

Um sistema para detecção de vagas de estacionamento utilizando câmeras

David H. S. Lima
Instituto Federal de Alagoas
Santana do Ipanema, Alagoas, Brasil
Email: david.lima@ifal.edu.br

Andre L. L. Aquino
Universidade Federal de Alagoas
Maceió, Alagoas, Brasil
Email: alla.lins@gmail.com

Eliana S. Almeida
Universidade Federal de Alagoas
Maceió, Alagoas, Brasil
Email: eliana.almeida@gmail.com

Resumo—This work proposes an embedded system that opportunistically detects free on-street parking using cameras. Intelligent boards are embedded in the cameras allowing a distributed processing and communication. Initially, it was evaluated two classical techniques for image processing applied to detect free on-street parking. The best one presents a success detection rate close to 100%. After that, it was evaluated the performance of the embedded system by using only the best image processing technique. This performance evaluation consider three different scenarios: centralized, hybrid, and embedded. The last one is the main proposal and contribution. The results reveal that embedded proposal had an average requisition time of 2.10s vs. 0.38s of centralized one. The hybrid one presents the worst results.

I. INTRODUÇÃO

Nesse trabalho é apresentado um sistema que utiliza imagens provenientes de câmeras previamente instaladas para detecção de vagas de estacionamento nas ruas. Na literatura existem soluções para o problema específico utilizando processamento de imagens, redes de sensores sem fios e veículos com sensores acoplados [1].

Inicialmente são avaliadas duas técnicas de processamento de imagens: *Subtração* e *Dilatação após Detecção de Bordas*. Em seguida, avaliamos o desempenho do sistema embutido utilizando a técnica *Dilatação após Detecção de Bordas*, que obteve os melhores resultados. A avaliação de desempenho considera três diferentes cenários: (i) *Centralizado*, que utiliza uma arquitetura cliente-servidor onde o servidor é responsável pelo processamento de imagem e a comunicação com os clientes; (ii) *Híbrido*, que é uma arquitetura onde o servidor é responsável pela comunicação com os clientes e as câmeras são responsáveis pelo processamento das imagens; e (iii) *Embutido*, que é uma arquitetura *ad-hoc* onde as câmeras são responsáveis tanto pelo processamento das imagens quanto pela comunicação com os clientes. Nas abordagens *Híbrida* e *Embutida*, é utilizada uma placa de processamento acoplada as câmeras adicionando a capacidade de processamento e de comunicação. Adicionalmente, foram avaliados cenários em larga escala utilizando simulação.

II. AVALIAÇÕES E RESULTADOS

As câmeras utilizadas nos experimentos foram as DLINK DCS-2130. Assumindo que a capacidade de processamento e comunicação pode ser realizada por uma placa acoplada, qualquer câmera pode ser utilizada. As placas acopladas nas câmeras foram as Placas Panda. O sistema operacional utilizado foi o Ubuntu-arm 12.04. Nas abordagens *Centralizada* e

Híbrida foi utilizado um servidor com um processador Intel Core2Duo P8600 de 2.4 GHz, 4 GB de memória RAM e Ubuntu 12.04 64 bits como sistema operacional. Todas as técnicas de processamento de imagens foram implementadas utilizando a biblioteca OPENCV na linguagem C++.

Foram utilizados sete pontos de monitoramento capturando imagens de dois estacionamentos distintos, durante 4 horas (8h às 12h). Todas as câmeras foram instaladas na mesma altura (4m). A distância entre cada câmera é cerca de 3m. As câmeras foram configuradas para capturar e salvar as imagens no repositório a cada movimento detectado. Com isso, foi criado um repositório de imagens para serem utilizadas na avaliação das técnicas de processamento de imagens. O repositório possui 3320 imagens.

Já para as simulações, foi utilizado o simulador de redes Sinalgo. 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.

A. Avaliação das técnicas de processamento de imagens

A técnica *Subtração* possui duas etapas de processamento de imagem. A primeira etapa é a subtração da imagem obtida com a requisição do cliente e uma imagem prévia do estacionamento vazio. Ao resultado, é aplicado uma técnica de binarização [2]. A técnica *Dilatação após Detecção de Bordas* aplica uma técnica de detecção de bordas [3], e, em seguida utiliza técnicas de binarização e dilatação.

Consideramos uma avaliação baseada em uma matriz de confusão [4]. Existem duas classes para serem inferidas (vagas *livre* e \sim *livre*). Para compor a matriz de confusão, foram identificados manualmente em cada imagem a presença de vagas de estacionamento. Para cada técnica foi quantificado o número de acertos e falhas (falsos negativos e positivos). Tabela I apresenta as matrizes de confusão para cada técnica avaliada levando em consideração as imagens do repositório.

Na Tabela I é possível notar que a técnica mais apropriada, nos experimentos, é a *Dilatação após Detecção de Bordas*, pois sua ACC e sua TPR são próximas de 1. Isto ocorre pois, diferentemente da técnica *Subtração*, as formas dos objetos são levadas em consideração. A alta PPV (0,958) na técnica *Dilatação após Detecção de Bordas* indica que muitos dos resultados positivos provenientes dos testes são valores verdadeiros positivos.

Tabela I. MATRIZES DE CONFUSÃO PARA CADA TÉCNICA

		SUBTRAÇÃO		
		Valores reais		
		livre	~livre	
Inferido	livre	TP = 1075	FP = 182	PPV \cong 0.855
	~livre	FN = 1966	TN = 97	NPV \cong 0.047
		TPR \cong 0.353	SPC \cong 0.347	ACC \cong 0.353

		DILATAÇÃO APÓS DETECÇÃO DE BORDAS		
		Valores reais		
		livre	~livre	
Inferido	livre	TP = 2939	FP = 128	PPV \cong 0.958
	~livre	FN = 102	TN = 151	NPV \cong 0.597
		TPR \cong 0.966	SPC \cong 0.541	ACC \cong 0.931

B. Avaliação de desempenho via experimentação

A avaliação de desempenho das abordagens propostas considera o tempo médio gasto para executar uma requisição completa do usuário 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 anteriormente.

Tabela II apresenta o tempo médio gasto para executar uma requisição em cada cenário. 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 possuir melhores resultados.

Tabela II. TEMPOS MÉDIOS DE UMA REQUISIÇÃO

Arquitetura	Média	IC	Desvio Padrão
<i>Centralizada</i>	0.24	[0.23, 0.24]	0.12
<i>Híbrida</i>	2.03	[2.00, 2.06]	0.52
<i>Embutida</i>	1.35	[1.33, 1.38]	0.62

C. 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.

A arquitetura *Embutida* possui algumas diferenças das demais, 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. Para encaminhar as mensagens na rede utilizamos a abordagem de roteamento desenvolvida por Maia et al. [5].

Para a simulação são levados em consideração dois parâmetros (Tabela III): o parâmetro *Requisições* é relativo a quantidade de requisições realizadas simultaneamente; o parâmetro *Câmeras* é relativo a quantidade de câmeras utilizadas na simulação.

Tabela III. VALORES DOS PARÂMETROS UTILIZADOS NA SIMULAÇÃO

Parâmetros	Valor 1	Valor 2	Valor 3
Requisições	100	500	1000
Câmeras	100	200	400

Na Figura 1 é apresentado o resultado da simulação para as três abordagens utilizadas. No eixo x temos a quantidade de câmeras utilizadas. No eixo y temos a média dos tempos (em segundos) das requisições e o intervalo de 95% de confiança destas médias. É possível notar que a abordagem *Embutida*, consegue ter melhores tempos do que a abordagem *Híbrida*. 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*.

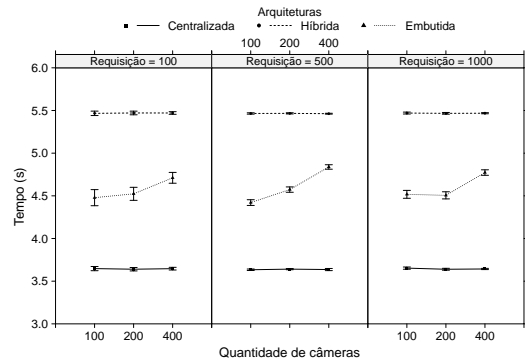


Figura 1. Resultado da simulação com todas arquiteturas

A quantidade de requisições não influencia no resultado obtido, visto que variando a quantidade de requisições realizadas o comportamento das curvas são similares. O principal fator que influencia o resultado é a quantidade de câmeras envolvidas.

III. CONCLUSÃO

Este trabalho apresentou um sistema para detecção de vagas de estacionamento utilizando placas acopladas à câmeras de segurança. Como trabalhos futuros temos o aperfeiçoamento das técnicas de processamento de imagens, utilizando mais características presentes nas imagens para obter melhores resultados; execução de experimento para avaliar o comportamento do sistema com uma maior quantidade de câmeras; adaptar solução para utilização em VANETs.

AGRADECIMENTOS

Agradecemos a Fundação de Amparo a Pesquisa de Alagoas pelo apoio parcial deste trabalho.

REFERÊNCIAS

- [1] G. Revathi and V. R. S. Dhulipala, "Smart parking systems and sensors: A survey," in *International Conference on Computing, Communication and Applications*, 2012.
- [2] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.
- [3] J. M. S. Previtt, "Object enhancement and extraction," *Picture Processing and Psychopictorics*, vol. 75, pp. 75 – 149, 1970.
- [4] S. V. Stehman, "Selecting and interpreting measures of thematic classification accuracy," *Remote Sensing of Environment*, vol. 62, no. 1, pp. 77–89, 1997.
- [5] G. Maia, A. L. L. Aquino, D. L. Guidoni, and A. A. F. Loureiro, "A multicast reprogramming protocol for wireless sensor networks based on small world concepts," *Journal of Parallel and Distributed Computing*, vol. 73, no. 9, pp. 1277–1291, 2013.