya, T. & Ray, A. K. (2005), *Image Processing: Principles and Applications*, Wiley-Interscience.
- Alhammad, A., Siewe, F. & Al-Bayatti, A. H. (2012), An infostation-based context-aware on-street parking system, *in* 'International Conference on Computer Systems and Industrial Informatics'.
- Baronti, P., Pillai, P., Chook, V. W. C., Chessa, S., Gotta, A. & Hu, Y. F. (2007), 'Wireless sensor networks: A survey on the state of the art and the 802.15.4 and ZigBee standards', *Computer communications* **30**(7), 1655–1695.
- Caliskan, M., Graupner, D. & Mauve, M. (2006), Decentralized discovery of free parking places, *in* '3rd International Workshop on Vehicular Ad Hoc Networks'.
- Cidade de São Paulo (2014), 'Companhia de engenharia de tráfego'. URL <http://www.cetesp.com.br>, acessado em maio de 2014.
- Distributed Computing Group (2008), 'Sinalgo - Simulator for Network Algorithms'. URL <http://disco.ethz.ch/projects/sinalgo/>, acessado em maio de 2014.
- Dougherty, E. R. (1992a), *An introduction to morphological image processing*, Tutorial texts in optical engineering, SPIE Optical Engineering Press.
- Dougherty, E. R. (1992b), *Mathematical Morphology in Image Processing*, Optical Science and Engineering, Taylor & Francis.
- Dougherty, E. R. (1994), *Digital Image Processing Methods*, Optical Science and Engineering, Taylor & Francis.
- Evenepoel, S., Van Ooteghem, J., Verbrugge, S., Colle, D. & Pickavet, M. (2014), 'On-street smart parking networks at a fraction of their cost: performance analysis of a sampling approach', *Transactions on Emerging Telecommunications Technologies* **25**(1), 136–149.
- Geng, Y. & Cassandras, C. G. (2012), 'A new "smart parking" system infrastructure and implementation', *Procedia - Social and Behavioral Sciences* **54**, 1278–1287.
- Gonzalez, R. & Woods, R. (1992), *Fundamentals of Digital Image Processing*, Addison-Wesley Publishing Company.
- Hart, P. E., Nilsson, N. J. & Raphael, B. (1968), 'A formal basis for the heuristic determination of minimum cost paths', *IEEE Transactions on Systems Science and Cybernetics* **4**(2), 100–107.

- Hartenstein, H. & Laberteaux, K. P. (2008), 'A tutorial survey on vehicular ad hoc networks', *IEEE Communications Magazine* **46**(6), 164–171.
- Idris, M. Y. I., Tamil, E. M., Noor, N. M., Razak, Z. & Fong, K. W. (2009), 'Parking guidance system utilizing wireless sensor network and ultrasonic sensor', *Information Technology Journal* **8**(2), 138–146.
- Institute of Electrical and Electronics Engineers (2012), 'IEEE standard for information technology – telecommunications and information exchange between systems local and metropolitan area networks–specific requirements part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications', *IEEE Std 802.11-2012 (Revision of IEEE Std 802.11-2007)* pp. 1–2793.
- Jermurawong, J., Ahsan, M., Haidar, A., Dong, H. & Mavridis, N. (2012), Car parking vacancy detection and its application in 24-hour statistical analysis, in 'International Conference on Frontiers of Information Technology'.
- Lixia, W. & Dalin, J. (2012), A method of parking space detection based on image segmentation and LBP, in 'International Conference on Multimedia Information Networking and Security'.
- Maia, G., Aquino, A. L. L., Guidoni, D. L. & Loureiro, A. A. F. (2013), 'A multicast reprogramming protocol for wireless sensor networks based on small world concepts', *Journal of Parallel and Distributed Computing* **73**(9), 1277–1291.
- Marques Filho, O. & Vieira Neto, H. (1999), *Processamento Digital de Imagens*, Brasport.
- Mathur, S., Jin, T., Kasturirangan, N., Chandrasekaran, J., Xue, W., Gruteser, M. & Trappe, W. (2010), ParkNet: Drive-by sensing of road-side parking statistics, in '8th International Conference on Mobile Systems, Applications, and Services'.
- McGill, R., Tukey, J. W. & Larsen, W. A. (1978), 'Variations of box plots', *The American Statistician* **32**(1), 12–16.
- Metz, C. E. (2006), 'Receiver operating characteristic analysis: a tool for the quantitative evaluation of observer performance and imaging systems', *Journal of American College of Radiology* **3**(6), 413–422.
- Naphade, M., Banavar, G., Harrison, C., Paraszczak, J. & Morris, R. (2011), 'Smarter cities and their innovation challenges', *Computer* **44**(6), 32–39.
- Nawaz, S., Efstratiou, C. & Mascolo, C. (2013), Parksense: A smartphone based sensing system for on-street parking, in '19th Annual International Conference on Mobile Computing & Networking'.

- Otsu, N. (1979), 'A threshold selection method from gray-level histograms', *IEEE Transactions on Systems, Man and Cybernetics* **9**(1), 62–66.
- ParkHelp (2014), 'ParkHelp'. URL <http://www.parkhelp.com/>, acessado em maio de 2014.
- Pelusi, L., Passarella, A. & Conti, M. (2006), 'Opportunistic networking: data forwarding in disconnected mobile ad hoc networks', *IEEE Communications Magazine* **44**(11), 134–141.
- Polycarpou, E., Lambrinos, L. & Protopapadakis, E. (2013), Smart parking solutions for urban areas, in 'IEEE 14th International Symposium and Workshops on a World of Wireless, Mobile and Multimedia Networks'.
- Prewitt, J. M. S. (1970), 'Object enhancement and extraction', *Picture Processing and Psychopictorics* **75**, 75–149.
- Prince, S. J. D. (2012), *Computer Vision: Models, Learning, and Inference*, Cambridge University Press.
- Ramos, H. S., Boukerche, A., Pazzi, R., Frery, A. & Loureiro, A. (2012), 'Cooperative target tracking in vehicular sensor networks', *IEEE Wireless Communications* **19**(5), 66–73.
- Rodrigues, J. M. (2013), Evolução da frota de automóveis e motos no brasil, Technical report, Instituto Nacional de Ciência e Tecnologia.
- Scott, D. W. (1992), *Multivariate Density Estimation. Theory, Practice and Visualization*, Wiley.
- Sezgin, M. & Sankur, B. (2004), 'Survey over image thresholding techniques and quantitative performance evaluation', *Journal of Electronic Imaging* **13**(1), 146–168.
- Shoup, D. C. (2006), 'Cruising for parking', *Transport Policy* **13**, 479–486.
- Spackman, K. A. (1989), Signal detection theory: Valuable tools for evaluating inductive learning, in '6th International Workshop on Machine Learning'.
- Stehman, S. V. (1997), 'Selecting and interpreting measures of thematic classification accuracy', *Remote Sensing of Environment* **62**(1), 77–89.
- Streetline Inc. (2014), 'Streetline'. URL <http://www.streetline.com/>, acessado em maio de 2014.
- Tanenbaum, A. S. & Steen, M. V. (2006), *Sistemas distribuídos: Princípios e paradigmas*, Pearson.

- TKH Group company (2014), 'Parkassist sensing success'. URL <http://www.parkassist.com/>, acessado em maio de 2014.
- Urbiotica (2014), 'Mov'SMART'. URL <http://www.urbiotica.com/>, acessado em maio de 2014.
- VehicleSense (2014), 'VehicleSense'. URL <http://www.vehiclesense.com/>, acessado em maio de 2014.
- Waterson, B. J., Hounsell, N. B. & Chatterjee, K. (2001), 'Quantifying the potential savings in travel time resulting from parking guidance systems - a simulation case study', *Journal of the Operational Research Society* **52**(10), 1067–1077.
- Worldsensing SL (2014), 'Fastprk'. URL <http://www.fastprk.com/>, acessado em maio de 2014.
- Wu, Q., Huang, C., Yu Wang, S., Chiu, W.-C. & Chen, T. (2007), Robust parking space detection considering inter-space correlation, *in* 'IEEE International Conference on Multimedia and Expo'.
- Xu, B., Wolfson, O., Yang, J., Stenneth, L., Yu, P. & Nelson, P. (2013), Real-time street parking availability estimation, *in* 'IEEE 14th International Conference on Mobile Data Management'.
- Yusnita, R., Fariza, N. & Norazwinawati, B. (2012), 'Intelligent parking space detection system based on image processing', *International Journal of Innovation, Management and Technology* **3**(3), 232–235.
- Zhang, Z., Li, X., Yuan, H. & Yu, F. (2013), 'A street parking system using wireless sensor networks', *International Journal of Distributed Sensor Networks* **2013**, 1–10.
- Zhao, H., Lu, L., Song, C. & Wu, Y. (2012), 'IPARK: Location-aware-based intelligent parking guidance over infrastructureless VANETs', *International Journal of Distributed Sensor Networks* **2012**, 1–12.
- Ziou, D. & Tabbone, S. (1998), 'Edge detection techniques - an overview', *International Journal of Pattern Recognition and Image Analysis* **8**(4), 537–559.