

Sistema para Detecção Oportunista de Vagas de Estacionamento via Câmeras de Segurança

David H. S. Lima^{1,2}, Andre L. L. Aquino¹, Eliana S. Almeida¹

¹ Instituto de Computação – Universidade Federal de Alagoas (UFAL)
Maceió – Alagoas – Brasil

{dhs.lima, alla.lins, eliana.almeida}@gmail.com

Abstract. *The present work proposes a system that opportunistically detects free parking spaces by using surveillance cameras that are becoming widely available in public places. We applied different image processing techniques to detect free spots under three different scenarios. We obtained promising results where the success detection rate reaches up to 84% and the request processing time lies in range 0.21s to 24.26s.*

Resumo. *O trabalho desenvolvido propõe um sistema para detecção oportunista de vagas de estacionamento utilizando câmeras de segurança. Nós aplicamos diferentes técnicas de processamento de imagem para detectar vagas de estacionamento livres em três diferentes cenários. Obtemos resultados promissores onde a taxa de sucesso de detecção alcança até 84% e o tempo de processamento da requisição fica no intervalo de 0.21s até 24.26s.*

1. Introdução

Devido ao aumento da frota de veículos particulares, a busca por vagas de estacionamento consome um bom tempo dos condutores. Este problema ocorre em geral em áreas urbanas, onde o número de veículos é maior do que a quantidade de vagas de estacionamento disponíveis. Para a comodidade dos condutores, os estacionamentos deveriam ser responsáveis por informar a disponibilidade de vagas. No entanto, tal sistema, se realizado de maneira manual, precisaria de uma grande quantidade de recursos humanos, portanto a utilização de sistemas sem a supervisão humana para detecção de vagas tem sido implantados em diversos estacionamentos.

É importante que tais sistemas não necessitem realizar grandes alterações nos ambientes na qual eles sejam instalados, por este motivo surge a necessidade da utilização de sistemas oportunistas, que são sistemas que aproveitam recursos disponibilizados por outros sistemas. No trabalho aqui proposto, o sistema aproveitará imagens provenientes de câmeras de segurança previamente instaladas, sem ter a necessidade de realizar a instalação de novas câmeras.

O objetivo deste trabalho é a proposição de uma ferramenta para sugestão de vagas de estacionamentos que utiliza câmeras oportunistas. Avaliamos a aplicação de três técnicas de processamento de imagem para a detecção de vagas de estacionamentos em imagens obtidas das câmeras de segurança. As técnicas utilizadas são formadas por sucessivas aplicações de morfologias, a primeira técnica, chamada de SB, é formada pela aplicação da binarização ao resultado da subtração da imagem do estacionamento no momento da requisição e da imagem do estacionamento vazio. A segunda técnica é a ESB,

que é a técnica SB com a realização da equalização antes de realizar a subtração das imagens. A terceira técnica é a BBD, que consiste da aplicação de um método de detecção de bordas, da binarização e da dilatação na imagem. Para avaliar a ferramenta desenvolvida consideramos três cenários distintos: o primeiro, consiste de uma arquitetura de comunicação cliente servidor com o processamento sendo realizado no servidor (CS); o segundo cenário é uma arquitetura de comunicação cliente servidor com o processamento sendo realizado no cliente, ou seja, nas câmeras (CSP); o terceiro cenário consiste em uma arquitetura *ad-hoc* com processamento sendo realizado nas câmeras (CP).

Os resultados revelaram que a técnica que obteve uma melhor taxa de acerto foi a ESB, com uma taxa no melhor cenário de cerca de 84%, enquanto que a pior técnica foi a BBD, tendo no pior caso apenas cerca de 16% de acerto. O melhor cenário na avaliação foi o cenário com a arquitetura de comunicação cliente servidor com o processamento sendo realizado no servidor, que foi em alguns casos cerca de cinco vezes mais rápido do que os outros cenários.

O artigo está dividido da seguinte forma: Seção 2 apresenta trabalhos correlatos. Seção 3 detalha os cenários utilizados nos experimentos. Seção 4 detalha a aplicação desenvolvida neste trabalho. Seção 5 apresenta os experimentos realizados e os resultados obtidos. Por fim, a seção 6 mostra as considerações finais do trabalho e futuras direções.

2. Trabalhos Correlatos

Existem trabalhos que apresentam soluções para detecção de vagas em estacionamento utilizando diversas técnicas, dentre as quais podemos citar a utilização de sensores [Park et al. 2008, Yang et al. 2012] e através do processamento de imagens [Huang et al. 2012].

O trabalho de [Yusnita et al. 2012] apresenta uma solução para o problema de detecção de vagas utilizando processamento de imagens. Foi desenvolvido um sistema que possui seis etapas: inicialização do sistema, aquisição de imagem, segmentação de imagem, transformação da imagem para tom de cinza, binarização e a detecção das bordas dos objetos. O objetivo do sistema é realizar a detecção de imagem que detecta a quantidade de vagas disponíveis. Para criação dos cenários foram utilizados carros de brinquedo e foi realizado em um ambiente interno, enquanto que o nosso trabalho utiliza um experimento em ambiente real que inclui diversas variáveis ao problema.

O trabalho de [Lixia and Dalin 2012] tem como objetivo verificar a disponibilidade de vagas de estacionamento tratando-as como células. Para verificar a disponibilidade, os autores utilizam da busca por características em comum aos veículos, com isto eles localizam nas células por segmentos de retas que possam ser detectados após a segmentação da imagem. Esta proposta tem como principais problemas a falha quando existem objetos encobrendo parte da vaga e quando a coloração dos veículos se assemelha com o da estrada.

O trabalho de [Jermurawong et al. 2012] utiliza redes neurais para determinar o estado de ocupação de vagas de estacionamentos baseados na extração de características visuais dos estacionamentos. As características extraídas são relacionadas a luminosidade, *pixels*, borda, cores e tempo. Para alcançar a solução, os autores tiveram que resolver três problemas: a mudança da intensidade da luz, a limitação da luz durante o

amanhecer e o anoitecer e os diferentes padrões de luminosidade durante a noite. O trabalho avalia diferentes cenários para esse problema e apresenta um experimento realizado durante um dia completo. Para detecção das vagas os autores utilizam redes neurais para determinar se a vaga está ocupada ou não, enquanto que a nossa abordagem utiliza puramente processamento de imagens.

Os trabalhos apresentados anteriormente são propostas de soluções para detecção de vagas em estacionamentos utilizando processamento de imagens. Em todos os trabalhos são apresentadas técnicas que realizam apenas a decisão da existência ou não da vaga de estacionamento. No trabalho proposto visamos não apenas a detecção da existência de vaga como também a indicação de qual região está vaga detectada.

3. Cenários Utilizados na Avaliação do Sistema

O objetivo deste trabalho é a proposição de uma ferramenta para sugestão de vagas de estacionamentos que utiliza câmeras oportunistas.

As câmeras utilizadas no projeto foram as DCS-2130¹ da marca DLINK (ver figura 1(a)), em alternativa poderiam ser utilizadas quaisquer câmeras *ip* para a arquitetura CS, enquanto que para as arquiteturas CSP e CP qualquer câmera poderia ser utilizada, já que a comunicação é realizada junto a placa. Estas câmeras são apropriadas para ambientes internos, possuem a capacidade de captar imagens em HD, além de possuir conectividade via *wireless*, via cabo ou via *bluetooth*. Elas não possuem controlador de luminosidade.

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 1(b) podemos visualizar a placa Panda.

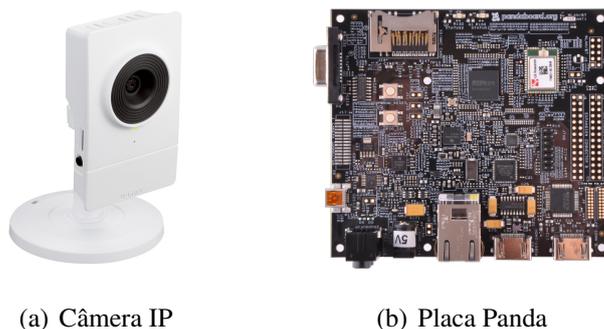


Figura 1. Dispositivos utilizados no sistema

Para realizarmos testes, foram escolhidas três cenários distintos, CS, CSP e CP. O cenário CS consiste na utilização de uma arquitetura cliente-servidor, na qual temos o servidor como sendo uma máquina comum. A requisição do cliente ao servidor é realizada diretamente. Ao receber a requisição, o próprio servidor realizará todo o processamento na imagem e enviará a resposta para o cliente.

¹<http://www.dlink.com.br/produtos-detalhes/items/DCS-2130.html>

²<http://pandaboard.org/>

O cenário CSP consiste da utilização de um servidor para gerenciar as conexões com o cliente e para enviar as requisições para as câmeras e da utilização de placas de processamento acopladas as câmeras para realizar o processamento das imagens. Neste cenário o servidor ao receber uma requisição de um cliente, irá encaminhá-la para a respectiva câmera que processará a imagem utilizando a placa externa e enviará a resposta do processamento para o servidor.

O cenário CP é um cenário *ad-hoc*, onde a comunicação é realizada diretamente entre os clientes e as câmeras. Neste cenário as câmeras também possuem placas externas que realizam o processamento. Ao receber uma requisição, a câmera irá processá-la e enviará a resposta diretamente para o cliente.

4. Aplicação para Detecção de Vagas de Estacionamento

O sistema desenvolvido neste trabalho consiste de uma aplicação *Android* onde o usuário requisita ao sistema a identificação da existência ou não de vagas de estacionamento nos locais escolhidos. O fluxo de execução do aplicativo pode ser visto na figura 2. Na tela inicial o usuário poderá escolher entre os três cenários para a execução do sistema. A tela seguinte é a que o usuário selecionará o estacionamento ou região na qual ele pretende estacionar o carro. Na tela seguinte será apresentada a resposta do sistema, onde o nome em verde significa que existem vagas e o nome em vermelho significa que não existem.



Figura 2. Execução da Aplicação

Na figura 3 é apresentado o fluxograma de execução do sistema. Ao acessar o sistema o usuário deverá *selecionar o estacionamento ou região* na qual ele deseja verificar a existência de vagas. O dispositivo *Android* criará *uma lista o estacionamento ou região selecionada* pelo usuário e *enviará esta lista para o módulo de coleta de vagas*. O módulo de coleta de vagas pode ser o servidor ou a placa *Panda*, dependendo do cenário escolhido. Ao receber a lista com os estacionamentos selecionados, o *módulo de coleta de vagas* irá *selecionar individualmente os estacionamentos* escolhidos pelo usuário. O módulo de coleta de vagas então *requisita as imagens para a câmera* do estacionamento selecionado, a *câmera obtém a imagem e a envia para o módulo de coleta de vagas*. Ao receber as imagens das câmeras, o módulo de coleta de vagas irá realizar o *processamento nas imagens* para detectar a existência de vagas. Após verificar todos os estacionamentos o *módulo de coleta de vagas* enviará a resposta para o usuário, que irá visualizá-la em seu dispositivo.

Na etapa do *processamento das imagens* provenientes das câmeras, o sistema apresenta três técnicas distintas para resolução do processamento da detecção de vagas. As

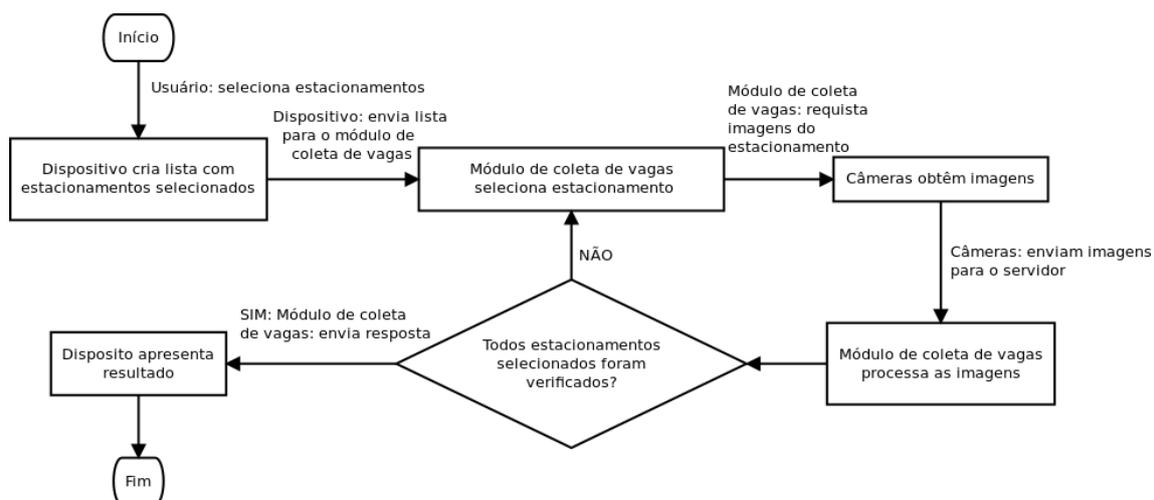


Figura 3. Fluxograma do Sistema

técnicas apresentadas possuem semelhanças em sua composição, pois são formadas por sucessivas aplicações de transformações nas imagens. As técnicas são formadas por conversão para cinza, binarização, equalização, detecção de bordas e dilatação. A conversão de cinza é uma técnica que converte a imagem do sistema de cor RGB para tom de cinza. A operação de binarização é uma técnica que tem como objetivo a conversão de uma imagem em tom de cinza para uma imagem com apenas dois valores possíveis para um *pixel*, em geral 0 ou 1. Em termos gerais a binarização funciona escolhendo um limiar na qual os valores que tiverem acima deste limiar recebem o valor 0 ou 1 e os que tiverem abaixo recebem o outro valor possível. Existem diversas técnicas de binarização, a utilizada neste trabalho foi a proposta por Otsu [Otsu 1979]. A equalização da imagem através do histograma tem como objetivo melhorar o contraste da imagem, no caso foi aplicado a técnica de equalização global que consiste em equalizar a imagem levando em consideração todos os tons de cinza presentes na imagem. Detecção de bordas tem como objetivo determinar o contorno dos objetos contidos na imagem, utilizamos o operador de Prewitt [Prewitt 1970] para realizar a detecção de borda nas imagens. A dilatação é uma técnica que tem como objetivo realizar o fechamento de possíveis buracos na imagem.

Na figura 4 podemos ver as técnicas utilizadas para detecção de vagas. O cenário inicial utiliza apenas duas etapas para o processamento das imagens, a primeira etapa é a subtração da imagem obtida pela requisição do servidor para as câmeras pela imagem do estacionamento vazio (imagem base), a próxima etapa é a binarização da imagem resultante da subtração. O segundo cenário consiste em realizar uma equalização nas imagens, as demais operações são iguais ao primeiro cenário. O terceiro cenário é formado por três etapas e tem a vantagem de não necessitar da imagem base, algo que os cenários anteriores necessitavam. A primeira etapa é a utilização da detecção de bordas, o segundo passo é realizar a binarização da imagem obtida do passo anterior, com a imagem binarizada, é aplicada a técnica de dilatação.

Antes de aplicar as técnicas descritas anteriormente nas imagens, é preciso realizar um pré-processamento nas imagens para demarcar as áreas onde existem possíveis vagas de estacionamento, isto é feito para que no momento da busca pelas vagas não seja necessário realizar o processamento em toda a imagem.

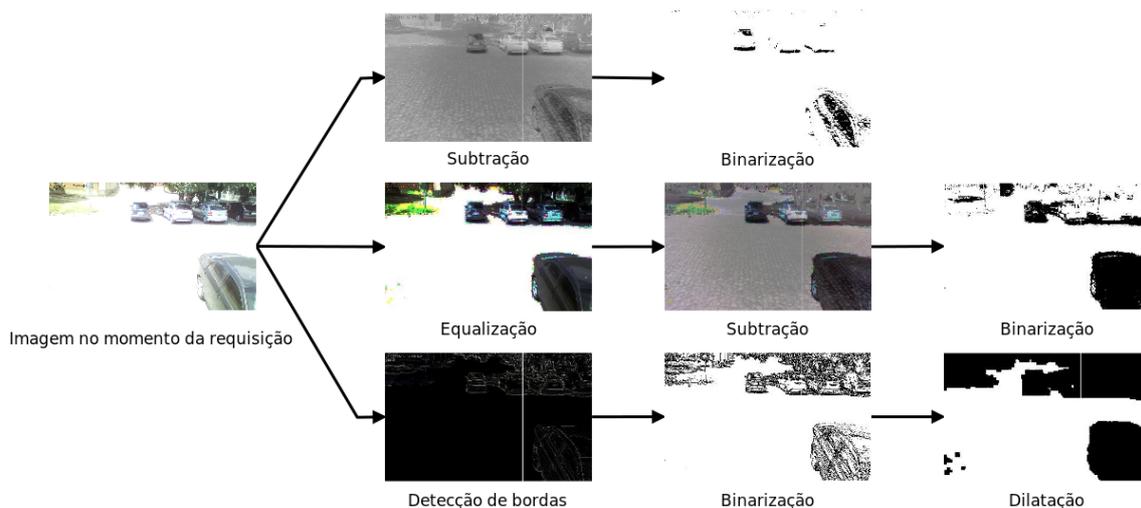


Figura 4. Técnicas Utilizadas para Detecção de Vagas

5. Experimento e Resultado Obtidos

Para desenvolvimento do trabalho foi realizado um experimento utilizando duas câmeras captando imagens de um estacionamento durante aproximadamente 10 horas contínuas (entre 8h e 18h) localizado na Universidade Federal de Alagoas. Com esse experimento, criamos um repositório de imagens que foram utilizadas na avaliação dos algoritmos aqui apresentados. As câmeras foram programadas para capturar e salvar as imagens em um repositório de imagens a cada movimento detectado em seu campo de visão. Ao salvar as imagens as câmeras guardam a informação do horário na qual a imagem foi capturada.

A análise do desempenho do sistema é feita verificando o tempo total que o sistema utiliza para realizar toda sua execução, desde a requisição até o recebimento da resposta do sistema pelo usuário em cada arquitetura citada anteriormente. É também analisado o tempo apenas de execução dos algoritmos levando em consideração apenas o processamento no computador e nas placas Panda.

Com as imagens capturadas pelas câmeras, foi realizada uma indexação das mesmas em uma linha temporal (figura 5), onde cada câmera possui a sua e as imagens são inseridas nos horários na qual foram salvas.possivelmente



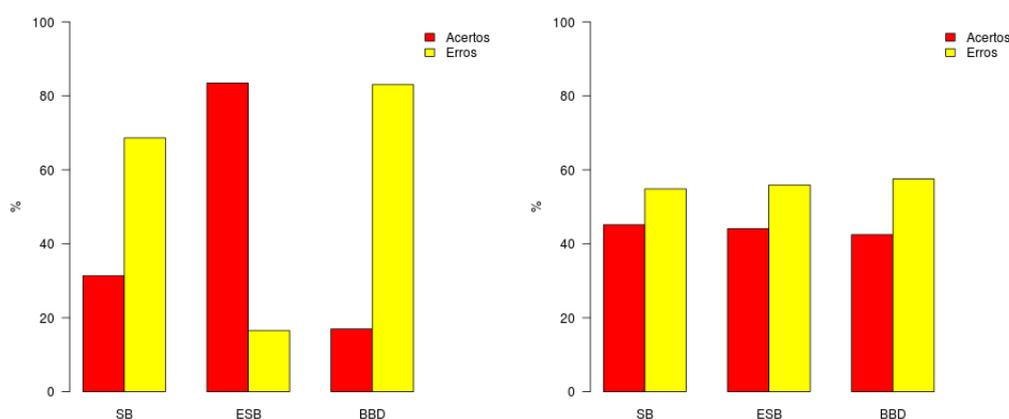
Figura 5. Exemplo de Linha do Tempo com Imagens

Para o processamento das imagens foram utilizados os três cenários definidos na seção 3. Todas as técnicas foram implementadas utilizando a linguagem de programação R [R Development Core Team 2011]. O servidor utilizado na aplicação é um computador com processador Intel Core2Duo P8600 de 2.4GHz, com memória de 4GB e utilizando o sistema operacional Ubuntu 12.04 64bits.

A avaliação dos resultados obtidos é feita por dois atributos: eficácia da solução e o tempo de execução. Para poder aferir a eficácia da solução, a priori, verificamos manualmente, em todas as imagens (aproximadamente 400 imagens), se existia ou não vaga disponível. Para o tempo de execução consideramos o tempo gasto entre o disparo de uma requisição e a resposta do sistema.

Na figura 6(a) é apresentado o gráfico com os resultados das simulações para a câmera 1. As duplas de barras representam os valores confirmados utilizando cada técnica. Podemos notar que existe uma grande diferença entre os resultados obtidos. A melhor solução para a câmera 1 foi a utilização da técnica ESB, que obteve uma taxa de acerto de aproximadamente 83.5%, enquanto que utilizando a BBD obteve 17% de acertos. A diferença de acerto é de 66.5%.

Na figura 6(b) temos os resultados das simulações realizadas com as imagens da câmera 2. Podemos notar que os resultados não tiveram uma variação tão grande quanto na câmera 1, eles se mantiveram praticamente constantes. No caso do estacionamento 2 a melhor técnica foi a SB, com uma taxa de acerto de 45.16% e a pior técnica foi a BBD, com uma taxa de acerto de 42.47%. Em ambas câmeras a pior técnica foi a BBD, isto se deve pelo fato principalmente da não utilização de uma imagem matriz (imagem do estacionamento vazio), com isto temos perda de informações necessárias para uma melhor detecção da existência de vagas.



(a) Resultados da Câmera 1

(b) Resultados da Câmera 2

Figura 6. Resultados da Eficácia das Técnicas Aplicadas

A diferença entre os resultados encontrados na simulação, é dado pela diferente posição da visada das câmeras. Enquanto que a câmera 1 possui uma visada diagonal do estacionamento, a câmera 2 possui uma visão frontal do estacionamento. Outro fator que influenciou os resultados obtidos foi a variação da luminosidade no estacionamento, as técnicas desenvolvidas agem de diferentes formas quanto a variação de luminosidade.

Nas figuras 7 à 9 temos o gráfico para o tempo total gasto pelo sistema. Podemos notar que a utilização do cenário CS tem um ganho considerável com relação os cenários que incluem a placa Panda. Isso ocorre pois o processamento no servidor é em média aproximadamente 460% mais rápido do que o da placa panda. Nos cenários CSP e CP,

que utilizam a placa Panda, podemos notar que a os tempos são praticamente os mesmos, ou seja, a diferença entre utilizar um servidor ou um ambiente totalmente distribuído é quase nula, o que favorece a implantação do sistema oportunista. As variações bruscas encontradas nas curvas se dá pelo fato do sistema ao encontrar uma vaga finaliza a execução, porém nas soluções desenvolvidas não existe um critério de seleção da área que será verificada primeiro, no caso do nosso trabalho as áreas são demarcadas previamente e são processadas em uma ordem fixa, sempre iniciando na primeira área demarcada.

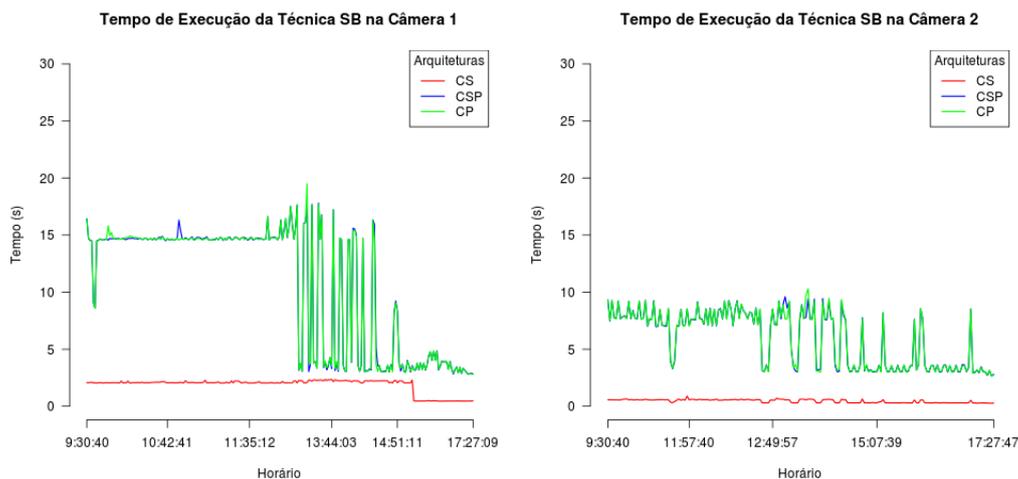


Figura 7. Tempo de Execução Total da Técnica #1

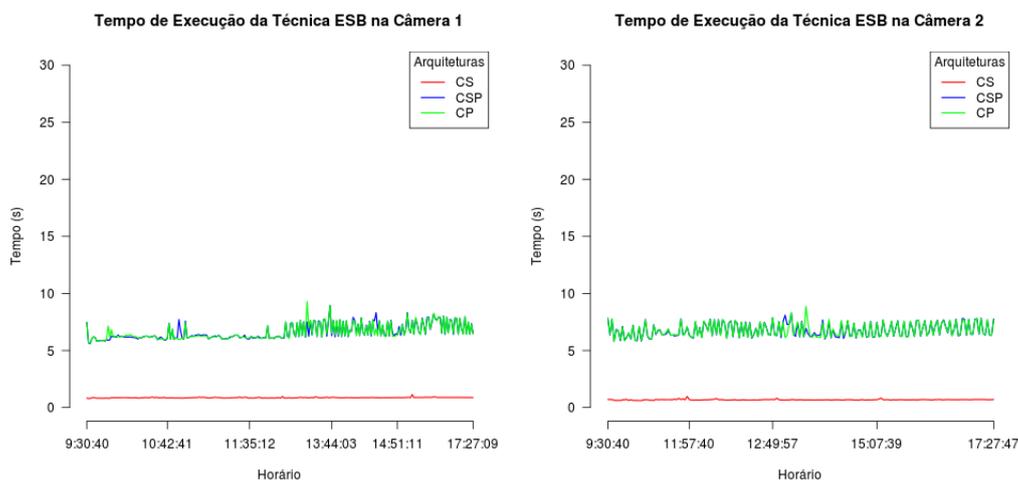


Figura 8. Tempo de Execução Total da Técnica #2

6. Considerações Finais

O trabalho teve como objetivo a apresentação de uma ferramenta para detecção de vagas de estacionamentos através de câmeras oportunistas. Foi também realizado a avaliação de três técnicas para a detecção de vagas de estacionamentos em imagens obtidas das câmeras.

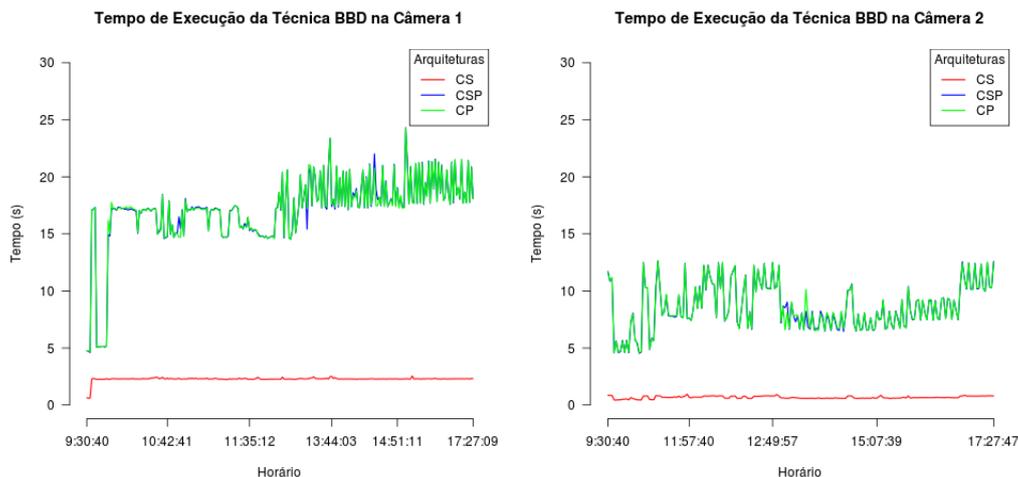


Figura 9. Tempo de Execução Total da Técnica #3

Os resultados obtidos nos experimentos serviram para a seleção da melhor técnica, apesar da qualidade da solução apresentar uma alta taxa de erros (33.88%). Com isso, é necessário desenvolver novas técnicas que solucionem os problemas encontrados e que melhorem ainda mais a taxa de acerto da solução. No entanto, identificamos que o tempo de execução com a placa panda não é afetado (na média a diferença é de 0.8 segundos) quando utilizamos um servidor para tratar as requisições. Isso mostra que uma estrutura completamente pode favorecer a sugestão de estacionamentos em um ambiente oportunista.

Como trabalhos futuros serão desenvolvidas diferentes técnicas para melhorar os resultados obtidos e a utilização de novas placas para processamento. Será também estudado critérios para seleção automática de possíveis áreas de estacionamentos, onde deverão ser selecionadas para o processamento primeiramente áreas com maior probabilidade de existir vagas. Além disso será desenvolvido um cenário totalmente distribuído onde cada câmera conversa apenas entre si e possuirá sua placa de processamento própria.

Referências

- Huang, C.-C., Dai, Y.-S., and Wang, S.-J. (2012). A surface-based vacant space detection for an intelligent parking lot. In *International Conference on ITS Telecommunications*.
- Jermurawong, J., Ahsan, M., Haidar, A., Dong, H., and 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. and Dalin, J. (2012). A method of parking space detection based on image segmentation and lbp. In *International Conference on Multimedia Information Networking and Security*.
- Otsu, N. (1979). A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man and Cybernetics*, 9:62–66.

- Park, W.-J., Kim, B.-S., Seo, D.-E., Kim, D.-S., and Lee, K.-H. (2008). Parking space detection using ultrasonic sensor in parking assistance system. In *IEEE Intelligent Vehicles Symposium*.
- Prewitt, J. M. S. (1970). Object enhancement and extraction. *Picture Processing and Psychopictorics*, 75:75 – 149.
- R Development Core Team (2011). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.
- Yang, J., Portilla, J., and Riesgo, T. (2012). Smart parking service based on wireless sensor networks. In *Conference of the IEEE Industrial Electronics Society*.
- Yusnita, R., Fariza, N., and Norazwinawati, B. (2012). Intelligent parking space detection system based on image processing. *International Journal of Innovation, Management and Technology*, 3(3):232 – 235.